

ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

" CANCELACION ADAPTIVA DE RUIDO "

MARIA SOLEDAD JIMENEZ JIMENEZ

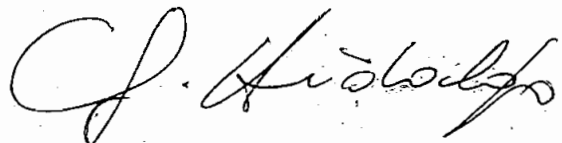
TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO
EN ELECTRONICA Y TELECOMUNICACIONES

QUITO, MARZO DE 1988



Certifico que el presente trabajo
ha sido realizado en su totalidad
por la señorita:

María Soledad Jiménez Jiménez.

A handwritten signature in cursive script, appearing to read 'G. Hidalgo'.

Ing. Gualberto Hidalgo
DIRECTOR DE TESIS

A MIS PADRES

Y HERMANOS

A PATRICIO

AGRADECIMIENTO

Al Ing. Gualberto Hidalgo por su acertada dirección en el desarrollo de la presente tesis.

A todas aquellas personas que de una u otra forma prestaron su contingente en la realización de este trabajo.

INDICE

CAPITULO I INTRODUCCION A FILTROS ADAPTIVOS

1.1	Procesamiento adaptivo.....	1
1.1.1	Sistemas adaptivos de procesamiento de señal.....	2
1.2	Filtros adaptivos.....	7
1.2.1	Filtros programables.....	9
1.2.1.1	Filtros recursivos.....	9
1.2.1.2	Filtros no recursivos.....	11
1.2.2	Algoritmos adaptivos.....	12
1.3	Cancelación adaptiva de ruido.....	15
1.3.1	Principio de cancelación adaptiva de ruido..	17
1.3.2	Cancelación adaptiva de ruido en una señal periódica.....	21
1.4	Generación de la señal a procesarse.....	24
1.4.1	Generación de la señal periódica.....	25
1.4.2	Generación de ruido blanco y coloreado.....	26
1.4.2.1	Generación de ruido blanco gaussiano.....	26
1.4.2.2	Generación de ruido coloreado.....	29

CAPITULO II ALGORITMO ADAPTIVO DE MINIMOS CUADRADOS MEDIOS O LMS (LEAST-MEAN-SQUARE)

2.1	Fundamento teórico.....	38
2.1.1	El combinador lineal adaptivo.....	38
2.1.2	El algoritmo adaptivo LMS	41
2.1.3	El filtro adaptivo LMS	49
2.2	Análisis de convergencia.....	52
2.2.1	Convergencia del vector de coeficientes de ponderación.....	52

2.2.2	Constantes de tiempo y curva de aprendizaje.	58
2.3	Implementación de programas.....	67

CAPITULO III ALGORITMO DE MINIMOS CUADRADOS RECURSIVOS O
RLS (RECURSIVE-LEAST-SQUARES)

3.1	Fundamento teórico.....	74
3.1.1	El estimador recursivo.....	74
3.1.2	Derivación del algoritmo adaptivo RLS	87
3.1.2.1	Algoritmo RLS preventaneado.....	87
3.1.2.2	Algoritmo RLS ponderado exponencialmente....	87
3.1.2.3	Interpretación estocástica.....	105
3.1.3	El filtro adaptivo RLS	107
3.2	Análisis de convergencia.....	109
3.3	Implementación de programas.....	110

CAPITULO IV RESULTADOS Y CONCLUSIONES

4.1	Resultados experimentales.....	115
4.1.1	Señal a procesarse.....	115
4.1.2	Resultados sobre el filtro adaptivo LMS	120
4.1.3	Resultados sobre el filtro adaptivo RLS	146
4.2	Análisis comparativo de los resultados obtenidos mediante los algoritmos desarrollados.	169
4.2.1	Eficiencia computacional.....	169
4.2.2	Velocidad de convergencia.....	172
4.2.3	Calidad de la señal filtrada.....	177
4.3	Conclusiones.....	178
4.4	Recomendaciones.....	180

ANEXOS

ANEXO A MANUAL DE USO DEL PROGRAMA

A.1	Objetivo.....	A1
A.2	Requerimientos de hardware.....	A1

A.3	Requerimientos de software.....	A2
A.4	Descripción de los programas.....	A4
A.5	Ejecución del programa.....	A8

ANEXO B LISTADOS DE LOS PROGRAMAS

B.1	Programa TESIS.BAS	B2
B.2	Programa DATGEN.BAS	B7
B.3	Programa ALGORITHM.BAS	B16
B.4	Programa CURVA.BAS	B23
B.5	Programa FOURIER.BAS	B31
B.6	Archivo FILTRO.BAT	B36
B.7	Archivo MONO.BAT	B36
B.8	Archivo COLOR.BAT	B36
B.9	Archivo GRAFICO.BAT	B37
B.10	Hoja de trabajo GRAFICO.WK1	B40

ANEXO C GENERACION DE RUIDO BLANCO

C.1	Conceptos básicos.....	C1
C.1.2	Distribución normal.....	C6
C.1.3	Método del límite central.....	C7

ANEXO D FILTROS DE BUTTERWORTH

D.1	Diseño de filtros de Butterworth análogos y digitales.....	D1
D.2	Conversión de filtros análogos en digitales por medio de la transformación bilineal.....	D7
D.3	Transformación en frecuencia.....	D12

BIBLIOGRAFIA

RESUMEN

El presente trabajo de tesis tiene como finalidad estudiar los filtros adaptivos en el contexto de canceladores adaptivos de ruido y, utilizar dicha técnica en una aplicación específica como es la eliminación de una interferencia de banda ancha (ruido blanco o coloreado) en una señal periódica. El filtro adaptivo que se utilizará tiene dos partes constitutivas fundamentales que lo caracterizan : el filtro programable y el algoritmo adaptivo.

Como filtro programable se utilizará una estructura no recursiva o filtro FIR (Finite-Impulse-Response) y, como algoritmos adaptivos se estudiarán y aplicarán dos de ellos: el algoritmo de mínimos cuadrados medios o LMS (Least-Mean-Squares) y el algoritmo de mínimos cuadrados recursivos o RLS (Recursive-Least-Squares).

En el capítulo I se realiza un breve estudio sobre los tipos de filtros programables (recursivos y no recursivos), sobre los algoritmos adaptivos (LMS y RLS) y, se establece el principio de cancelación adaptiva de una interferencia de banda ancha en una señal periódica.

En el capítulo II se trata exclusivamente el algoritmo adaptivo LMS, su derivación y, se realiza un estudio teórico sobre su convergencia.

En el capítulo III se estudia, en un desarrollo similar al del capítulo II, el algoritmo adaptivo RLS, su derivación y su convergencia.

Finalmente en el capítulo IV se desarrollan varios ejemplos respecto a cada uno de los algoritmos implementados y, se hace una análisis comparativo entre dichas técnicas. Como resultado del estudio realizado en este trabajo se derivan algunas conclusiones, las cuales se adjuntan al final del capítulo.

Adicionalmente se incluyen algunos anexos, así: el anexo A, que trata sobre el manual de uso de los programas desarrollados; el anexo B, donde constan los listados de dichos programas; el anexo C, donde se realiza un breve estudio y justificación sobre la técnica utilizada para la generación de ruido blanco; y, el anexo D, donde se estudian los tipos de filtros que se utilizaron para filtrar el ruido blanco a fin de obtener el ruido coloreado.

CAPITULO I

INTRODUCCION A FILTROS ADAPTIVOS

1.1 PROCESAMIENTO ADAPTIVO

Las técnicas adaptivas están evolucionando muy rápidamente en el campo de procesamiento de señal en comunicaciones; principalmente debido al desarrollo de métodos de procesamiento sofisticados y poderosos y, en particular al hecho de que los dispositivos de hardware usados para implementar sistemas de comunicaciones modernos son adecuados para realizar los algoritmos adaptivos.

Existen dos operaciones básicas en comunicaciones, estas son : la transmisión y el procesamiento de señales, a esta última le compete tanto la preparación de la señal para la transmisión como la recuperación de la misma luego de la recepción; se debe entonces encontrar algún mecanismo que permita cumplir de la mejor forma dicho cometido.

Los sistemas convencionales de procesamiento de señal, suponen que se conoce el canal de transmisión, y que las señales a transmitirse o recibirse son bien definidas y estacionarias; este conocimiento previo hace posible determinar un método de procesamiento de señal óptimo, el mismo que puede implementarse en un sistema fijo y usarse todo el tiempo.

Sin embargo, frecuentemente las características del canal de transmisión no son bien conocidas o varían con el tiempo y, las señales a tratarse son en su mayoría no estacio-

narias; en tales circunstancias se hace necesario utilizar un sistema que tenga la capacidad de ajustar sus propios parámetros automáticamente de modo que se pueda adaptar a un canal de transmisión particular y a una señal particular, un sistema con tales características constituye un procesador adaptivo.

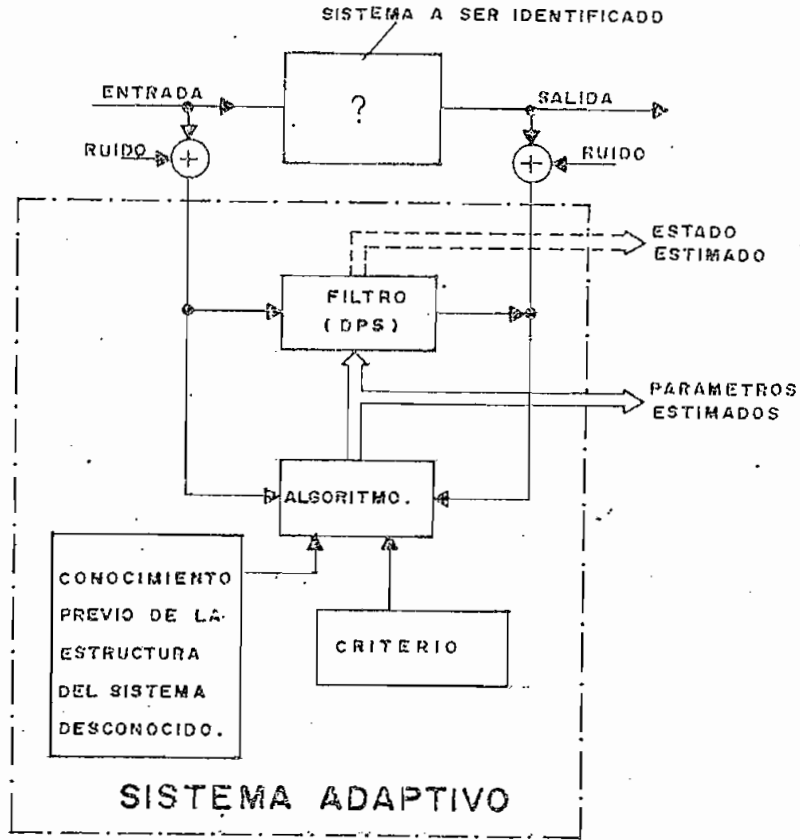
1.1.1 SISTEMAS ADAPTIVOS DE PROCESAMIENTO DE SEÑAL

Los sistemas adaptivos se pueden clasificar en tres clases principales de acuerdo a la meta que ellos persiguen, así:

a) Sistemas para Identificación de Sistemas

Una configuración típica se muestra en la figura 1.1, en la cual el objetivo es determinar los parámetros o el estado de un sistema desconocido; para llevar a cabo este cometido se requiere disponer de un mínimo conocimiento previo acerca de la estructura del sistema y tener acceso tanto a la señal de entrada como a la de salida. Los sistemas de control adaptivos son casi invariablemente de este tipo; la salida es el conjunto de parámetros estimados del sistema desconocido (o un estimado de su estado). Por lo general en aplicaciones de control estos estimados se usan posteriormente para intervenir en la entrada o en el sistema bajo análisis, a fin de que sus parámetros o estado adquieran algún valor deseado. En procesamiento de señal los parámetros estimados son usualmente la salida deseada.

Sin embargo, la identificación de sistemas constituye la meta fundamental para sistemas adaptivos en control, pero rara vez lo es para sistemas adaptivos en telecomunicaciones.



D.P.S. \equiv Dispositivo de Procesamiento de Señal

Fig. 1.1 Configuración típica de un sistema para identificación de sistemas.

b) Sistemas para Estimación de Señal

Su estructura se muestra en la figura 1.2 y, su finalidad constituye el estimar la salida de un sistema desconocido basándose en una medida probablemente ruidosa de la señal de entrada y una versión distorsionada de la salida. Por lo general no es necesario tener un conocimiento previo acerca de la naturaleza precisa del sistema desconocido, sino únicamente de algunas características globales (por ejemplo lineal o no lineal, recursivo o no recursivo, etc.). Los parámetros determinados a través del algoritmo

por tanto no necesariamente tienen una relación directa con los parámetros físicos del sistema desconocido, pero son tales que el dispositivo de procesamiento de señal (filtro) puede producir un estimado suficientemente confiable de la salida.

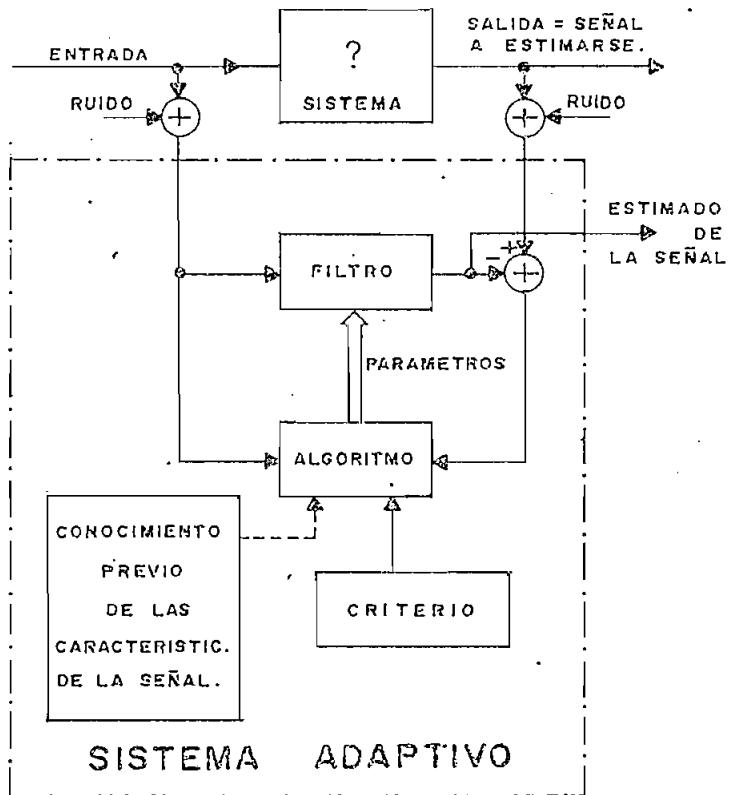


Fig. 1.2 Configuración típica de un sistema para estimación de señales.

Puesto que los sistemas adaptativos están estimando la señal de salida, consideraciones acerca de la entrada y salida pueden influenciar notablemente tanto en la implementación como en el comportamiento del sistema adaptativo, por consiguiente es menester estar seguros de su validez, caso contrario podrían provocar comportamientos inesperados en el sistema.

Un ejemplo típico de esta clase de sistemas lo constituye un cancelador adaptivo de eco y un cancelador adaptivo de ruido.

c) Sistemas para Corrección de Señales

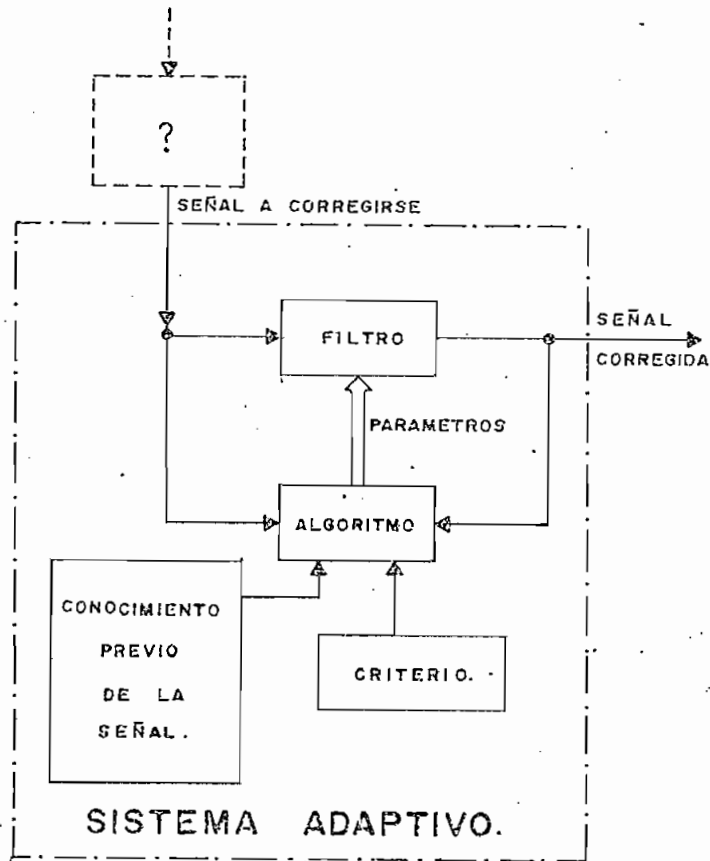


Fig. 1.3 Configuración típica de un sistema para corrección de señales.

La estructura básica de un sistema para corrección de señales se muestra en la figura 1.3, en el cual las características de la señal de entrada requieren corregirse; probablemente esto es necesario a causa de que un sistema desconocido alteró las características de la señal original, a la que el sistema de corrección no tiene acceso.

El sistema adaptivo debe tener un criterio bien definido acerca de las propiedades deseadas en la señal corregida; así como también debe disponer de alguna información previa acerca del tipo de señal a tratarse y el tipo de deterioro a corregirse. Un ejemplo típico lo encontramos en los ecualizadores, codificadores predictivos lineales de señal hablada, etc.

Luego de haber realizado una descripción muy general de los tipos de sistemas adaptivos para procesamiento de señal, se desprende algunas características inherentes a los mismos, así:

- En todos los sistemas adaptivos para procesamiento de señal, es necesario disponer de algún conocimiento previo de la señal de entrada y/o del sistema desconocido; tal conocimiento previo es de fundamental importancia, pues de resultar dicha información incorrecta podría producir serios errores y una degradación en el comportamiento del sistema.
- Es importante tener en consideración que la adaptación empieza luego de que se inicia la transmisión y requiere de algún tiempo antes de que se produzca un resultado confiable, y puesto que durante este tiempo la comunicación será insatisfactoria, es necesario que la adaptación sea rápida. Un comportamiento similar se requiere frente a cambios rápidos en las características del sistema desconocido o en la señal.
- Es preciso disponer de un criterio de calidad, que evalúe si efectivamente el sistema adaptivo está realizando una búsqueda óptima para la determinación de sus parámetros.

- Se debe determinar un algoritmo que permita calcular a partir de las señales disponibles, los valores de los parámetros que satisfagan óptimamente el criterio de calidad. La mayoría de algoritmos se basan en el criterio del error cuadrático medio.
- Finalmente, todo sistema adaptivo debe incluir en su estructura un filtro programable, algunas veces llamado "dispositivo de procesamiento de señal", en el cual, en base a la señal de entrada disponible y usando los parámetros generados por el algoritmo, se produce un estimado de la señal de salida deseada. La mayoría de estructuras para filtros usados en sistemas adaptivos son lineales aunque también existe tendencia a tratar con sistemas no lineales.

1.2 FILTROS ADAPTIVOS

Los filtros adaptivos tienen aplicación en diversas áreas de procesamiento de señal, que incluyen ecualización automática, arreglos adaptivos de antenas, control adaptivo, reconocimiento de patrones, canceladores de eco, canceladores de ruido, etc.

Operación del Filtro Adaptivo

La figura 1.4 muestra el esquema de un filtro adaptivo, en el cual constan como partes esenciales un filtro programable y un algoritmo adaptivo. La respuesta de frecuencia o función de transferencia del filtro programable se modifica para que puedan pasar sin degradación las componentes deseadas de la señal y, atenuar cualquier interferencia o distorsión en la señal de entrada. El algoritmo es la parte del filtro adaptivo donde se derivan los paráme-

tros (coeficientes) del filtro programable en base a mediciones de la señal de entrada y del error de salida.

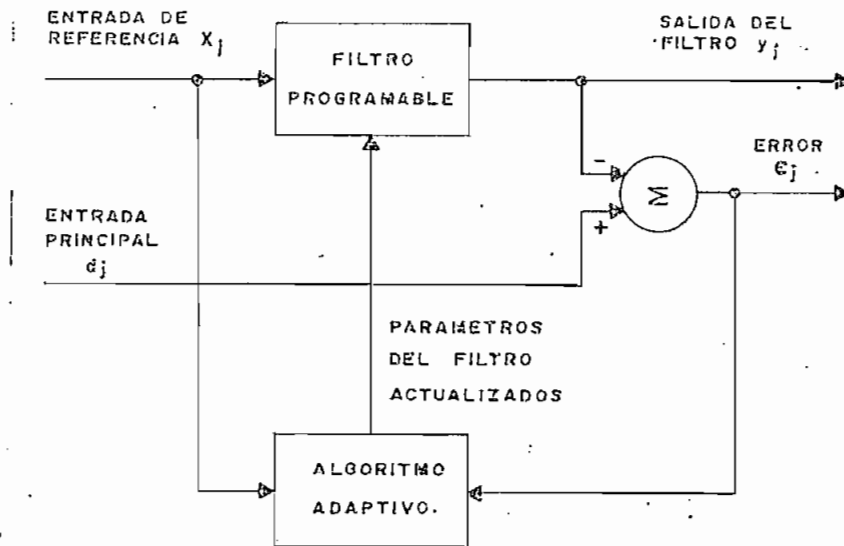


Fig. 1.4 Esquema general de un filtro adaptivo.

Como se observa en la figura, la señal de entrada x_j se procesa en el filtro programable obteniendo a la salida una señal y_j , la cual se compara con la entrada principal d_j , dándonos como resultado una señal de error e_j , que se utiliza posteriormente en el algoritmo adaptativo para actualizar los coeficientes de ponderación del filtro (usualmente de modo iterativo), de tal manera que el error se minimiza progresivamente.

En un sistema adaptivo se requiere un mínimo de información previa acerca de la señal de entrada, el filtro adaptivo estima los estadísticos de dicha señal y ajusta su propia respuesta de modo que se minimize el error e_j .

1.2.1 FILTROS PROGRAMABLES

Existen diversos tipos de filtros programables que se pueden usar en el diseño de filtros adaptivos, sin embargo, dos de ellos son de fundamental interés:

1.2.1.1 FILTROS RECURSIVOS

Constituyen la estructura de filtro digital más generalizada, figura 1.5. La respuesta de un filtro recursivo de n-etapas está determinada por una ecuación de diferencias de orden n; donde el valor de una muestra de la salida del filtro y_j , está dada por una combinación lineal de las muestras de entrada presentes y pasadas (x_{j-k}), así como también de las muestras de salida anteriores (y_{j-k}):

$$y_j = - \sum_{k=1}^n b_k y_{j-k} + \sum_{k=0}^n a_k x_{j-k} \quad (1.1)$$

Tal estructura da lugar al diseño de un filtro "polo-cero" donde los coeficientes b determinan la localización de los polos, y los coeficientes a la localización de los ceros; el número de polos y ceros u orden del filtro está determinado por el número de etapas de retardo.

La función de transferencia de un filtro recursivo n-etapas tendrá entonces la siguiente forma:

$$H(z) = \frac{\sum_{k=0}^n a_k z^{-k}}{1 + \sum_{k=1}^n b_k z^{-k}} \quad (1.2)$$

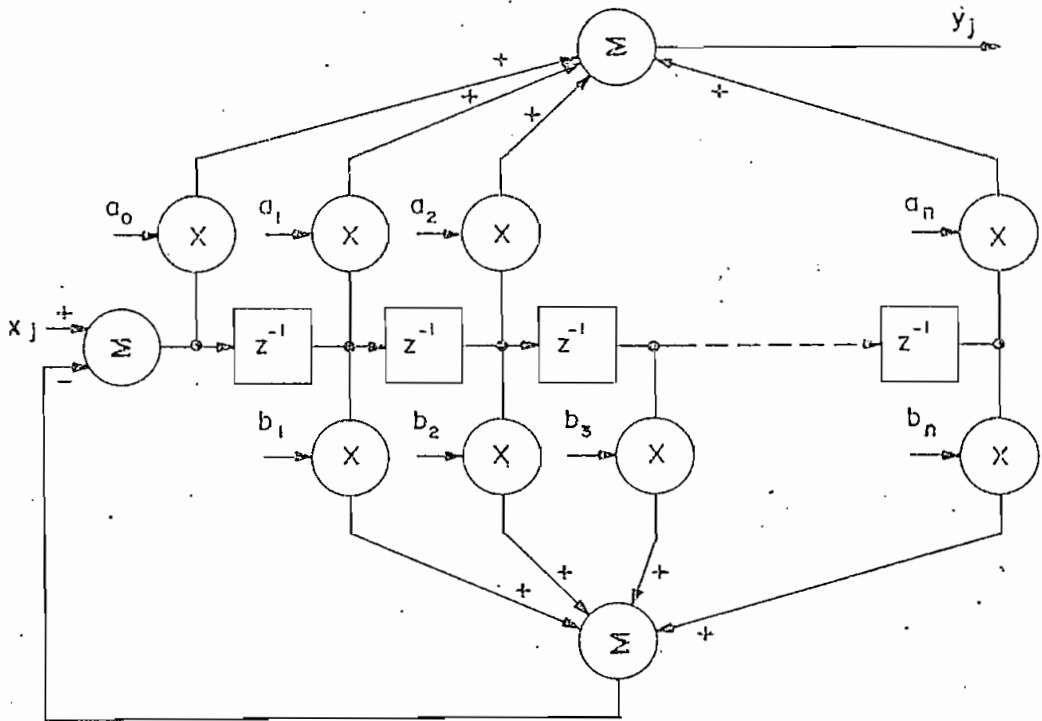


Fig.1.5 Estructura de un filtro recursivo (IIR).

Tal estructura recursiva teóricamente tendría una memoria infinita, efectivamente con un solo impulso en la excitación este tipo de filtro puede responder hasta un tiempo infinito; por esta razón se los denomina Filtros de Respuesta Impulso Infinita (IIR), los cuales no son incondicionalmente estables a menos que se den ciertas restricciones sobre los valores de los coeficientes b . Sin embargo la presencia tanto de polos como de ceros permite realizar un filtro de características de corte agudo, incorporando un ancho de banda de transición angosto con un número relativamente modesto de etapas de retardo.

Una desventaja de los filtros IIR es que no ofrecen control sobre la respuesta de fase del filtro; sin embargo el mayor problema es su posible inestabilidad debida a polos que pueden estar localizados fuera de la región estable.

1.2.1.2 FILTROS NO RECURSIVOS

Una forma de superar la desventaja de la inestabilidad potencial en los filtros, es diseñar un filtro todo cero (figura 1.6), el cual es incondicionalmente estable y, tiene una memoria limitada, la misma que se determina por el número de etapas de retardo y, da lugar al diseño de filtros no recursivos FIR (Filtros de respuesta impulso finita). Las muestras retardadas de la señal de entrada se multiplican por los coeficientes "h", y los productos se suman para formar la salida del filtro. De modo que la respuesta de un filtro no recursivo de n-etapas estará dada por una ecuación de diferencias de orden n:

$$y_j = \sum_{k=0}^n h_k x_{j-k} \quad (1.3)$$

donde h_k en realidad constituye la respuesta impulso unitaria del filtro, de manera que la función de transferencia (transformada Z de la respuesta impulso) será¹:

$$H(z) = \sum_{k=0}^n h_k z^{-k} \quad (1.4)$$

Los filtros FIR constituyen el tipo de filtro adaptivo más comunmente utilizado, principalmente debido a su simplicidad y generalidad. Su función de transferencia se puede modificar fácilmente para controlar los coeficientes de ponderación del filtro.

¹ Nótese que la ecuación (1.3) representa el sumatorio de convolución.

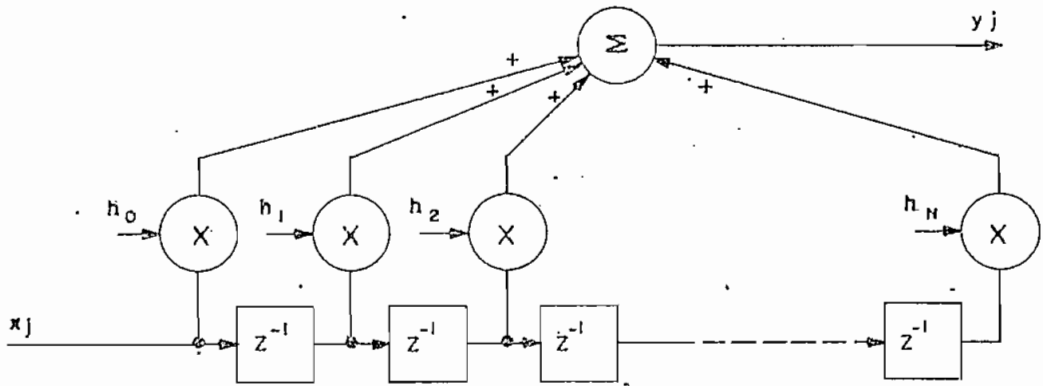


Fig. 1.6 Estructura de un Filtros de respuesta impulso finita (FIR).

Debido a que este filtro incorpora solo ceros es necesario un gran número de elementos de retardo para conseguir una respuesta de frecuencia de corte agudo; sin embargo el filtro es siempre estable y puede proveer una respuesta de fase lineal.

Estos dos tipos de filtros IIR y FIR pueden constituirse en el filtro programable de nuestro sistema adaptivo, figura 1.4. Sin embargo, el presente trabajo de tesis se desarrollará en base a un filtro FIR; principalmente debido a la estabilidad incondicional que estos ofrecen y a su simplicidad y generalidad.

1.2.2 ALGORITMOS ADAPTIVOS

Una vez seleccionada la estructura del filtro, debe hacerse lo mismo respecto al algoritmo adaptivo. Existen algu-

nas alternativas las cuales ofrecen una mayor complejidad a cambio de un comportamiento mejorado. La complejidad es fácil de caracterizar; y, respecto al comportamiento del algoritmo adaptivo existen dos medidas que lo evalúan, estas son: la velocidad de adaptación y la exactitud de la función de transferencia luego de la adaptación; existe un compromiso entre estas dos medidas, así pues, para una clase particular de algoritmo adaptivo a medida que la velocidad de adaptación se incrementa, empeora la exactitud de la función de transferencia luego de la adaptación.

Los algoritmos adaptivos pueden ser clasificados en:

- algoritmos en bloque
- algoritmos recursivos

En los algoritmos en bloque, la señal de entrada es dividida en el tiempo en bloques; cada uno de ellos se procesa independientemente y, se determinan los parámetros óptimos del filtro adaptivo para cada uno de los bloques.

En los algoritmos recursivos, el algoritmo adaptivo se implementa como un conjunto continuo de operaciones recursivas, de manera que con cada nueva muestra de datos de entrada, se genera un nuevo conjunto de parámetros.

Los algoritmos en bloque tienen una memoria finita, pues el conjunto actual de parámetros depende sólo de un segmento finito de las señales de entrada pasadas; mientras que los algoritmos recursivos tienen una memoria infinita, puesto que el conjunto actual de parámetros depende (al menos en teoría) de toda una historia completa del pasado de la señal de entrada. Sin embargo, algunas versiones de algoritmos recursivos tienen también una memoria finita, de modo que ésta no es una característica distintiva absoluta.

Los algoritmos en bloques y los algoritmos recursivos de memoria finita son de gran interés para aquellos medios que cambian relativamente rápido, pues en tales medios la memoria infinita en lugar de ser beneficiosa es nociva; un ejemplo de una señal de entrada cuyos estadísticos están variando rápidamente es la señal hablada (speech), siendo común utilizar algoritmos en bloques para procesar dichas señales. Por otro lado, en un medio estacionario o de variaciones muy lentas, tanto los algoritmos en bloques como los recursivos pueden utilizarse con buenos resultados.

La idea del compromiso entre la velocidad de adaptación y la exactitud del filtro adaptivo, está íntimamente ligada con la longitud de la memoria del algoritmo. Así, cuando se incrementa la longitud de la memoria, se considera mayor información acerca de la señal de entrada en la adaptación, y por consiguiente la exactitud puede ser mayor; sin embargo, el inconveniente está en que la adaptación se vuelve más lenta puesto que nueva información no se asimila tan rápidamente.

Los algoritmos de adaptación en bloques son usualmente del tipo de Mínimos Cuadrados LS (Least Square). En esta clase de algoritmos, la diferencia entre la respuesta deseada del filtro y la respuesta actual (señal de error), se eleva al cuadrado, se suma sobre todo el bloque, y luego se minimiza en base a la elección de los parámetros óptimos del filtro. El algoritmo LS se puede implementar en una forma recursiva, dando origen al denominado algoritmo de Mínimos Cuadrados Recursivos RLS (Recursive Least Squares), donde la señal de error al cuadrado se suma no sobre un bloque, sino íntegramente desde el inicio. A fin de controlar o limitar la memoria del algoritmo, a menudo se usa una función de ponderación la cual da más peso a las muestras más recientes y menor peso a las primeras mues-

tras de la señal de entrada; se suele conocer a ésta versión como el algoritmo de Mínimos Cuadrados Recursivos Ponderados.

El algoritmo de Mínimos Cuadrados Medios LMS (Least Mean Square) o de la Gradiente SG (Stochastic Gradient) es una técnica de adaptación recursiva simple y es una de las pioneras en el campo de sistemas adaptivos; en los que la determinación de los parámetros actuales, en un instante dado, se obtiene en base a un estimado del gradiente de la señal de error.

Dentro de la gran variedad de algoritmos adaptivos existentes, éste trabajo estudiará dos de ellos: el algoritmo de Mínimos Cuadrados Medios (LMS), sobre el cual trata el capítulo II; y el algoritmo de Mínimos Cuadrados Recursivos (RLS) que se estudiará en el capítulo III. El algoritmo LMS ofrece una baja complejidad computacional frente al RLS, pero como contraparte la velocidad de convergencia es mucho menor en el LMS; estas dos alternativas ofrecen una amplia posibilidad de realizar un análisis comparativo de su comportamiento.

1.3 CANCELACION ADAPTIVA DE RUIDO

La cancelación adaptiva de ruido constituye una de las principales aplicaciones de los filtros adaptivos en procesamiento de señal. El primer sistema de cancelación adaptiva de ruido fue diseñado y construido en 1965 por dos estudiantes de la Universidad de Stanford, el propósito era cancelar la interferencia de 60Hz. presente a la salida de un amplificador y registrador de electrocardiogramas; a partir de este año la técnica de cancelación adaptiva de ruido se ha aplicado de manera exitosa a un gran número de problemas como lo son, la eliminación de inter-

ferencias periódicas en general, la eliminación de interferencias de banda ancha (ruido), la cancelación de ruido en señales habladas (speech), la eliminación de ecos en líneas de transmisión telefónica a gran distancia, etc.

El método usual de estimar una señal que contiene ruido aditivo, consiste en pasar dicha señal a través de un filtro que tiende a suprimir el ruido mientras mantiene la señal relativamente inalterada. Para tal propósito se podrían usar tanto filtros fijos como adaptivos, el diseño de un filtro fijo requiere de un conocimiento previo de la señal y del ruido; en tanto que un filtro adaptivo tiene la capacidad de ajustar sus parámetros automáticamente y su diseño requiere de un mínimo conocimiento previo acerca de la señal y de las características del ruido.

La cancelación de ruido es una variación del filtrado óptimo, el cual es muy ventajoso en muchas aplicaciones; utiliza una entrada auxiliar o de referencia obtenida por medio de uno o mas sensores localizados en puntos en el campo del ruido donde la señal es débil o indetectable, ésta entrada se filtra, de forma tal que tienda a parecerse lo más cercanamente posible al ruido que contamina la señal de la entrada principal, y posteriormente se sustrae de dicha entrada principal (que contiene tanto la señal como el ruido); como resultado el ruido primario se atenúa o elimina por cancelación.

A primera vista, el sustraer el ruido mediante ésta técnica podría parecer un proceso peligroso, efectivamente si se lo hace en una forma inapropiada podría producir un incremento en la potencia del ruido a la salida; sin embargo si el filtrado y la substracción se controlan con un proceso adaptivo adecuado, es posible cancelar el ruido sin riesgo de producir distorsión en la señal o incrementar el nivel de ruido a la salida. En casos donde la cancelación

adaptiva de ruido es aplicable, se puede conseguir niveles de rechazo al ruido que sería difícil o imposible conseguir por técnicas de filtrado directo.

1.3.1 PRINCIPIO DE CANCELACION ADAPTIVA DE RUIDO

La figura 1.7 muestra el principio de un cancelador adaptivo de ruido. Una señal s_j se transmite a través de un canal a un sensor que también recibe un ruido n_j el cual es no correlacionado con la señal, la señal más el ruido s_j+n_j constituye la entrada principal al cancelador; la entrada de referencia es provista por un segundo sensor que recibe un ruido n_{1j} no correlacionado con la señal s_j pero correlacionado en alguna forma desconocida con el ruido n_j (se podría pensar que el ruido n_{1j} se obtiene desde un sensor localizado en un punto en el campo del ruido donde la señal es débil o indetectable). El ruido n_{1j} se filtra a fin de generar a la salida una señal y_j que sea una réplica tan cercana como sea posible del ruido n_j , de manera que al restarla de la entrada principal s_j+n_j produzca una salida del sistema $z_j=s_j+n_j-y_j$.

Si de antemano se conoce las características del canal a través del cual se transmiten las entradas de referencia y principal hacia los sensores respectivos, es posible diseñar un filtro fijo capaz de transformar n_{1j} en n_j , de modo que al restar la salida del filtro desde la entrada principal produciría una salida del sistema igual a la señal sin ruido. Sin embargo, en vista de que las características del canal de transmisión son desconocidas o conocidas sólo aproximadamente y rara vez de naturaleza fija, no es posible usar filtros fijos; más aún, si fuera posible usar un filtro fijo, sus características deberían ajustarse con una precisión difícil de conseguir, y el más pequeño error

provocaría un incremento en la potencia de ruido de salida.

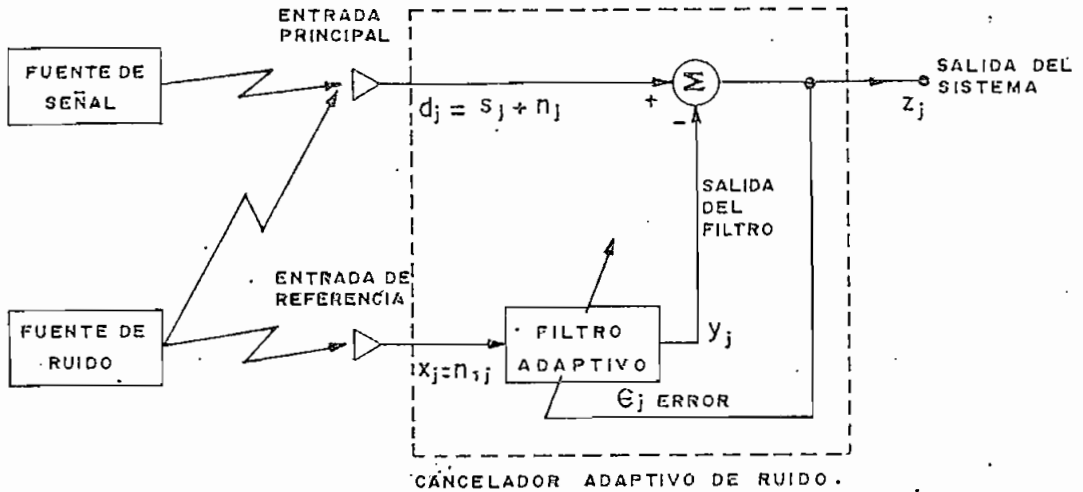


Fig. 1.7 Concepto de cancelación adaptiva de ruido.

Un filtro adaptivo difiere de uno fijo en que tiene la capacidad de ajustar automáticamente su propia respuesta impulsiva; el ajuste se realiza por medio de un algoritmo que responde a una señal de error dependiente entre otras cosas de la salida del filtro. Con un algoritmo apropiado el filtro puede operar bajo condiciones cambiantes y se puede reajustar por sí mismo continuamente para minimizar la señal de error.

La señal de error que se usa en el proceso adaptivo depende de la naturaleza de la aplicación. En los sistemas para cancelación de ruido el objetivo práctico es producir una salida del sistema $z_j = s_j + n_{1j} - y_j$ que sea el mejor estimado de la señal s_j . Este objetivo se consigue realimentando la salida del sistema (la señal de error) al filtro adaptivo y ajustando el filtro a través de un algoritmo adaptivo para minimizar la potencia de salida total del sistema.

El éxito de un sistema de cancelación adaptiva de ruido depende entonces de la disponibilidad de una señal de referencia adecuada, la misma que contenga ruido no correlacionado con la señal pero altamente correlacionado con el ruido aditivo que contamina dicha señal.

Se podría pensar que es necesario conocer previamente tanto la señal s_j como el ruido n_j y n_{1j} antes de diseñar y utilizar el filtro para producir la señal de cancelación del ruido y_j . Sin embargo, mediante un argumento simple se puede demostrar que sólo es necesario en algunos casos un mínimo de información previa acerca de la señal s_j , del ruido n_j y n_{1j} ; o de sus interrelaciones estadísticas o determinísticas.

Asumamos que s_j, n_j, n_{1j} y y_j son estadísticamente estacionarias y tienen media cero; asumamos también que s_j es no correlacionada con n_j ni tampoco con n_{1j} , y supongamos que n_j es correlacionado con n_{1j} . La señal de salida del sistema z_j es:

$$z_j = s_j + n_j - y_j \quad (1.5)$$

elevando al cuadrado se obtiene:

$$z_j^2 = s_j^2 + (n_j - y_j)^2 + 2s_j(n_j - y_j) \quad (1.6)$$

tomando el valor esperado a ambos lados de la ecuación (1.6) y considerando que s_j es no correlacionada con n_j y y_j , se llega a la siguiente expresión:

$$\begin{aligned} E[z_j^2] &= E[s_j^2] + E[(n_j - y_j)^2] + 2E[s_j(n_j - y_j)] \\ &= E[s_j^2] + E[(n_j - y_j)^2] \end{aligned} \quad (1.7)$$

Cuando se ajusta el filtro a fin de minimizar $E[z_j^2]$, la potencia de la señal $E[s_j^2]$ no se afecta; por consiguiente, la potencia de salida mínima será:

$$\min E[z_j^2] = E[s_j^2] + \min E[(n_j - y_j)^2] \quad (1.8)$$

de manera que al ajustar el filtro para minimizar $E[z_j^2]$, también se minimizará $E[(n_j - y_j)^2]$; por tanto, la salida del filtro y_j es el mejor estimado del ruido en la entrada principal n_j . Mas aún, cuando se minimiza $E[(n_j - y_j)^2]$, también se minimiza $E[(z_j - s_j)^2]$, puesto que de acuerdo a la ecuación (1.4)

$$(z_j - s_j) = (n_j - y_j) \quad (1.9)$$

Por tanto, el ajustar el filtro para minimizar la potencia de salida total es equivalente a hacer que la salida z_j sea el mejor estimado de la señal s_j .

La salida z_j contendrá la señal s_j más el ruido, de acuerdo con la ecuación (1.5) el ruido de salida está dado por $(n_j - y_j)$. Puesto que al minimizar $E[z_j^2]$ se minimiza $E[(n_j - y_j)^2]$; al minimizar la potencia de salida total se minimiza la potencia del ruido de salida, y dado que la señal en la salida permanece constante, se maximiza la relación señal a ruido en la salida.

De acuerdo con la ecuación (1.7) la potencia de salida más pequeña que se puede conseguir es $E[z_j^2] = E[s_j^2]$, cuando se logra esto, $E[(n_j - y_j)^2] = 0$; y por tanto, $y_j = n_j$, y $z_j = s_j$. Concluyendo se puede decir que el minimizar la potencia de salida permite obtener una señal a la salida del sistema libre de ruido.

1.3.2 CANCELACION ADAPTIVA DE RUIDO EN UNA SENAL PERIODICA

En ésta sección se describe el principio de cancelación adaptiva de ruido aplicado en la eliminación de una interferencia de banda ancha que contamina a una señal periódica, donde no se dispone de una fuente de referencia externa; y, en él se basa el presente trabajo de tesis. Al parecer la técnica de cancelación adaptiva de ruido no se podría aplicar para reducir o eliminar esta clase de interferencia, puesto que no se dispone de la entrada de referencia necesaria en el proceso adaptivo como se estableció en la sección anterior; sin embargo, es posible obtener a partir de la entrada principal una entrada de referencia adecuada para esta aplicación, esto se realiza insertando una línea de retardo fija en la entrada principal y tomando la salida de dicha línea de retardo como la entrada de referencia, y por lo tanto ésta representa una versión retardada de la entrada principal.

Cancelador de Interferencia Periódica en una señal de banda ancha, sin fuente de referencia externa

Consideremos el caso donde una señal de banda ancha está contaminada por una interferencia periódica (lo contrario a nuestro problema) y así mismo no se dispone de una entrada de referencia externa libre de señal; aplicando el principio anteriormente expuesto es posible obtener una entrada de referencia adecuada a partir de la señal de entrada principal, la cual permitirá realizar la cancelación satisfactoriamente, tal principio se muestra gráficamente en la figura 1.8 [1].

El retardo debe ser de una longitud suficiente para garantizar que las componentes de señal de banda ancha en la entrada de referencia sean no correlacionadas con las de

la entrada principal, esto es s_j y s_{j-T} sean no correlacionadas entre sí y sean no correlacionadas con la señal de interferencia; las componentes de interferencia, a causa de su naturaleza periódica permanecerán correlacionadas entre sí, es decir n_j y n_{j-T} conservarán su alto grado de correlación. El filtro adaptivo tratará de producir a su salida una señal que sea lo más similar posible a la señal de interferencia periódica, de manera que al restarla de la entrada principal generará a la salida del cancelador de ruido la señal de banda ancha libre de interferencia.

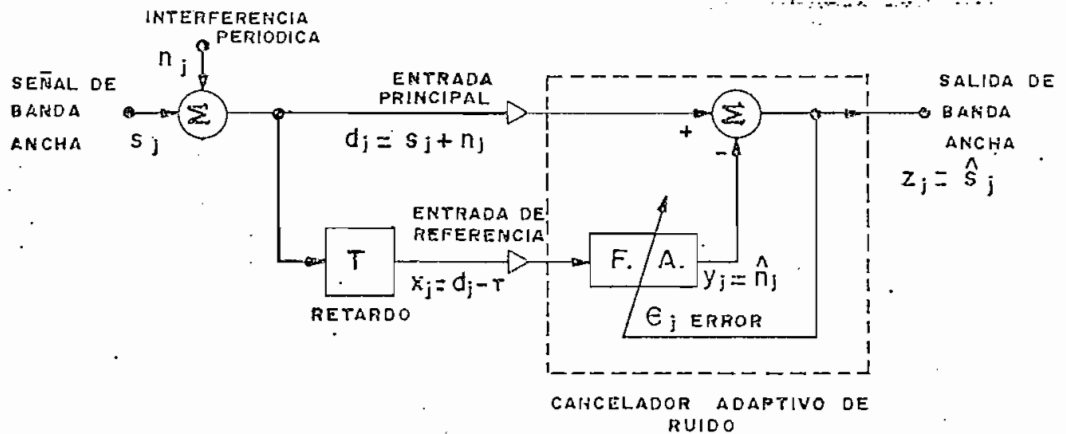


Fig. 1.8 Cancelador de interferencia periódica sin fuente de referencia externa.

Usando los argumentos de la sección 1.3.1, se puede demostrar que el minimizar la potencia en la salida del sistema $z_j = \hat{s}_j$ provocará que la señal $y_j = \hat{n}_j$ sea el mejor estimado de la señal n_j ¹.

¹ El símbolo $\hat{\cdot}$ denota estimado

Cancelador de interferencia de banda ancha en una señal periódica, sin fuente de referencia externa

El mismo principio utilizado en la aplicación anterior se puede usar, con una ligera variante, para eliminar una interferencia de banda ancha que contamina a una señal periódica; así pues, si la salida del sistema de cancelación se toma desde la salida del filtro, el resultado es un sistema capaz de extraer una onda periódica a partir de una señal contaminada por un ruido de banda ancha aditivo. La figura 1.9 muestra el cancelador adaptivo de ruido para esta aplicación [1].

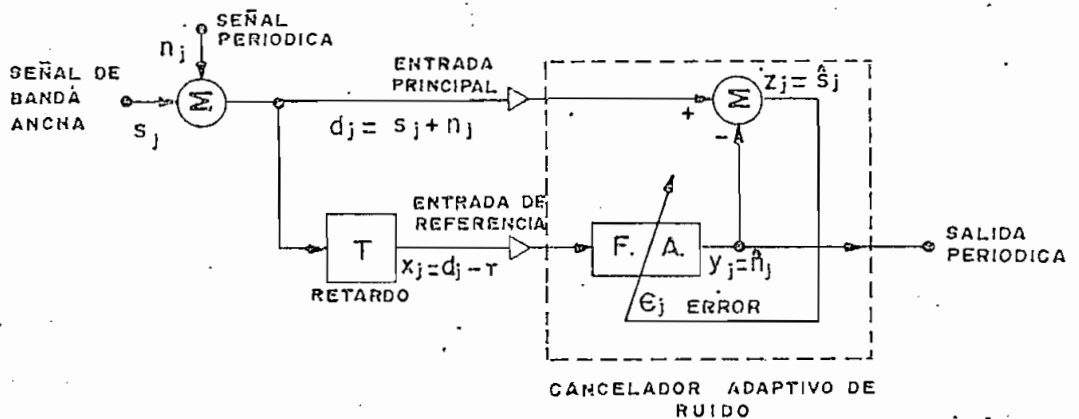


Fig. 1.9 El cancelador de ruido adaptivo como un sistema para eliminar una interferencia de banda ancha.

Ambos sistemas de cancelación de ruido son idénticos en su estructura, pues los dos tratan a la onda periódica como la interferencia y a la de banda ancha como la señal deseada; por tanto en los dos casos a la salida del filtro programable se trata de reproducir el ruido (señal periódica), de manera que al restarlo de la entrada principal éste se elimina por cancelación. De esta forma, se tendrá disponible la señal periódica a la salida del filtro y, la

señal de banda ancha a la salida del sistema de cancelación de ruido.

La presente tesis tratará exclusivamente sobre los filtros adaptivos utilizados en un cancelador de interferencia de banda ancha donde no se dispone de una fuente de referencia externa; siendo ésta una de las tantas aplicaciones que tienen los filtros adaptivos en cancelación de ruido. Esta misma aplicación será estudiada mediante dos tipos de filtros adaptivos, los cuales son idénticos en su estructura, ambos del tipo no recursivo (filtros FIR), pero difieren en el algoritmo utilizado para la actualización de los coeficientes; así pues se estudiará el algoritmo LMS y el algoritmo RLS.

El cancelador de ruido adaptivo se simulará por medio de un computador, con la entrada una senoide o suma de sinusoides que representarán a la señal, y ruido blanco o coloreado que representará a la interferencia.

1.4 GENERACION DE LA SEÑAL A PROCESARSE

En esta sección se describirán tanto la naturaleza de las señales a procesarse en el filtro adaptivo, como el método utilizado para obtener dichas señales.

A fin de simular el comportamiento del sistema de cancelación adaptiva de ruido, se procesará una señal constituida por una onda periódica mas ruido aditivo. La señal periódica estará conformada por una o varias sinusoides con diferentes características de amplitud y frecuencia; en tanto que la interferencia será una señal de ruido blanco gaussiano, o ruido coloreado el cual se obtiene luego de filtrar el ruido blanco.

Consecuentemente la señal a procesarse será de la forma:

$$X = \sum_{i=1}^M A(i) \text{ Sen}(2\pi \text{ FS}(i) t) + n \quad (1.10)$$

donde n constituye el término de ruido blanco o coloreado, M el número de sinusoides, $A(i)$ la amplitud y $\text{FS}(i)$ la frecuencia correspondientes a cada una de las sinusoides.

1.4.1 GENERACION DE LA SEÑAL PERIODICA

Se mencionó anteriormente que la señal periódica estará constituida por una o más sinusoides, estableciéndose como máximo 3, puesto que un número mayor de sinusoides carece de sentido práctico.

Considérese entonces una señal sinusoidal de frecuencia FS que es muestreada con una frecuencia FM superior a la especificada por el teorema del muestreo, esto es:

$$S(i) = A(i) \text{ Sen} \frac{2\pi \text{ FS}(i) (j-1)}{\text{FM}} \quad j=1,2,\dots,\text{NUM} \quad (1.11)$$

$$\text{FM} \geq 2 \text{ FS}(i) \quad (1.12)$$

donde: FS = frecuencia de la señal

j = variable de tiempo discreta

FM = frecuencia de muestreo

NUM = número de muestras

i = subíndice para la i -ésima senoide.

Si la señal corresponde a una suma de sinusoides, cada una de ellas se genera en idéntica forma; sin embargo, se debe satisfacer el teorema del muestreo para la máxima frecuencia, esto es:



$$FM \geq 2 FS \text{ máx.}$$

donde FS máx. corresponde a la máxima frecuencia de las sinusoides generadas.

1.4.2 GENERACION DE RUIDO BLANCO Y COLOREADO

En esta sección se describirán dos procesos, la generación de ruido blanco gaussiano y la generación de ruido coloreado.

1.4.2.1 GENERACION DE RUIDO BLANCO GAUSSIANO

Un análisis y justificación de la técnica utilizada para la generación de las muestras de ruido blanco gaussiano se encuentra en el ANEXO C; por tanto en esta sección se realizará únicamente una breve descripción de las características del ruido, y de su generación.

El ruido blanco se caracteriza por que su espectro tiene todas las componentes de frecuencia en igual proporción, tal como lo muestra la figura 1.10(a); y una función de autocorrelación como la de la figura 1.10(b), donde se observa que dos muestras diferentes cualesquiera de ruido blanco gaussiano son no correlacionadas.

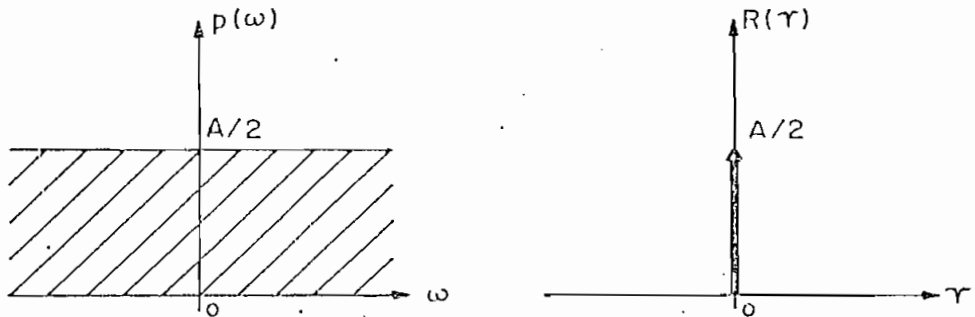


Fig. 1.10 Señal de ruido blanco gaussiano. (a) espectro de potencia. (b) función de autocorrelación.

El ruido blanco constituye una idealización, puesto que no existe en la realidad; sin embargo, para todos los casos prácticos se considera al ruido de banda ancha como ruido blanco, y se lo trata como tal, cometándose un error despreciable. En adelante se considerará al ruido blanco simplemente como un ruido de banda ancha y se generará entonces este tipo de ruido.

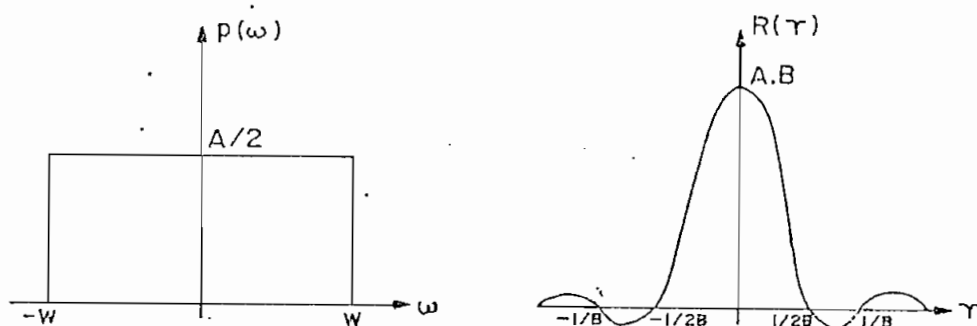


Fig. 1.11 Ruido blanco gaussiano de banda limitada. (a) espectro de densidad de potencia. (b) autocorrelación.

El ruido blanco gaussiano de banda limitada, es un proceso cuya densidad espectral de potencia es constante en una banda de frecuencia limitada $[-W, W]$ y cero fuera de ella, como lo muestra la figura 1.11(a); y cuya función de autocorrelación se visualiza en la figura 1.11(b).

Para generar ruido blanco gaussiano, deben generarse muestras, es decir valores de variables aleatorias con distribución gaussiana o normal, que tengan un valor medio cero y varianza AB ; el método que se utilizará para generar este tipo de variables aleatorias es el método del límite central, el cual se encuentra ampliamente desarrollado en la tesis sobre Simulación Estadística [2], y en forma condensada se encuentra descrito en el ANEXO C.

RUTINA DE GENERACION DE RUIDO BLANCO

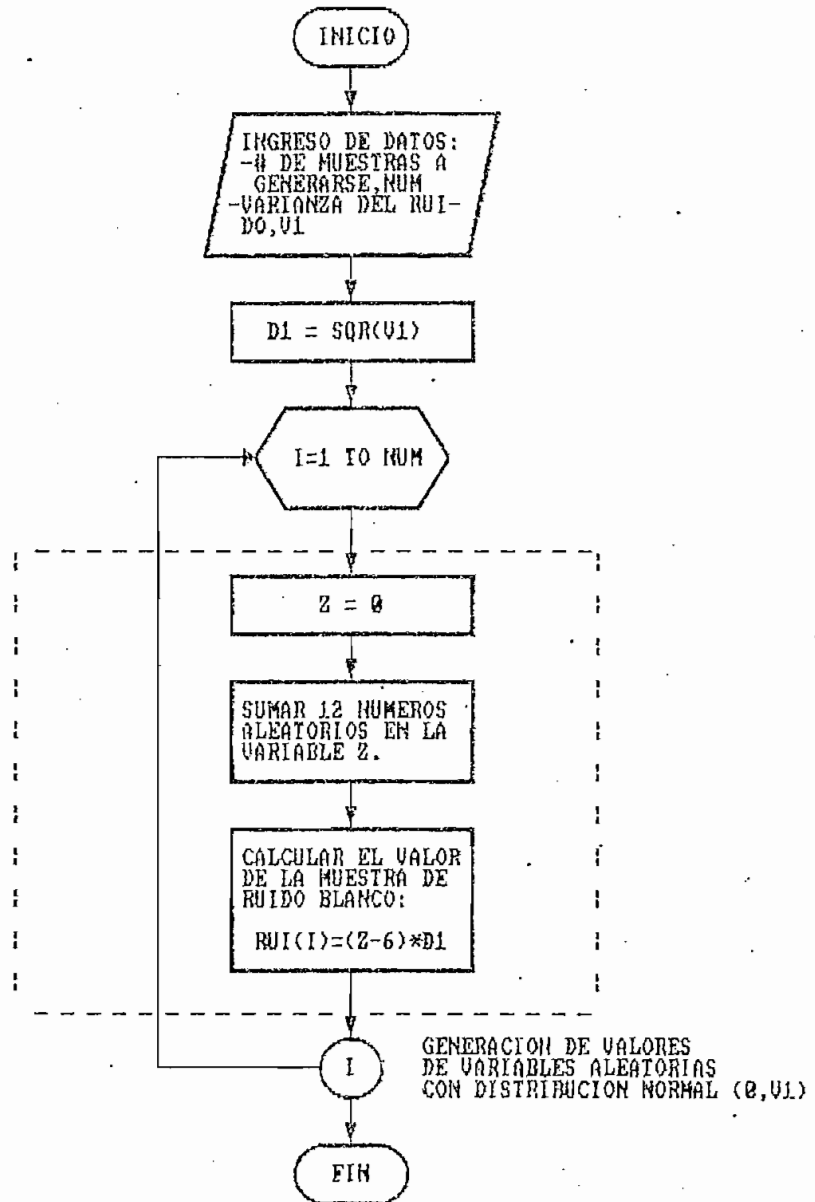


Fig.1.12 Diagrama de flujo para generar ruido blanco gaussiano.

La figura 1.12 muestra el diagrama de flujo para generar ruido blanco gaussiano. Las variables utilizadas en este diagrama de flujo tienen el siguiente significado:

NUM = número de muestras de ruido blanco
V1 = varianza del ruido
D1 = desviación estándar del ruido
Z = variable aleatoria, resultado de la suma de 12
 números aleatorios.
RUI(I) = vector que contiene las muestras de ruido-
 blanco, donde $I=1 \dots \text{NUM}$

La línea segmentada señala la parte del programa donde se generan los valores de las variables aleatorias con distribución normal $(0, V1)$, como lo señala el ANEXO C.

1.4.2.2 GENERACION DE RUIDO COLOREADO

Un ruido se considera coloreado cuando su espectro está limitado a un cierto rango de frecuencias. En general ruido coloreado se genera pasando ruido blanco a través de un filtro que tenga una función de transferencia apropiada.

Para la generación de ruido coloreado se utilizaron filtros digitales de Butterworth en tres versiones: pasabajos; pasabanda y pasaaltos; su análisis e implementación se encuentra detallado en el ANEXO D [3].

Los diagramas de flujo que se muestran a continuación constituyen las rutinas para generar ruido coloreado mediante filtros de Butterworth. Así:

a) Filtro de Butterworth Pasabajos

La rutina de simulación de un filtro de Butterworth pasabajos requiere que se den como datos: la frecuencia de corte FC en Hz. , la frecuencia de muestreo FM en Hz. y, el orden del filtro N. El filtro digital de Butterworth pasabajos tendrá NS=N/2 secciones en cascada, a la K-ésima sección le corresponderá una función de transferencia:

$$H_K(z) = \frac{A(K) (z^2 + 2z + 1)}{z^2 + B(K)z + C(K)} \quad (1.13)$$

donde $K=1,2,\dots,NS$

El valor de las constantes A(K),B(K) y C(K) viene dado por:

$$A(K) = WCP^2 X \quad (1.14)$$

$$B(K) = 2 (WCP^2 - 1) X \quad (1.15)$$

$$C(K) = 1 + WCP^2 + (2 WCP CS) X \quad (1.16)$$

donde:

$$WCP = \text{TAN}(\pi FC/FM) \quad (1.17)$$

$$CS = \text{COS} (\pi (2(K+NS)-1)/4 NS) \quad (1.18)$$

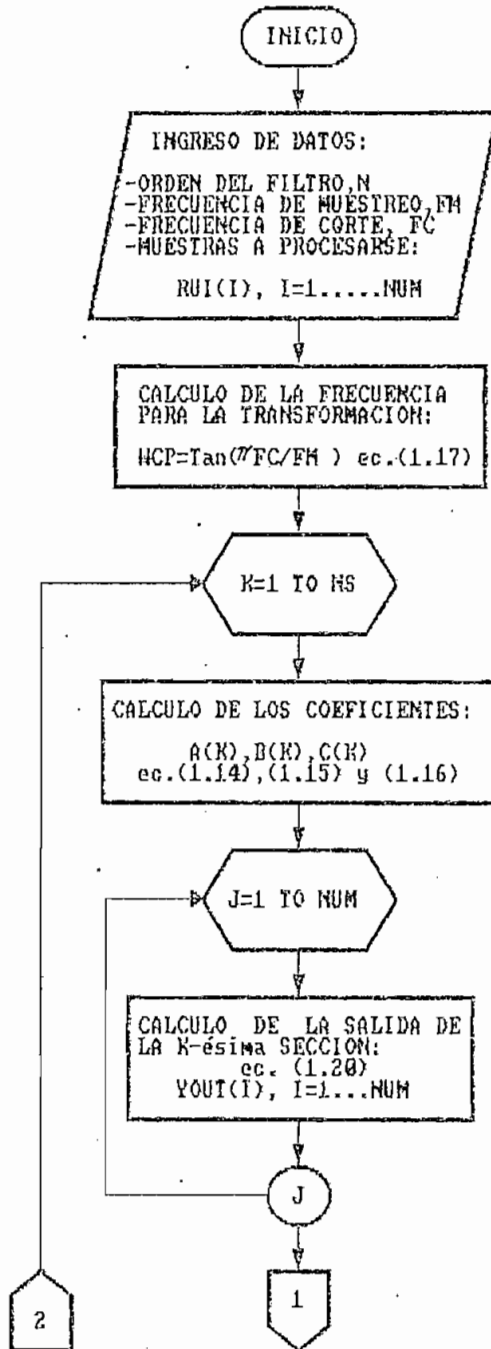
$$X = \frac{1}{1 + WCP^2 - 2 WCP CS} \quad (1.19)$$

de manera que si RUI_K(I) y YOUT_K(I) son la entrada y salida respectivamente de la K-ésima sección, tendremos:

$$YOUT_K(I) = A(K) [RUI_K(I) + 2 RUI_K(I-1) + RUI_K(I-2)] - B(K) YOUT_K(I-1) - C(K) YOUT_K(I-2) \quad (1.20)$$

El algoritmo para la obtención de un filtro pasabajos digital de Butterworth se esquematiza en la figura 1.13.

RUTINA DE SIMULACION
DE UN FILTRO PASABAJOS BUTTERWORTH



...CONTINUACION

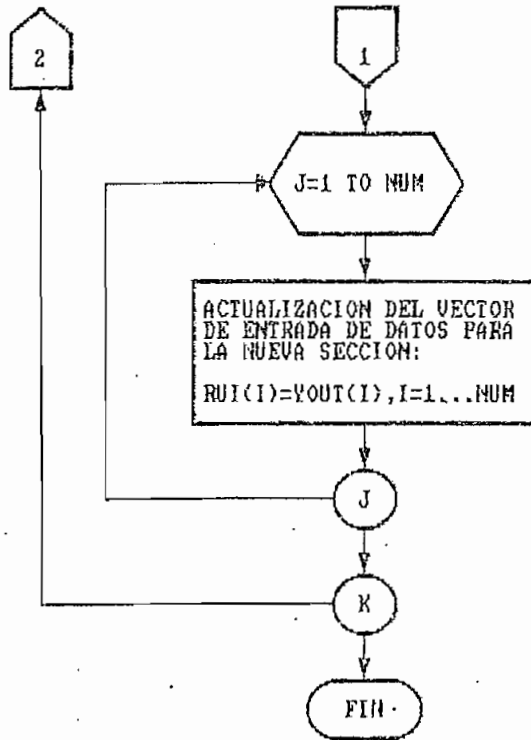


Fig. 1.13 Rutina para implementación de un filtro de Butterworth pasabajos.

b) Filtro de Butterworth Pasaaltos.

Para diseñar un filtro digital de Butterworth pasaaltos; se sigue un procedimiento muy similar al desarrollado para el filtro Butterworth pasabajos, se dan así mismo como datos: la frecuencia de muestreo F_M en Hz, la frecuencia de corte F_C en Hz y, el orden del filtro N . El filtro estará constituido por $NS=N/2$ secciones en cascada, cada una de las cuales con una función de transferencia:

$$H_K(z) = \frac{A(K) (z^2 - 2z + 1)}{z^2 + B(K)z + C(K)} \quad K=1,2,\dots,NS \quad (1.21)$$

donde el valor de las constantes A(K), B(K) y C(K) se determina mediante las siguientes relaciones:

$$A(K) = \frac{1}{1 + WCP^2 - 2 WCP CS} \quad (1.22)$$

$$B(K) = 2 (WCP^2 - 1) \quad (1.23)$$

$$C(K) = (1 + WCP^2 + (2 WCP CS)) A(K) \quad (1.24)$$

Si RUI_K(I) y YOUT_K(I) son la entrada y la salida respectivamente de la K-ésima sección, tendremos:

$$YOUT_K(I) = A(k) [RUI_K(I) - 2 RUI_K(I-1) + RUI_K(I-2)] - B(k) YOUT_K(I-1) - C(k) YOUT_K(I-2) \quad (1.25)$$

De manera que la rutina para la simulación del filtro pasaltos es la misma de la figura 1.13.

c) Filtro de Butterwoth Pasabanda.

Finalmente, el diagrama de flujo de la figura 1.14, muestra el mecanismo a utilizarse para la simulación de un filtro de Butterworth pasabanda, donde se dan como datos: las frecuencias de la banda de paso F1 y F2 en Hz, la frecuencia de muestreo FM en Hz y, el orden del filtro N; el filtro constará de NS=N/2 secciones en cascada, cada una con función de transferencia de la forma:

$$H_K(z) = \frac{A(K) (z^4 - 2z^2 + 1)}{z^4 + B(K)z^3 + C(K)z^2 + D(K)z + E(K)} \quad (1.26)$$

donde el valor de las constantes A(K), B(K) y C(K) viene dado por:

$$A(K) = WC^2 / X \quad (1.27)$$

$$B(K) = (4S + 2R - 2P - 4) / X \quad (1.28)$$

$$C(K) = (6S - 2Q + 6) / X \quad (1.29)$$

$$D(K) = (4S - 2R + 2P - 4) / X \quad (1.30)$$

$$E(K) = (S - R + Q - P + 1) / X \quad (1.31)$$

y se cumple:

$$WC = W2 - W1 \quad (1.32)$$

$$Q = WC^2 + (2 W1 W2) \quad (1.33)$$

$$S = W1^2 W2^2 \quad (1.34)$$

$$CS = \text{COS} ((2(K+NS)-1)/4 NS) \quad (1.35)$$

$$P = -2 WC CS \quad (1.36)$$

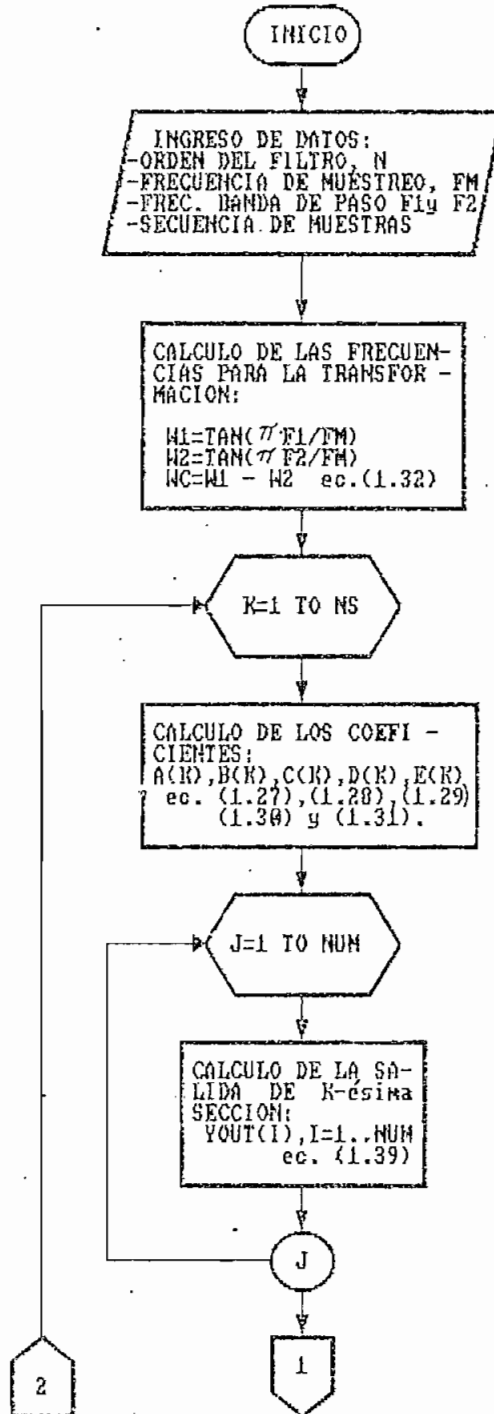
$$R = P W1 W2 \quad (1.37)$$

$$X = 1 + P + Q + R + S \quad (1.38)$$

Si RUI(I) es la entrada al filtro y YOUT(I) es su salida, se cumplirá la siguiente relación entre ellas.

$$\begin{aligned} YOUT_k(I) = & A(k) [RUI_k(I) - 2 RUI_k(I-2) + RUI_k(I-4)] - \\ & - B(k) YOUT_k(I-1) - C(k) YOUT_k(I-2) - \\ & - D(k) YOUT_k(I-3) - E(k) YOUT_k(I-4) \end{aligned} \quad (1.39)$$

RUTINA DE SIMULACION
DE UN FILTRO PASABANDA BUTTERWORTH



...CONTINUACION

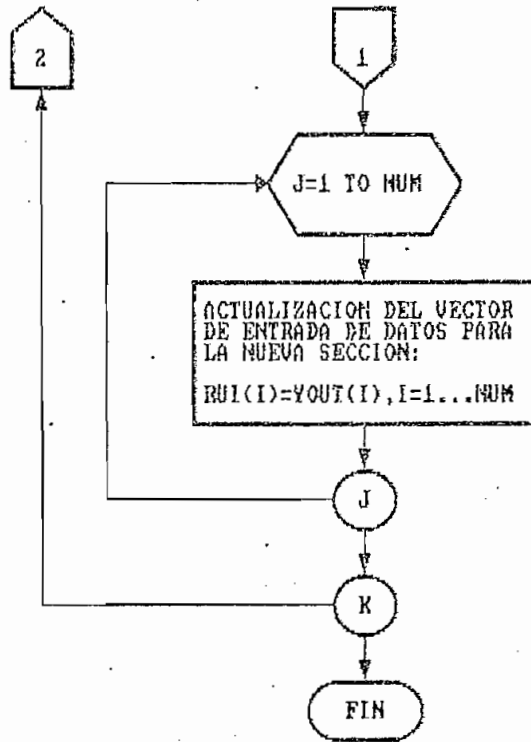


Fig. 1.14 Diagrama de flujo para la obtención de un filtro Butterworth pasabanda.

CAPITULO II

ALGORITMO ADAPTIVO DE MINIMOS CUADRADOS MEDIOS

o

LMS (LEAST-MEAN-SQUARES)

En este capítulo se estudiará el algoritmo adaptivo LMS conocido también como técnica del gradiente, desarrollado por Widrow y Hopf, y que constituye un predecesor del filtro adaptivo RLS. El algoritmo LMS ha ganado considerable popularidad a partir de los años 60, su simplicidad y fácil implementación lo hacen atractivo para muchas aplicaciones en las cuales interesa bajos requerimientos computacionales.

Las desventajas fundamentales del algoritmo LMS se relacionan con sus propiedades de convergencia, pues este algoritmo se basa en una técnica de búsqueda del gradiente para minimizar una función de comportamiento cuadrática, en tanto que el algoritmo RLS es mas bien un procedimiento del tipo Newton-Raphson, que como es bien conocido converge generalmente mucho más rápido que la técnica de búsqueda del gradiente.

2.1 FUNDAMENTO TEORICO

2.1.1 EL COMBINADOR LINEAL ADAPTIVO

El concepto de estimación óptima es de fundamental importancia en el tratamiento de los filtros adaptivos. El problema del filtrado adaptivo involucra dos procedimientos

de estimación: la estimación de la salida del filtro deseada y , la estimación de los coeficientes requeridos a fin de conseguir el primer objetivo.

El combinador lineal adaptivo constituye un estimador del tipo no recursivo; su estructura se muestra en la figura 2.1, donde el estimado y_j (señal de salida del combinador lineal) se define en términos de una combinación lineal de la señal de entrada X_j :

$$y_j = w_1x_{1j} + w_2x_{2j} + \dots + w_nx_{nj} \quad (2.1)$$

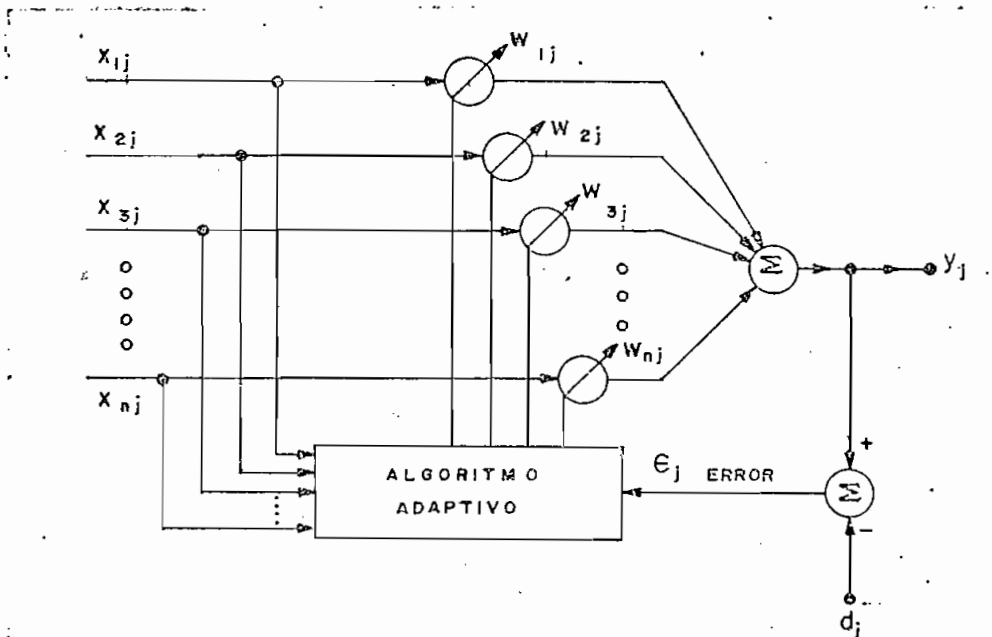


Fig. 2.1 Diagrama esquemático del combinador lineal adaptivo.

El vector señal de entrada X_j se define como¹:

$$X_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{bmatrix} \quad (2.2a)$$

Se asume que las componentes de la señal de entrada, aparecen simultáneamente sobre todas las líneas de entrada, a instantes discretos de tiempo indexados por el subíndice j . Los coeficientes de ponderación o factores de multiplicación w_1, w_2, \dots, w_n son ajustables, el vector de coeficientes de ponderación se define de la siguiente manera:

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad (2.2b)$$

De acuerdo a la ecuación (2.1), la señal de salida y_j es igual al producto interno de X_j^T y W :

$$y_j = X_j^T W = W^T X_j \quad (2.3)$$

donde W^T representa la matriz transpuesta de W . En general se utilizará el superíndice T para denotar que se trata de una matriz o vector transpuesto.

¹ En general se usará notación en mayúsculas para identificar vectores y matrices; y, notación en minúsculas para escalares.

Se define la señal de error ϵ_j , como la diferencia entre la señal d_j (la cual se provee externamente y se suele denominar como la señal deseada actual) y la salida del combinador lineal y_j .

$$\epsilon_j = d_j - y_j$$

$$\epsilon_j = d_j - X_j^T W = d_j - W^T X_j \quad (2.4)$$

2.1.2 EL ALGORITMO ADAPTIVO LMS

El algoritmo adaptivo tiene como finalidad ajustar el valor de los coeficientes de ponderación del combinador lineal a fin de que se minimice el error cuadrático medio (valor esperado del cuadrado de la señal de error); se trata entonces de encontrar un conjunto de coeficientes de ponderación, tales que produzcan una señal a la salida del sistema que sea igual o tan cercana como sea posible a la respuesta deseada, de modo que el error que se genere sea el mínimo.

Si asumimos que la señal de entrada al filtro y la respuesta deseada son estadísticamente estacionarias (lo cual se cumple en el contexto del estudio y aplicación dada a la presente tesis) y que los coeficientes de ponderación son fijos, se puede obtener una expresión general para el error cuadrático medio como una función de los valores de los coeficientes de ponderación.

Así, elevando al cuadrado la ecuación (2.4) se obtiene una expresión para el error cuadrático dado por:

$$\epsilon_j^2 = d_j^2 - 2d_j X_j^T W + W^T X_j X_j^T W \quad (2.5)$$

El error cuadrático medio será entonces el valor esperado del error al cuadrado¹.

$$E[\epsilon_j^2] = E[d_j^2] - 2E[d_j X_j^T]W + W^T E[X_j X_j^T]W \quad (2.6)$$

Definiendo el vector R_{xd} como la correlación cruzada entre la respuesta deseada d_j (un escalar) y el vector de datos de entrada X_j , y la matriz R_{xx} como la autocorrelación de los datos de entrada, se tiene:

$$R_{xd} = E[d_j X_j] = E \begin{bmatrix} d_j X_{1j} \\ d_j X_{2j} \\ \vdots \\ d_j X_{nj} \end{bmatrix} \quad (2.7)$$

$$R_{xx} = E[X_j X_j^T] = E \begin{bmatrix} X_{1j}X_{1j} & X_{1j}X_{2j} & X_{1j}X_{3j} & \dots \\ X_{2j}X_{1j} & X_{2j}X_{2j} & X_{2j}X_{3j} & \dots \\ X_{3j}X_{1j} & X_{3j}X_{2j} & X_{3j}X_{3j} & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & X_{nj}X_{nj} \end{bmatrix} \quad (2.8)$$

La matriz R_{xx} es simétrica, definida positiva y en raras excepciones semidefinida positiva. Se puede entonces expresar el error cuadrático medio como:

$$E[\epsilon_j^2] = E[d_j^2] - 2R_{xd}^T W + W^T R_{xx} W \quad (2.9)$$

El error cuadrático medio es por tanto una función cuadrática del vector de coeficientes de ponderación ^W en cualquier tiempo particular, cuya representación tendrá la forma de una superficie hiperparaboidal cóncava. El hecho

¹ E[...] representa el valor esperado.

de ajustar los valores de los coeficientes de ponderación para minimizar el error involucra descender a lo largo de dicha superficie con la finalidad de obtener el punto mínimo de la misma, para tal propósito por lo general se utiliza métodos del gradiente.

Para los casos no estacionarios, el proceso adaptivo debe rastrear el fondo de la superficie, sin embargo, tanto el punto mínimo, la orientación como la curvatura de dicha superficie de error podrían estar cambiando todo el tiempo, como se muestra en la figura 2.2. Para los casos estacionarios, como el que se estudiará en la presente tesis, los estadísticos R_{xx} y R_{xd} son constantes, y de acuerdo con la ecuación (2.9) se necesita considerar sólo una superficie de error.

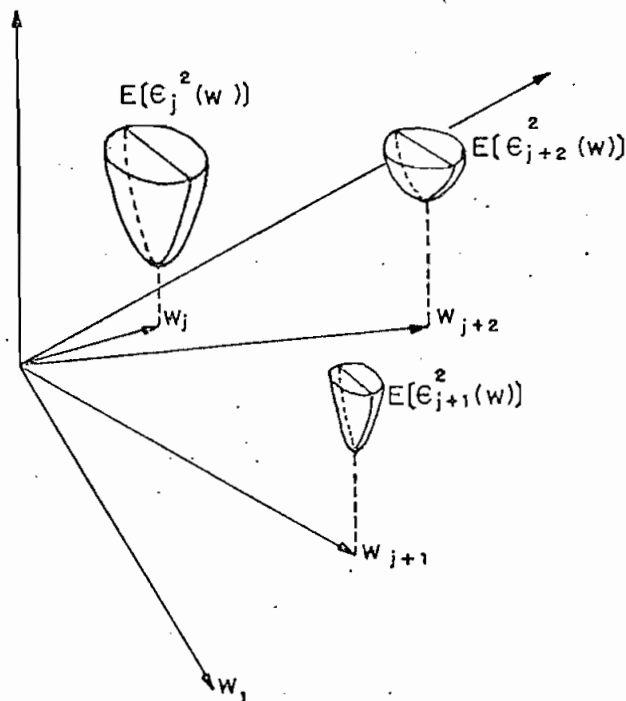


Fig. 2.2 Superficie de error variando en el tiempo, para un proceso no estacionario.

El gradiente en cualquier punto de la superficie de comportamiento se puede obtener diferenciando la función de error cuadrático medio mse (mean square error), ecuación (2.9), con respecto al vector de coeficientes de ponderación.

$$\nabla_J = \frac{\delta E[\epsilon_J^2]}{\delta W}$$

sustituyendo en esta expresión, la fórmula para el error cuadrático medio, dada por (2.9), se tiene:

$$\nabla_J = \frac{\delta}{\delta W} \left[E[d_J^2] - (2 R_{xcd}^T W) + (W^T R_{xxx} W) \right]$$

$$\nabla_J = -2 \frac{\delta}{\delta W} (R_{xcd}^T W) + \frac{\delta}{\delta W} (W^T R_{xxx} W) \quad (2.10)$$

pues $\frac{\delta}{\delta W} [d_J^2] = 0$, ya que d_J es una señal provista externamente y por tanto independiente de W . La derivada de cada uno de los productos de la ecuación (2.10) viene dada por:

$$\frac{\delta}{\delta W} (R_{xcd}^T W) = R_{xcd} \quad (2.11a)$$

$$\frac{\delta}{\delta W} (W^T R_{xxx} W) = 2 R_{xxx} W \quad (2.11b)$$

De modo que el gradiente del error cuadrático medio es igual a:

$$\nabla J = -2R_{xd} + 2R_{xx} W \quad (2.12)$$

Igualando a cero el gradiente de la función de error cuadrático medio, se obtiene el vector de coeficientes de ponderación óptimo W^* , al cual generalmente se le conoce como el vector de coeficientes de ponderación de Wiener:

$$-2R_{xd} + 2R_{xx} W^* = 0$$

entonces

$$W^* = R_{xx}^{-1} R_{xd} \quad (2.13)$$

El vector de coeficientes de ponderación óptimo W^* nos permite obtener el error cuadrático medio mínimo.

Una forma de encontrar el vector de coeficientes de ponderación óptimo es resolviendo la ecuación (2.13), tal solución es generalmente correcta, pero presenta serios problemas computacionales cuando el número de coeficientes de ponderación n es grande y cuando la velocidad de los datos es alta. Adicionalmente a la necesidad de invertir una matriz $n \times n$, este método requiere de muchas medidas de autocorrelación y correlación cruzada a fin de obtener los elementos de R_{xx} y R_{xd} . Además debido a que es necesario una muestra estadística infinita para estimar exactamente los elementos de las matrices de correlación, lo cual en la práctica no es procedente, la solución de la ecuación (2.13) no sería perfecta.

El algoritmo LMS constituye un método que nos permite obtener una solución aproximada a la ecuación (2.13), su exactitud está limitada por el tamaño de la muestra estadística, puesto que de acuerdo a este método se determinan los valores del vector de coeficientes de ponderación basándose en mediciones en tiempo finito de la señal de datos de entrada. Este método no requiere mediciones explí-

citadas de las funciones de correlación ni inversión de matrices.

El algoritmo LMS es una implementación del método de la pendiente más pronunciada. Este es un procedimiento iterativo en el cual la actualización del vector de coeficientes, W_{j+1} , se realiza en base al vector de coeficientes de ponderación anterior, W_j , más un término proporcional al gradiente negativo, esto es:

$$W_{j+1} = W_j - \mu \nabla J \quad (2.14)$$

La constante escalar μ es el factor que controla la estabilidad y la velocidad de convergencia.

El procedimiento iterativo descrito requiere conocer el gradiente exacto (∇J) de la función del error cuadrático medio, en la práctica dicho gradiente no se conoce a priori y necesita estimarse a partir de los datos. El algoritmo LMS estima un gradiente instantáneo en una forma burda pero eficiente; esto lo hace asumiendo que ϵ_j^2 , el cuadrado de una muestra de error simple, es un estimado del error cuadrático medio, es decir $E[\epsilon_j^2] = \epsilon_j^2$; y luego diferenciando ϵ_j^2 con respecto a W .

Entonces las relaciones entre el gradiente verdadero y el estimado vendrán dadas por las siguientes expresiones:

Gradiente verdadero :

$$\nabla J = \begin{bmatrix} \frac{\delta E[\epsilon_j^2]}{\delta w_1} \\ \vdots \\ \frac{\delta E[\epsilon_j^2]}{\delta w_n} \end{bmatrix} \quad W = W_j \quad (2.15a)$$

El gradiente estimado será entonces:

$$\hat{\nabla}_J = \begin{bmatrix} \frac{\delta[\epsilon_J^2]}{\delta w_1} \\ \vdots \\ \frac{\delta[\epsilon_J^2]}{\delta w_n} \end{bmatrix} \quad W=W_J \quad (2.15b)$$

donde derivando uno a uno los elementos del vector gradiente estimado se tiene:

$$\begin{aligned} \frac{\delta[\epsilon_J^2]}{\delta w_1} &= 2\epsilon_J \frac{\delta[\epsilon_J]}{\delta w_1} \\ \vdots & \\ \frac{\delta[\epsilon_J^2]}{\delta w_n} &= 2\epsilon_J \frac{\delta[\epsilon_J]}{\delta w_n} \end{aligned}$$

de manera que la expresión para el vector gradiente estimado se reduce a:

$$\hat{\nabla}_J = 2\epsilon_J \begin{bmatrix} \frac{\delta[\epsilon_J]}{\delta w_1} \\ \vdots \\ \frac{\delta[\epsilon_J]}{\delta w_n} \end{bmatrix} \quad W=W_J = 2\epsilon_J \frac{\delta}{\delta W} [\epsilon_J] \quad (2.15c)$$

Las componentes del gradiente estimado están relacionadas con las derivadas parciales del error instantáneo con respecto a las componentes del vector de coeficientes de ponderación, las mismas que se las puede obtener diferenciando la ecuación (2.4),

$$\frac{\delta}{\delta W} [\epsilon_j] = \frac{\delta}{\delta W} [d_j - X_j^T W] = -X_j \quad (2.16)$$

De modo que el gradiente estimado de acuerdo a la ecuación (2.15c) vendrá dado por:

$$\begin{aligned} \hat{\nabla}_j &= 2 \epsilon_j [-X_j] \\ \hat{\nabla}_j &= -2 \epsilon_j X_j \end{aligned} \quad (2.17)$$

El gradiente estimado en la ecuación (2.17) es exacto, como se demuestra mediante el siguiente argumento. Para un vector de coeficientes de ponderación dado, el valor esperado del gradiente estimado será:

$$\begin{aligned} E[\hat{\nabla}_j] &= E[-2\epsilon_j X_j] \\ &= -2 E[\{d_j - X_j^T W\} X_j] \end{aligned}$$

y puesto que se asume que los coeficientes de ponderación son fijos:

$$\begin{aligned} E[\hat{\nabla}_j] &= -2 (E[d_j X_j] - E[X_j X_j^T] W) \\ &= -2 (R_{xd} - R_{xx} W) \\ &= -2 R_{xd} + 2 R_{xx} W \end{aligned} \quad (2.18)$$

Comparando la ecuación (2.18) con la ecuación (2.12) se observa que :

$$E[\hat{\nabla}_j] = \nabla_j$$

Es decir, para un vector de coeficientes de ponderación dado, el valor esperado del gradiente estimado es igual al gradiente verdadero.

Sustituyendo el gradiente estimado por el gradiente verdadero en la ecuación (2.14), se tiene una expresión para la actualización del vector de coeficientes de ponderación de acuerdo con el algoritmo LMS de Wiener-Hopf:

$$W_{j+1} = W_j + 2 \mu \epsilon_j X_j \quad (2.19)$$

De acuerdo a esta última expresión la actualización del vector de coeficientes de ponderación W_{j+1} se realiza añadiendo al vector de coeficientes de ponderación actual, W_j , el vector de entrada escalado por el valor de la señal de error.

El algoritmo LMS está representado en realidad por la ecuación (2.19), la cual se usa directamente como una fórmula de adaptación del vector de coeficientes de ponderación.

2.1.3 EL FILTRO ADAPTIVO LMS.

Es posible implementar el combinador lineal adaptivo anteriormente visto en base a una línea de retardos, para formar el filtro adaptivo como el de la figura 2.3, la cual muestra en detalle la estructura del filtro, incorporando el algoritmo LMS.

Debido a la estructura de la línea de retardos, el vector de señal de entrada tendrá la forma:

$$X_j = \begin{bmatrix} x_j \\ x_{j-1} \\ \vdots \\ x_{j-n+1} \end{bmatrix} \quad (2.20)$$

Donde las componentes del vector de datos de entrada X_j , esto es: $x_j, x_{j-1}, \dots, x_{j-n+1}$, son versiones retardadas de la señal de entrada x_j .

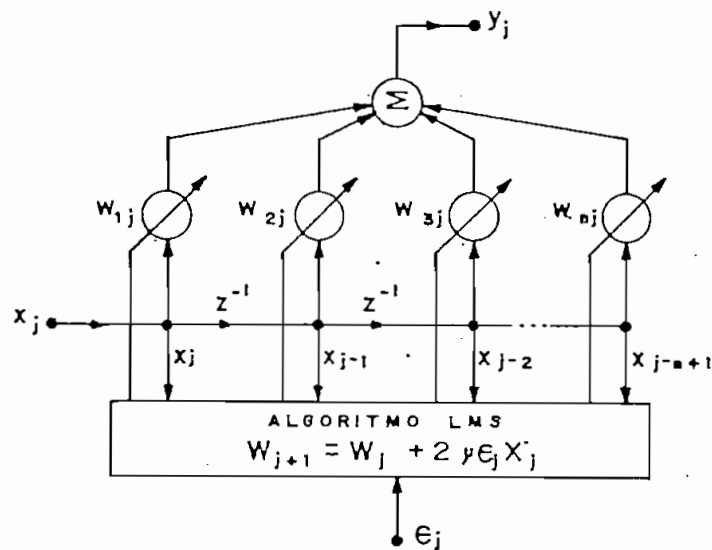


Fig. 2.3 Diagrama de bloques de un filtro adaptivo LMS.

El Filtro Adaptivo LMS como un cancelador de interferencias de banda ancha en una señal periódica, sin fuente de referencia externa.

Partiendo del filtro adaptivo de la figura 2.3 y tomando en consideración el principio de cancelación de ruido establecido en la sección 1.3.2, es posible implementar un sistema capaz de eliminar una interferencia de banda ancha

presente en una señal periódica, donde la señal de referencia se la obtiene como una versión retardada de la señal de entrada principal, y en el cual se utiliza el algoritmo adaptivo LMS para la actualización de los coeficientes del filtro. Tal sistema de cancelación adaptiva de ruido se muestra en la figura 2.4.

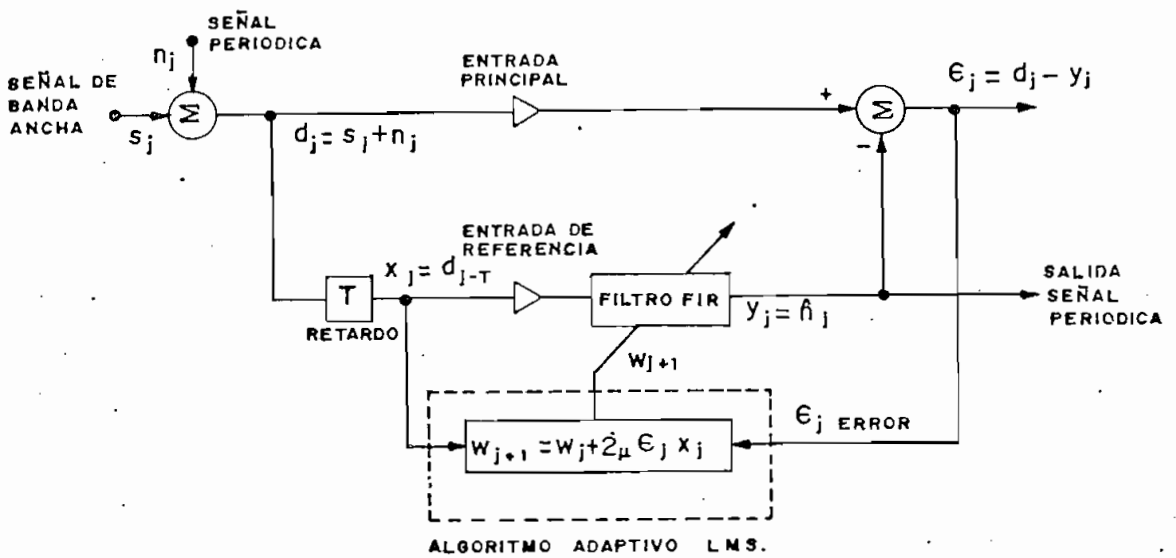


Fig. 2.4 Sistema de cancelación de interferencia de banda ancha en una señal periódica, mediante un filtro adaptivo LMS.

Para el diseño de un filtro como el anteriormente descrito se deben tomar en consideración dos aspectos fundamentales, a saber:

1. la longitud del retardo entre una muestra y su sucesiva debe ser al menos tan pequeña como el recíproco de dos veces el ancho de banda de la señal, de acuerdo con el teorema del muestreo.

2. la longitud total en tiempo real de la línea de retardo, se determina por el recíproco de la resolución en frecuencia deseada para el filtro. De acuerdo a esto el número de coeficientes de ponderación que se requieran es generalmente igual al doble de la razón del ancho de banda de la señal total a la resolución en frecuencia del filtro.

2.2 ANALISIS DE CONVERGENCIA

2.2.1 CONVERGENCIA DEL VECTOR DE COEFICIENTES DE PONDERACION

Cuando los vectores de entrada son no correlacionados todo el tiempo, el valor esperado del vector de coeficientes de ponderación, $E[W_j]$, converge al vector de coeficientes de ponderación de Wiener, W^* , dado por la ecuación (2.13). Cuando los vectores de entrada están correlacionados, una correlación extremadamente alta y una adaptación rápida puede provocar que el vector de coeficientes de ponderación converja a un vector algo diferente del dado por la solución de Wiener, experiencias prácticas han demostrado sin embargo que este efecto es por lo general insignificante.

A fin de demostrar que el valor esperado del vector de coeficientes de ponderación converge al vector de coeficientes de Wiener, se asume que el tiempo entre iteraciones sucesivas en el algoritmo LMS es suficientemente grande como para que las muestras de los vectores de entrada X_j y X_{j+1} sean no correlacionadas, ésta consideración es común en el campo de las aproximaciones estadísticas.

De acuerdo con la ecuación (2.9), el vector de coeficientes de ponderación W_j es función solamente de los vectores

$$(2.9) \quad E[e_j^2] = E[d_j^2] - 2R_{xd}^T W_j + W_j^T R_{xx} W_j$$

de entrada: $X_{j-1}, X_{j-2}, \dots, X_1$, y puesto que los vectores de entradas sucesivas son no correlacionados, W_j es independiente de X_j . Para procesos de entrada estacionarios que satisfacen esta condición, el valor esperado $E[W_j]$ del vector de coeficientes de ponderación luego de un gran número de iteraciones se puede demostrar que converge a la solución de Wiener dada por (2.13).

Tomando el valor esperado de ambos lados de la ecuación (2.14), se obtiene una ecuación de diferencias en el valor esperado del vector de coeficientes de ponderación.

$$\begin{aligned} E[W_{j+1}] &= E[W_j] + 2\mu E[X_j \{d_j - X_j^T W_j\}] \\ &= E[W_j] + 2\mu \{E[d_j X_j] - E[X_j X_j^T] E[W_j]\} \end{aligned}$$

puesto que W_j es independiente de X_j

$$\begin{aligned} E[W_{j+1}] &= E[W_j] + 2\mu \{R_{xd} - R_{xx} E[W_j]\} \\ &= \{E[W_j] - 2\mu R_{xx} E[W_j]\} + 2\mu R_{xd} \\ &= [I - 2\mu R_{xx}] E[W_j] + 2\mu R_{xd} \quad (2.21) \end{aligned}$$

donde I es la matriz identidad. Con un vector de coeficientes de ponderación inicial W_0 , luego de la primera iteración la ecuación (2.21) quedará:

primera iteración:

$$E[W_1] = [I - 2\mu R_{xx}] E[W_0] + 2\mu R_{xd}$$

si realizamos una nueva iteración se tendrá:

segunda iteración:

$$E[W_2] = [I - 2\mu R_{xx}] E[W_1] + 2\mu R_{xd}$$

sustituyendo $E[W_1]$ en esta expresión

$$E[W_2] = \{I - 2\mu R_{xx}\} \{[I - 2\mu R_{xx}] E[W_0] + 2\mu R_{xd}\} + 2\mu R_{xd}$$

$$E[W_2] = [I - 2\mu R_{xx}]^2 E[W_0] + [I - 2\mu R_{xx}] 2\mu R_{xd} + 2\mu R_{xd}$$

$$E[W_2] = [I - 2\mu R_{xx}]^2 E[W_0] + 2\mu \{[I - 2\mu R_{xx}] R_{xd} + R_{xd}\}$$

la cual puede expresarse como:

$$E[W_2] = [I - 2\mu R_{xx}]^2 E[W_0] + 2\mu \{[I - 2\mu R_{xx}]^1 R_{xd} + [I - 2\mu R_{xx}]^0 R_{xd}\}$$

al realizar $j+1$ iteraciones:

$$E[W_{j+1}] = [I - 2\mu R_{xx}]^{j+1} E[W_0] + 2\mu \sum_{i=0}^j [I - 2\mu R_{xx}]^i R_{xd}$$

(2.22)

Puesto que la matriz de correlación de los datos de entrada R_{xx} es simétrica y definida positiva, es posible usar una matriz de transformación de similaridad Q y representarla como:

$$R_{xx} = QQ^{-1}$$

$$R_{xx} = QQ^T \quad (2.23)$$

donde Q es una matriz ortogonal y, por tanto se cumplirá que $Q^{-1} = Q^T$, es decir la matriz inversa es igual a la transpuesta. Q constituye la matriz diagonal de los valores propios, todos ellos positivos debido a que R_{xx} es una

matriz definida positiva, en consecuencia ζ tendrá la forma:

$$\zeta = \begin{bmatrix} \zeta_1 & 0 & \dots & 0 \\ 0 & \zeta_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & & \zeta_n \end{bmatrix} \quad (2.24)$$

La ecuación (2.22) podrá expresarse ahora en forma diagonal como:

$$E[W_{j+1}] = [I - 2\mu\zeta Q^{-1}]^{j+1} E[W_0] + 2\mu \sum_{i=0}^j [I - 2\mu\zeta Q^{-1}]^i R_{xd}$$

la matriz identidad en esta ecuación se puede sustituir por la expresión $I = QIQ^{-1}$, por tanto

$$E[W_{j+1}] = [QIQ^{-1} - 2\mu\zeta Q^{-1}]^{j+1} E[W_0] + 2\mu \sum_{i=0}^j [QIQ^{-1} - 2\mu\zeta Q^{-1}]^i R_{xd}$$

Consideremos el término $[QIQ^{-1} - 2\mu\zeta Q^{-1}]^{j+1}$, el mismo que será igual a:

$$\begin{aligned} [QIQ^{-1} - 2\mu\zeta Q^{-1}]^{j+1} &= \{Q[I - 2\mu\zeta]Q^{-1}\}^{j+1} \\ &= Q[I - 2\mu\zeta]Q^{-1} Q[I - 2\mu\zeta]Q^{-1} \dots Q[I - 2\mu\zeta]Q^{-1} \\ &= Q [I - 2\mu\zeta]^{j+1} Q^{-1} \end{aligned}$$

tomando en consideración esta igualdad, la ecuación para el valor esperado del vector de coeficientes de ponderación quedará de la forma:

$$E[W_{j+1}] = Q [I - 2\mu C]^{j+1} Q^{-1} E[W_0] + 2\mu Q \left\{ \sum_{i=0}^j [I - 2\mu C]^i Q^{-1} R_{xd} \right\} \quad (2.25)$$

La matriz $[I - 2\mu C]$ es una matriz diagonal, donde todos los términos de la diagonal $(1 - 2\mu c_i)$ deben ser menores que la unidad, como se demostrará posteriormente, de modo que se tendrá:

$$\lim_{j \rightarrow \infty} [I - 2\mu C]^{j+1} \rightarrow 0$$

El primer término de la ecuación (2.25) disminuye a medida que el número de iteraciones aumenta, de manera que tiende a desaparecer; el segundo término de la ecuación (2.25) generalmente converge a un límite diferente de cero, así pues, el factor que incluye el sumatorio convergirá a:

$$\lim_{j \rightarrow \infty} \sum_{i=0}^j [I - 2\mu C]^i = \frac{1}{2\mu C} = \frac{C^{-1}}{2\mu} \quad (2.26)$$

donde para determinar el límite del sumatorio se utiliza la fórmula para la suma de una serie geométrica, la misma que señala:

$$\sum_{i=0}^{\infty} [1 - 2\mu e_p]^i = \frac{1}{1 - (1 - 2\mu e_p)} = \frac{1}{2\mu e_p} \quad (2.27)$$

Por tanto en el límite la ecuación (2.25) tendrá un valor dado por:

$$\lim_{j \rightarrow \infty} E[W_{j+1}] = 2\mu Q \frac{C^{-1}}{2\mu} Q^{-1} R_{xd} = Q C^{-1} Q^{-1} R_{xd}$$

de acuerdo a la ecuación (2.23), $R_{xx} = Q\bar{C}Q^{-1}$, de modo que:
 $R_{xx}^{-1} = [Q\bar{C}Q^{-1}]^{-1} = Q\bar{C}^{-1}Q^{-1}$ sustituyendo esta igualdad en
la última expresión, se tiene:

$$\lim_{j \rightarrow \infty} E[W_{j+1}] = R_{xx}^{-1} R_{xd} \quad (2.28)$$

Comparando este resultado con el de la ecuación (2.13), se ve que cuando el número de iteraciones se incrementa sin límite, el valor esperado del vector de coeficientes de ponderación converge a la solución de Wiener.

La convergencia del valor esperado del vector de coeficientes de ponderación a la solución de Wiener, se garantiza únicamente si la constante de proporcionalidad " μ " está dentro de ciertos límites. Puesto que todos los términos de la diagonal de la matriz $[I - 2\mu\bar{C}]$ deben tener una magnitud menor que la unidad; y, considerando que los valores propios de \bar{C} son positivos, los límites de μ estarán dados por:

$$\begin{aligned} & |1 - 2\mu\bar{c}_{\max}| < 1 \\ \text{ó} \\ & 0 < \mu < 1/\bar{c}_{\max} \end{aligned} \quad (2.29)$$

donde \bar{c}_{\max} es el máximo valor propio de R_{xx} . La ecuación (2.29) es una condición necesaria y suficiente para la convergencia del algoritmo LMS. La condición de convergencia sobre μ puede relacionarse con la traza de matriz de autocorrelación R_{xx} como sigue.

Considerando que¹

$$\varphi_{\text{máx}} < \text{traza}[R_{xx}] \quad (2.30)$$

esta condición se deriva del hecho que para matrices definidas positivas se tiene:

$$\text{traza}[R_{xx}] = \sum_{i=1}^n \varphi_i > \varphi_{\text{máx}} \quad (2.31)$$

Por lo tanto, es posible obtener una convergencia satisfactoria haciendo que se cumpla la siguiente desigualdad:

$$0 < \mu < \frac{1}{\text{traza}[R_{xx}]} \quad (2.32)$$

En la práctica cuando se desea una adaptación precisa, debe trabajarse con una constante de convergencia μ mucho menor, aunque esto implica que el proceso de adaptación se torne más lento, de manera que debe elegirse μ tal que:

$$0 < \mu \ll \frac{1}{\text{traza}[R_{xx}]} \quad (2.33)$$

La ventaja de usar la ecuación (2.32) es que la traza de la matriz de autocorrelación R_{xx} puede fácilmente calcularse de acuerdo a²:

$$\text{traza}[R_{xx}] = \sum_{i=1}^n E[(x_i)^2] \quad (2.34)$$

-
- 1 La traza de una matriz constituye la suma de todos los elementos de la diagonal de dicha matriz.
 - 2 Recuérdese que n es el orden del filtro, y por lo tanto la matriz de autocorrelación es de orden $n \times n$, en consecuencia existirán n elementos en la diagonal.

de esta manera se obvia el cálculo de los valores propios de la matriz de autocorrelación de las muestras de entrada R_{xx} , ya que estos en la práctica rara vez se los conoce, y por tanto la ecuación (2.29) no es siempre fácil de aplicar. Puesto que es posible calcular la traza $[R_{xx}]$ de manera simple aplicando la ecuación (2.34), y considerando que $\text{traza}[R_{xx}] > c_{\text{máx}}$, al cumplirse la ecuación (2.32) se garantiza la convergencia.

2.2.2 CONSTANTES DE TIEMPO Y CURVA DE APRENDIZAJE

A fin de analizar la estabilidad y las constantes de tiempo que determinan la velocidad de convergencia del algoritmo LMS, se expresa el fenómeno transitorio mientras se produce el proceso de adaptación, en términos de las coordenadas normales del sistema.

Para el efecto consideremos nuevamente la expresión para el error cuadrático medio dado por la ecuación (2.9)

$$E[\epsilon_j^2] = E[d_j^2] - 2 \frac{R_{xx}^T W}{R_{xd}^T} + W^T R_{xx} W$$

el error cuadrático medio mínimo se obtiene cuando en la ecuación anterior en lugar de W se utiliza el vector de coeficientes de ponderación óptimo W^* , esto es:

$$E[\epsilon_j^2]_{\text{mín}} = E[d_j^2] - 2R_{xd}^T W^* + (W^*)^T R_{xx} W^* \quad (2.35)$$

llamemos a la expresión $E[\epsilon_j^2]$ como Γ ; y, sustituyendo en la ecuación (2.35) a W^* por el valor dado en (2.13) se tendrá:

$$\begin{aligned} \Gamma_{\text{mín}} &= E[d_j^2] - 2R_{xd}^T (R_{xx}^{-1} R_{xd}) + (R_{xx}^{-1} R_{xd})^T R_{xx} (R_{xx}^{-1} R_{xd}) \\ &= E[d_j^2] - 2R_{xd}^T R_{xx}^{-1} R_{xd} + R_{xd}^T (R_{xx}^{-1})^T R_{xx} R_{xx}^{-1} R_{xd} \end{aligned}$$

si se considera en esta última expresión que $(R_{xxx}^{-1})^T = R_{xxx}^{-1}$, de modo que:

$$\begin{aligned} \Gamma_{\min} &= E[d_j^2] - 2 R_{xd}^T R_{xxx}^{-1} R_{xd} + R_{xd}^T R_{xxx}^{-1} R_{xd} \\ &= E[d_j^2] - R_{xd}^T R_{xxx}^{-1} R_{xd} \\ &= E[d_j^2] - R_{xd}^T W^* \end{aligned} \quad (2.36)$$

sustituyendo (2.36) en la ecuación (2.9), obtenemos una fórmula más útil para el error cuadrático medio; así:

$$\Gamma = E[\epsilon_j^2] = \Gamma_{\min} + R_{xd}^T W^* - 2R_{xd}^T W + W^T R_{xxx} W$$

considerando en esta última expresión que $R_{xd}^T W = W^T R_{xd}$:

$$\Gamma = \Gamma_{\min} + R_{xd}^T W^* - W^T R_{xd} - R_{xd}^T W + W^T R_{xxx} W$$

a partir de (2.13) se tiene $R_{xd} = R_{xxx} W^*$, y por tanto de ésta ecuación se desprende que $R_{xd}^T = W^{*T} R_{xxx}$ (puesto que R_{xxx} es simétrica). En consecuencia la expresión para Γ quedará de la forma:

$$\begin{aligned} \Gamma &= \Gamma_{\min} + W^{*T} R_{xxx} W^* - W^T R_{xd} W^* - W^{*T} R_{xx} W + W^T R_{xxx} W \\ &= \Gamma_{\min} + (W^{*T} R_{xxx}) W^* - W^T (R_{xxx} W^*) - (W^{*T} R_{xxx}) W + \\ &\quad W^T R_{xxx} W \end{aligned}$$

reordenando términos

$$\begin{aligned} \Gamma &= \Gamma_{\min} + W^T R_{xxx} W - W^T R_{xxx} W^* - W^{*T} R_{xxx} W + W^{*T} R_{xxx} W^* \\ &= \Gamma_{\min} + W^T R_{xxx} (W - W^*) - W^{*T} R_{xxx} (W - W^*) \\ &= \Gamma_{\min} + (W^T R_{xxx} - W^{*T} R_{xxx}) (W - W^*) \end{aligned}$$

sacando como factor común a R_{xxx}

$$\Gamma = \Gamma_{\min} + (W^T - W^{*T}) R_{xx} (W - W^*)$$

$$\Gamma = \Gamma_{\min} + (W - W^*)^T R_{xx} (W - W^*) \quad (2.37)$$

Esta última ecuación constituye una expresión para el error cuadrático medio que permitirá dibujar la curva de aprendizaje, la cual constituye un método útil que permite analizar el progreso del proceso adaptivo. Cuando se usa el error cuadrático medio como un criterio para evaluar el comportamiento del proceso, es posible grafizarlo para cada etapa del proceso de aprendizaje como una función del número de ciclos de adaptación.

Se define V como la diferencia entre W y la solución de Wiener W^* , esto es:

$$V \triangleq (W - W^*) \quad (2.38)$$

la ecuación (2.37) quedará entonces de la forma:

$$\Gamma = \Gamma_{\min} + V^T R_{xx} V \quad (2.39)$$

diferenciando la ecuación (2.39) conduce a otra forma para el gradiente:

$$\nabla = 2 R_{xx} V \quad (2.40)$$

De acuerdo con la ecuación (2.23) se tenía:

$$R_{xx} = Q C Q^{-1} = Q C Q^T \quad (2.41)$$

por tanto la ecuación (2.39), usando esta transformación, se puede expresar como:

$$\Gamma = \Gamma_{\min} + V^T Q C Q^{-1} V \quad (2.42)$$

definiendo una versión transformada de V como:

$$V' \cong Q^{-1}V \quad \text{y} \quad V \cong QV' \quad (2.43)$$

la ecuación (2.42) puede expresarse en forma normal como:

$$\Gamma = \Gamma_{\min} + V'^T Q V' \quad (2.44)$$

pues $V'^T = V^T(Q^{-1})^T$, pero $Q^T = Q^{-1}$, entonces $V'^T = V^T Q$
Las coordenadas primadas son por tanto los ejes principales de la superficie cuadrática. La transformación (2.43) se puede así mismo aplicar al vector de coeficientes de ponderación, de manera que tendremos:

$$W' = Q^{-1}W \quad \text{y} \quad W = QW' \quad (2.45)$$

por otro lado de acuerdo con la ecuación (2.14):

$$W_{j+1} = W_j + \mu(-\nabla_j) \quad (2.46)$$

el gradiente a la j -ésima iteración es ∇_j . Usando las ecuaciones (2.40), (2.41) y (2.43); la ecuación (2.45) se transforma en:

$$\begin{aligned} W_{j+1} &= W_j + \mu(-2 R_{xx} V_j) \\ &= W_j - 2\mu(QCQ^{-1})QV_j' \\ &= W_j - 2\mu Q C V_j' \end{aligned} \quad (2.47)$$

además $V = W - W^*$, en consecuencia a la j -ésima iteración $W_j = V_j + W^*$ y $W_{j+1} = V_{j+1} + W^*$. De modo que la ecuación (2.47) quedará de la forma:

$$V_{j+1} + W^* = V_j + W^* - 2\mu Q C V_j'$$

$$V_{j+1} = V_j - 2\mu Q C V_j' + W^* - W^*$$

$$\begin{aligned} &= QV_j' - 2\mu QCV_j' \\ &= Q [V_j' - 2\mu CV_j'] \end{aligned}$$

si premultiplicamos ambos lados de la ecuación por Q^{-1} , se tiene:

$$Q^{-1}V_{j+1} = V_j' - 2\mu CV_j'$$

de manera que:

$$V_{j+1}' = (I - 2\mu C)V_j'$$

y por lo tanto:

$$V_{j+1}' - (I - 2\mu C)V_j' = 0 \quad (2.48)$$

tal ecuación de diferencias tiene una solución simple en coordenadas primadas, dada por :

$$V_j' = (I - 2\mu C)^j V_0' \quad (2.49)$$

donde V_0' es la condición inicial, esto es:

$$V_0' = W_0' - W^* \quad (2.50)$$

A partir de la ecuación (2.49) se observa que el transitorio en coordenadas primadas será geométrico, la razón geométrica de la p-ésima coordenada es:

$$r_p = (1 - 2\mu c_p) \quad p=1,2,\dots,n \quad (2.51)$$

una envolvente exponencial que se puede ajustar a una secuencia geométrica. Si la unidad básica de tiempo es un ciclo de iteración, la constante de tiempo τ_p se puede determinar así:

$$r_p = \exp \left[- \frac{1}{\tau_p} \right] = 1 - \frac{1}{\tau_p} + \frac{1}{2! \tau_p^2} - \dots \quad (2.52)$$

el caso de interés general es una adaptación lenta, esto es con valores grandes de τ_p , de modo que se cumplirá la siguiente relación:

$$r_p = (1 - 2\mu\zeta_p) \simeq 1 - \frac{1}{\tau_p}$$

de donde es posible obtener una expresión para τ_p como función de ζ_p :

$$\tau_p \simeq \frac{1}{2\mu\zeta_p} \quad p=1,2,\dots,n \quad (2.53)$$

por tanto la ecuación (2.53) da la constante de tiempo del p-ésimo modo, donde ζ_p es el p-ésimo valor propio de la matriz de autocorrelación, R_{xx} , de la señal de entrada.

En el caso especial donde todos los valores propios son iguales, todas las constantes de tiempo serán también iguales, consecuentemente:

$$\tau = \frac{1}{2\mu\zeta}$$

Durante el proceso de adaptación, el error ϵ_j es no estacionario conforme el vector de coeficientes de ponderación tiende al vector de Wiener W^* . De acuerdo con la ecuación (2.44) se tenía:

$$\Gamma_j = \Gamma_{min} + V_j \tau C V_j \quad (2.54)$$

Usando (2.49), pero asumiendo que no existe ruido en el vector de coeficientes de ponderación, la ecuación (2.54) quedará:

$$\Gamma_j = \Gamma_{\min} + V_o'^T [I - 2\mu\zeta]^{jT} \zeta [I - 2\mu\zeta]^j V_o' \quad (2.55)$$

pues $[I - 2\mu\zeta]^j$ sigue siendo una matriz diagonal y en consecuencia $[I - 2\mu\zeta]^{jT} = [I - 2\mu\zeta]^j$. La expresión anterior quedará entonces de la forma:

$$\Gamma_j = \Gamma_{\min} + V_o'^T \zeta [I - 2\mu\zeta]^{2j} V_o' \quad (2.56)$$

a partir de la ecuación (2.55) y tomando en consideración que se pueden aplicar las sustituciones:

$$V_o' = Q^{-1}V_o \quad \text{por tanto} \quad V_o'^T = V_o^T Q$$

$$R_{xx} = Q\zeta Q^{-1} \quad \text{por tanto} \quad \zeta = Q^{-1}R_{xx}Q$$

$$I = Q^{-1}IQ$$

es posible llegar a una expresión para el error cuadrático medio de la forma:

$$\begin{aligned} \Gamma_j &= \Gamma_{\min} + V_o^T Q [I - 2\mu\zeta]^j \zeta [I - 2\mu\zeta]^j Q^{-1} V_o \\ &= \Gamma_{\min} + V_o^T Q [Q^{-1}IQ - 2\mu Q^{-1}R_{xx}Q]^j Q^{-1}R_{xx}Q [Q^{-1}IQ - \\ &\quad - 2\mu Q^{-1}R_{xx}Q]^j Q^{-1} V_o \\ &= \Gamma_{\min} + V_o^T Q Q^{-1} [I - 2\mu R_{xx}]^j Q Q^{-1} R_{xx} Q Q^{-1} [I - 2\mu R_{xx}]^j \\ &\quad Q Q^{-1} V_o \\ &= \Gamma_{\min} + V_o^T [I - 2\mu R_{xx}]^j R_{xx} [I - 2\mu R_{xx}]^j V_o \quad (2.57) \end{aligned}$$

De esta última expresión se observa que, si el proceso adaptivo es convergente, cuando el número de iteraciones se

aumenta considerablemente, en el límite el error cuadrático medio, Γ , converge al error cuadrático medio mínimo, Γ_{\min} , esto es:

$$\lim_{j \rightarrow \infty} \Gamma_j = \Gamma_{\min} \quad (2.58)$$

En los valores de los coeficientes de ponderación se produce un fenómeno de naturaleza decreciente en forma exponencial y, si observamos la ecuación (2.17) de ella se deduce que el error cuadrático medio es una función cuadrática de los valores de los coeficientes de ponderación; de modo que el transitorio en la función del error cuadrático medio debe también ser de naturaleza exponencial.

El error cuadrático medio tendrá una función de naturaleza exponencial que parte del valor Γ_0 (mse inicial) e irá, a medida que el algoritmo LMS converge, tendiendo al mse mínimo (Γ_{\min}). Para el p-ésimo modo se tendrá una razón geométrica de r_p^2 , como se desprende del análisis de la ecuación (2.56); y, una constante de tiempo dada por:

$$\tau_{p(\text{mse})} = \frac{1}{2} \tau_p = \frac{1}{4\mu\zeta_p} \quad (2.59)$$

La razón que justifica lo anteriormente expuesto es que, el cuadrado de una función exponencial es otra exponencial con constante de tiempo igual a la mitad de la constante de tiempo de la primera exponencial.

Cuando todas las constantes de tiempo son iguales, la curva de aprendizaje del error cuadrático medio, es una exponencial pura, con una constante de tiempo dada por:

$$T_{mse} = \frac{\tau}{2} = \frac{1}{4\mu c}$$

Cuando se grafizan las curvas de aprendizaje experimentales, estas son generalmente de la forma de exponenciales ruidosos a causa del ruido inherente presente en el proceso de adaptación; mientras mas lenta es la adaptación, mas pequeña será la amplitud del ruido aparente en la curva de aprendizaje.

2.3 IMPLEMENTACION DE PROGRAMAS

En algoritmo LMS en esencia consta de los siguientes pasos:

1.- Selección de Parámetros:

- Orden del filtro, n
- Número de ratrdos, T
- Número de muestras a procesarse, NUM

El número de retardos que se elija deberá ser de longitud suficiente para garantizar que las componentes de banda ancha de la entrada de referencia sean no correlacionadas con las de la entrada principal. Tal retardo podrá variar entre 1 y n (orden del filtro).

El orden del filtro podrá seleccionarse como máximo igual a 25 debido a limitaciones de memoria del computador, sin embargo éste es un rango suficiente puesto que con valores de n menores se consigue buenos resultados, como lo comprueban los ejemplos que constan en el capítulo IV.

La señal de entrada al filtro se genera en una rutina adicional, la cual entrega muestras discretas de la misma. El

máximo número de muestras que puede manejar el programa es igual a 2000 por razones de limitaciones de memoria del computador, tal número de muestras es suficiente para que el proceso adaptivo converja y se estabilice, y en consecuencia se tengan buenos resultados.

2.- Inicialización:

Deberán inicializarse tanto el vector de datos de entrada, $X(0)$, como el de coeficientes de ponderación, $W(0)$, ambos vectores se los inicializa con un valor de 0, pues se supone que antes de este instante no existe señal de entrada y, el vector de coeficientes del filtro puede inicializarse con cualquier valor.

3.- Conformación de la señal de referencia a partir de la entrada principal.

Disponiendo del archivo de datos y del número de retardos seleccionado, se obtiene la señal de referencia como una versión retardada (T retardos) de la entrada principal.

4.- Calcular el parámetro de convergencia crítico μ_{crit} .

Para el efecto deberá primero calcularse la traza de la matriz de autocorrelación, R_{xx} , la misma que vendrá dada por:

$$R_{xx} = E \begin{bmatrix} (x_j \quad x_j) & (x_j \quad x_{j-1}) & \dots & (x_j \quad x_{j-n+1}) \\ (x_{j-1} \quad x_j) & (x_{j-1} \quad x_{j-1}) & \dots & (x_{j-1} \quad x_{j-n+1}) \\ \cdot & & \dots & \\ \cdot & & \dots & \\ (x_{j-n+1}x_j) & (x_{j-n+1}x_{j-1}) & \dots & (x_{j-n+1}x_{j-n+1}) \end{bmatrix}$$

$$R_{xx} = \begin{bmatrix} E[x_j x_j] & E[x_j x_{j-1}] & \dots & E[x_j x_{j-n+1}] \\ E[x_{j-1} x_j] & E[x_{j-1} x_{j-1}] & \dots & E[x_{j-1} x_{j-n+1}] \\ \vdots & \vdots & \ddots & \vdots \\ E[x_{j-n+1} x_j] & E[x_{j-n+1} x_{j-1}] & \dots & E[x_{j-n+1} x_{j-n+1}] \end{bmatrix}$$

como el proceso es estacionario se tendrá:

$$R_{xx} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(n-1) \\ r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(n-1) & r_{xx}(n-2) & \dots & r_{xx}(0) \end{bmatrix}$$

De modo que la traza de la matriz de autocorrelación vendrá dada por:

$$\text{Traza}[R_{xx}] = n r_{xx}(0)$$

donde

$$\begin{aligned} r_{xx}(0) &= E[x_j x_j] \quad j=1 \dots \text{NUM} \\ &= \frac{1}{\text{NUM}} \sum_{j=1}^{\text{NUM}} x_j^2 \end{aligned}$$

El parámetro de convergencia crítico es el máximo valor de μ con el que el algoritmo aún converge, un valor de μ mayor puede provocar que el algoritmo diverja. Como se estableció en la sección 2.2, el valor de μ deberá estar limitado al rango:

$$0 < \mu < 1/c_{\max}$$

Sin embargo recientes resultados sobre el análisis de las propiedades de convergencia del algoritmo LMS [4] señalan que el parámetro μ debe estar restringido a un intervalo significativamente más pequeño que el dominio comunmente establecido. Así la estabilidad se asegura sí y solamente sí:

$$0 < \mu < 1/3 \rho_{\max}$$

Con este criterio se procedió a determinar el valor de μ crítico, de acuerdo al cual:

$$\mu_{\text{crítico}} = \frac{1}{3 \text{ traza}[R_{xx}]}$$

Lógicamente que con este valor de μ o con un valor cercano, el error en estado estable será significativo, y por tanto la señal a la salida del cancelador contendrá ruido en bajo nivel, de manera que es preferible usar un valor de μ menor, a pesar de que esto implica que la velocidad de convergencia del algoritmo disminuye.

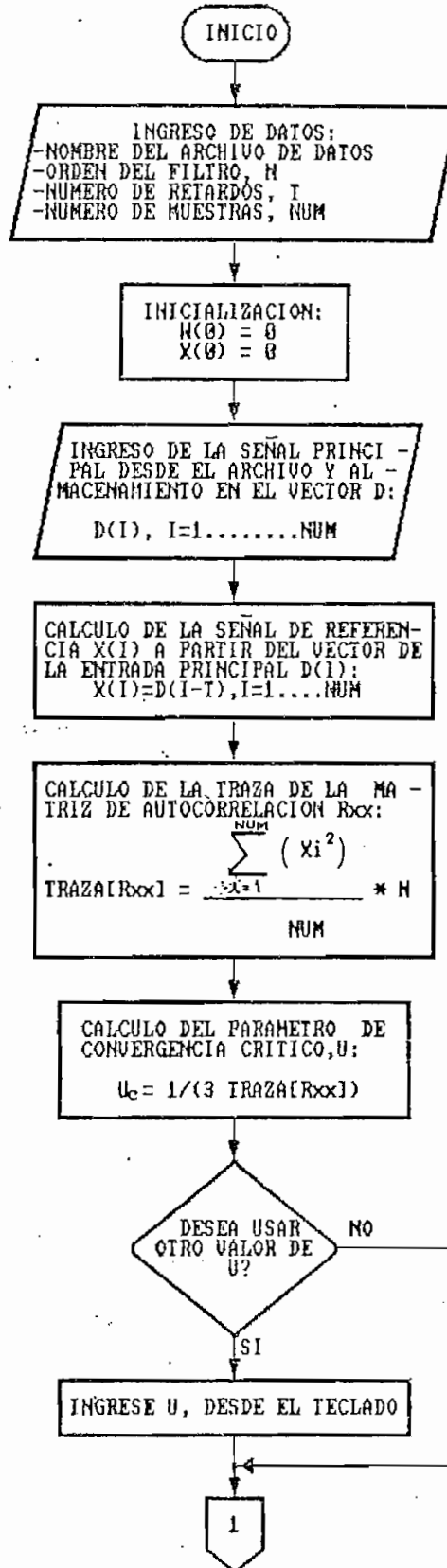
5.- Ejecución del algoritmo:

A cada paso del tiempo y con cada nueva muestra del archivo de datos se deberán realizar los siguientes pasos:

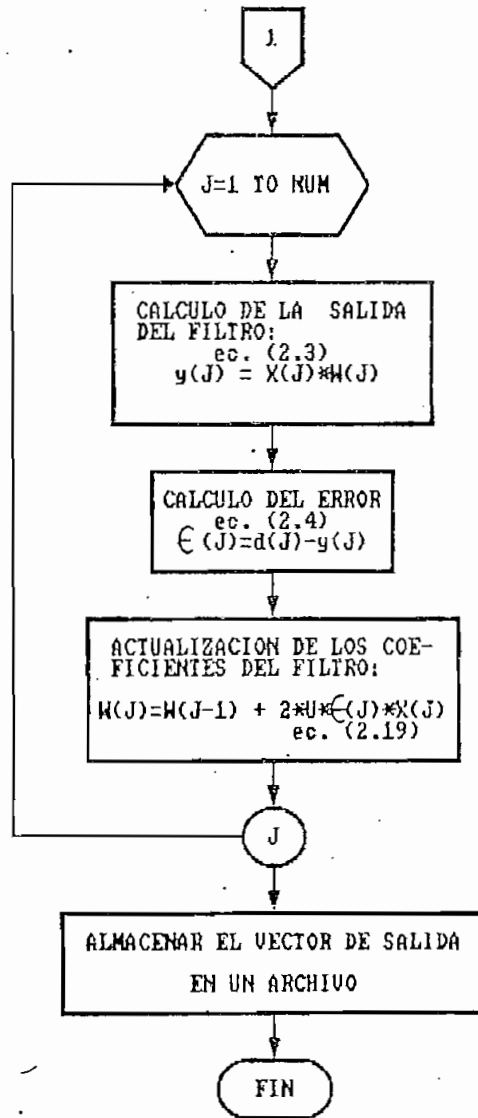
$y_j = X_j^T W$	calcular la salida del filtro.
$e_j = d_j - y_j$	calcular la señal de error.
$W_{j+1} = W_j + 2\mu e_j X_j$	actualizar los coeficientes del filtro.

Los pasos anteriormente descritos se sintetizan en el siguiente diagrama de flujo.

RUTINA DEL ALGORITMO LMS



...CONTINUACION



CAPITULO III

ALGORITMO DE MINIMOS CUADRADOS RECURSIVOS

O

RLS (RECURSIVE-LEAST-SQUARES)

En este capítulo se analizará y desarrollará el algoritmo adaptivo RLS en su forma ponderada exponencialmente, se estudiarán sus propiedades de convergencia y se determinarán tanto sus ventajas como sus desventajas.

Este algoritmo constituye al igual que el algoritmo LMS una ley adaptiva cuyo objetivo es minimizar la potencia de la salida del filtro, la necesidad de minimizar esta función de costo particular es característica de las aplicaciones como la que se da a la presente tesis, esto es la cancelación adaptiva de ruido, donde está involucrado un problema de estimación de mínimos cuadrados, como se manifestó en el capítulo anterior.

El algoritmo RLS ha sido muy utilizado principalmente en el campo de control para identificación de sistemas. A pesar de su comportamiento potencialmente superior, su uso en aplicaciones de procesamiento de señal ha sido muy limitado, debido principalmente a sus altos requerimientos computacionales, aunque en los últimos años se ha renovado el interés por el algoritmo RLS, especialmente en sus versiones rápidas (las cuales son muy eficientes computacionalmente).

3.1 FUNDAMENTO TEORICO

3.1.1 EL ESTIMADOR RECURSIVO

Antes de entrar al análisis del estimador recursivo óptimo, es necesario establecer que el estudio se fundamentará en procesos aleatorios estacionarios; y, por tanto se analizará un modelo para generación de señal de este tipo.

Se derivará posteriormente el estimador óptimo para un filtro recursivo generalizado de primer orden, cuyos resultados dan origen al llamado filtro escalar de Kalman, de forma tal que mediante una transformación directa es posible convertirlo al caso vectorial.

El estimador recursivo óptimo es esencialmente un proceso de estimación paramétrica, el cual se basa en un modelo autorregresivo (AR) para el proceso de generación de señal, de acuerdo al cual asumimos que la señal aleatoria que será estimada se puede modelar como un proceso recursivo de primer orden excitado por una señal de ruido blanco, g_{j-1} , con media cero y varianza σ_g^2 ; por consiguiente la señal se desarrolla en el tiempo de acuerdo a una ecuación dinámica de la forma:

$$d_j = a d_{j-1} + g_{j-1} \quad (3.1)$$

Tal modelo de generación de señal se muestra en la figura 3.1; con su correspondiente modelo de medida, el cual es en esencia un término de ganancia c con ruido blanco aditivo v_j de media cero y varianza σ_v^2 .

El dato observado al instante de muestreo j estará dado por:

$$x_j = c d_j + v_j \quad (3.2)$$

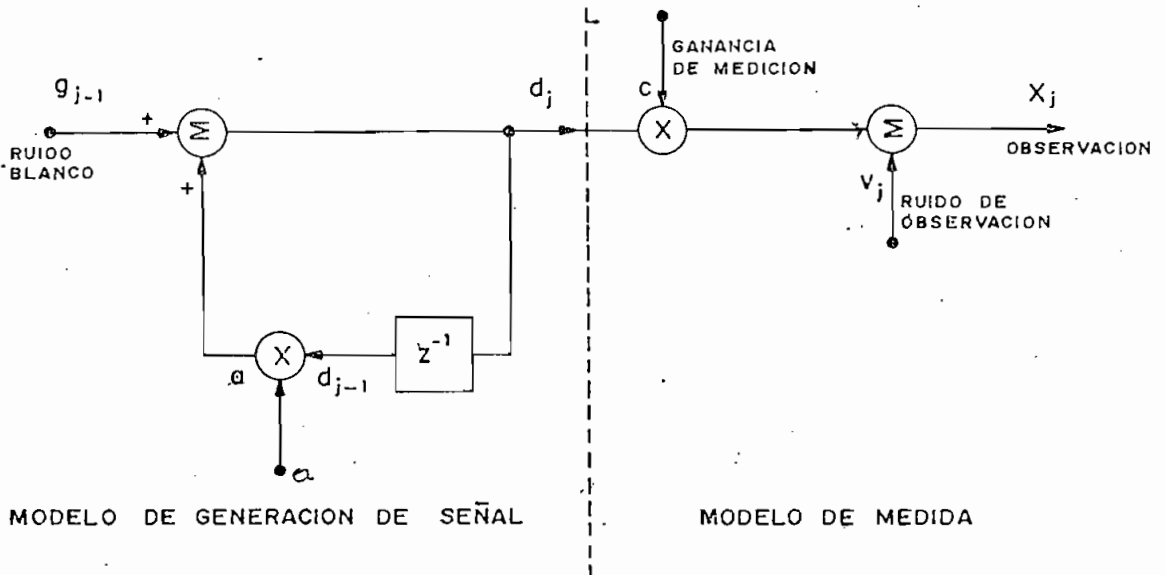


Fig. 3.1 Modelo recursivo de primer orden para generación de señal, con su correspondiente modelo de medición de datos.

El estimador recursivo de primer orden, cuya estructura se muestra en la figura 3.2, genera un estimado en base a una ecuación de la forma:

$$y_j = b_j y_{j-1} + k_j x_j \quad (3.3)$$

donde ambos términos de ganancia en el filtro b_j y k_j están variando en el tiempo. La señal y_j constituye un estimado de d_j . El primer término de la ecuación (3.3) representa el estimado previo ponderado por la ganancia del filtro b_j , el segundo término es la muestra de datos actual ponderada por la otra ganancia del filtro k_j . Nótese que el estimador recursivo realiza un estimado, y_j , de la señal d_j al instante j basándose en un estimado previo (y_{j-1}) y sólo un dato al instante j (x_j).

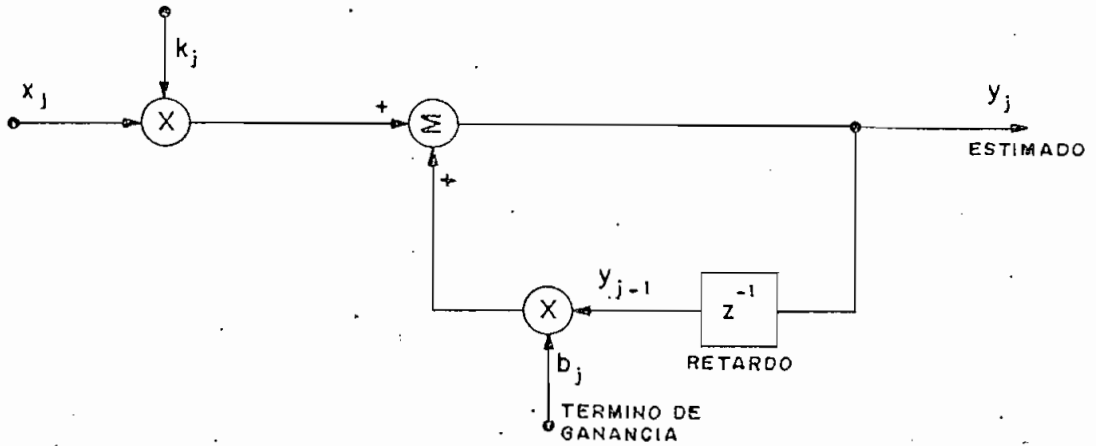


Fig. 3.2 Estimador recursivo de primer orden general.

Deseamos encontrar el estimado óptimo a partir de la ecuación (3.3), el "óptimo" se especifica como aquel estimado que minimiza el error cuadrático medio. Para conservar la nomenclatura del capítulo anterior designamos nuevamente al error cuadrático medio como Γ_j , esto es:

$$\Gamma_j = E[\epsilon_j^2] \quad (3.4)$$

donde $\epsilon_j = y_j - d_j$ es el error.

Consecuentemente

$$\begin{aligned} \Gamma_j &= E[(y_j - d_j)^2] \\ &= E[\{(b_j y_{j-1} + k_j x_j) - d_j\}^2] \end{aligned} \quad (3.5)$$

diferenciando Γ_j con respecto a b_j y k_j e igualando a cero tal resultado tendremos:

$$\frac{\delta \Gamma_j}{\delta b_j} = 2 E[(b_j y_{j-1} + k_j x_j - d_j) y_{j-1}] = 0 \quad (3.6)$$

$$\frac{\delta \Gamma_j}{\delta k_j} = 2 E[(b_j y_{j-1} + k_j x_j - d_j) x_j] = 0 \quad (3.7)$$

lo cual es equivalente a :

$$E[\epsilon_j y_{j-1}] = 0 \quad (3.8)$$

$$E[\epsilon_j x_j] = 0 \quad (3.9)$$

La ecuación (3.9) constituye el principio de ortogonalidad, el cual señala que el producto del error ϵ_j con cada muestra medida x_j es igual a cero en sentido promediado.

A partir de la ecuación (3.6) es posible obtener una relación entre b_j y k_j , esto es:

$$2 E [(b_j y_{j-1} + k_j x_j - d_j) y_{j-1}] = 0$$

$$E [(b_j y_{j-1}) y_{j-1}] = E [(d_j - k_j x_j) y_{j-1}] \quad (3.10)$$

si sumamos y restamos al lado izquierdo de la ecuación el término $[b_j d_{j-1}]$ se tiene:

$$E[[b_j(y_{j-1}-d_{j-1}) + b_j d_{j-1}]y_{j-1}] = E[(d_j - k_j x_j)y_{j-1}] \quad (3.11)$$

donde $y_{j-1}-d_{j-1} = \epsilon_{j-1}$; considerando la expresión para x_j dada por la ecuación (3.2) se llega a:

$$E[(b_j \epsilon_{j-1} + b_j d_{j-1}) y_{j-1}] = E[[d_j - k_j (c d_j + v_j)]y_{j-1}]$$

$$b_j E[\epsilon_{j-1} y_{j-1} + d_{j-1} y_{j-1}] = E[[d_j (1 - c k_j) - k_j v_j] y_{j-1}] \quad (3.12)$$

Tomando en consideración la expresión para y_{j-1} que se obtiene a partir de la ecuación (3.3), se tiene:

$$y_{j-1} = b_{j-1} y_{j-2} + k_{j-1} x_{j-1}$$

Analizando el término $E[\epsilon_{j-1} y_{j-1}]$ del lado izquierdo de la ecuación (3.12), y sustituyendo en él la expresión para y_{j-1} obtenida anteriormente queda:

$$\begin{aligned} E[\epsilon_{j-1} y_{j-1}] &= E[\epsilon_{j-1} b_{j-1} y_{j-2} + \epsilon_{j-1} k_{j-1} x_{j-1}] \\ &= b_{j-1} E[\epsilon_{j-1} y_{j-2}] + k_{j-1} E[\epsilon_{j-1} x_{j-1}] \end{aligned}$$

de acuerdo con las ecuaciones (3.8) y (3.9) se cumplen las siguientes relaciones: $E[\epsilon_{j-1} y_{j-2}] = 0$ y $E[\epsilon_{j-1} x_{j-1}] = 0$, de modo que el término $E[\epsilon_{j-1} y_{j-1}]$ en la ecuación (3.12) será nulo.

Por otro lado el término $E[v_j y_{j-1}]$ del lado derecho de la misma ecuación (3.12) es igualmente nulo a causa de que el estimado al instante $j-1$ es no correlacionado con el ruido de observación al instante j . Por consiguiente la ecuación (3.12) se reduce a:

$$b_j E[d_{j-1} y_{j-1}] = (1 - ck_j) E[d_j y_{j-1}] \quad (3.13)$$

de acuerdo con el modelo de generación de señal, ecuación (3.1), tendremos:

$$\begin{aligned} b_j E[d_{j-1} y_{j-1}] &= (1 - ck_j) E[(a d_{j-1} + g_{j-1}) y_{j-1}] \\ &= \{1 - ck_j\} \{E[ad_{j-1} y_{j-1}] + E[g_{j-1} y_{j-1}]\} \quad (3.14) \end{aligned}$$

donde $E[g_{j-1} y_{j-1}] = 0$; a fin de demostrar tal afirmación, se expresa la ecuación (3.3) del modelo de genera-

ción de señal sustituyendo en ella la expresión para x_j a partir de la ecuación (3.2), es decir:

$$y_j = b_j y_{j-1} + k_j (c d_j + v_j)$$

y sustituyendo d_j desde la ecuación (3.1).

$$\begin{aligned} y_j &= b_j y_{j-1} + c k_j (a d_{j-1} + g_{j-1}) + k_j v_j \\ &= b_j y_{j-1} + a c k_j d_{j-1} + c k_j g_{j-1} + k_j v_j \end{aligned}$$

de modo que para el instante de tiempo $j-1$ se tiene:

$$y_{j-1} = b_{j-1} y_{j-2} + a c k_{j-1} d_{j-2} + c k_{j-1} g_{j-2} + k_{j-1} v_{j-1} \quad (3.15)$$

al multiplicar ambos lados de la ecuación (3.15) por g_{j-1} y luego obtener el valor esperado de dicha expresión se tiene:

$$\begin{aligned} E[g_{j-1} y_{j-1}] &= b_{j-1} E[g_{j-1} y_{j-2}] + a c k_{j-1} E[g_{j-1} d_{j-2}] + \\ &\quad + c k_{j-1} E[g_{j-1} g_{j-2}] + k_{j-1} E[g_{j-1} v_{j-1}] \end{aligned}$$

donde los promedios de todos los productos del lado derecho de la ecuación (3.15) con g_{j-1} son cero debido a que todos los términos son no correlacionados con g_{j-1} ; en consecuencia:

$$E[g_{j-1} y_{j-1}] = 0 \quad (3.16)$$

al usar tal relación en la ecuación (3.14), ésta quedará de la forma:

$$b_j E[d_{j-1} y_{j-1}] = a(1 - c k_j) E[d_{j-1} y_{j-1}] \quad (3.17)$$

esto conduce a una relación final entre b_j y k_j :

$$b_j = a(1 - c k_j) \quad (3.18)$$

aplicando esta relación a la ecuación (3.3) se tiene:

$$y_j = a(1 - c k_j) y_{j-1} + k_j x_j$$

reagrupando términos se llega a la siguiente expresión para el estimado y_j :

$$y_j = a y_{j-1} + k_j [x_j - ac y_{j-1}] \quad (3.19)$$

La ecuación (3.19) constituye la definición del estimador recursivo de primer orden óptimo, también conocido como filtro escalar de Kalman. El primer término, ay_{j-1} , representa el mejor estimado de y_j , y es una predicción basada sobre observaciones pasadas. El segundo es un término de corrección dependiente de la diferencia entre la nueva muestra de datos y el estimado de dicha observación, $\hat{x}_j = acy_{j-1}$, y un factor de ganancia variable k_j , también conocida como la ganancia de Kalman. La forma de este filtro se ilustra en la figura 3.3.

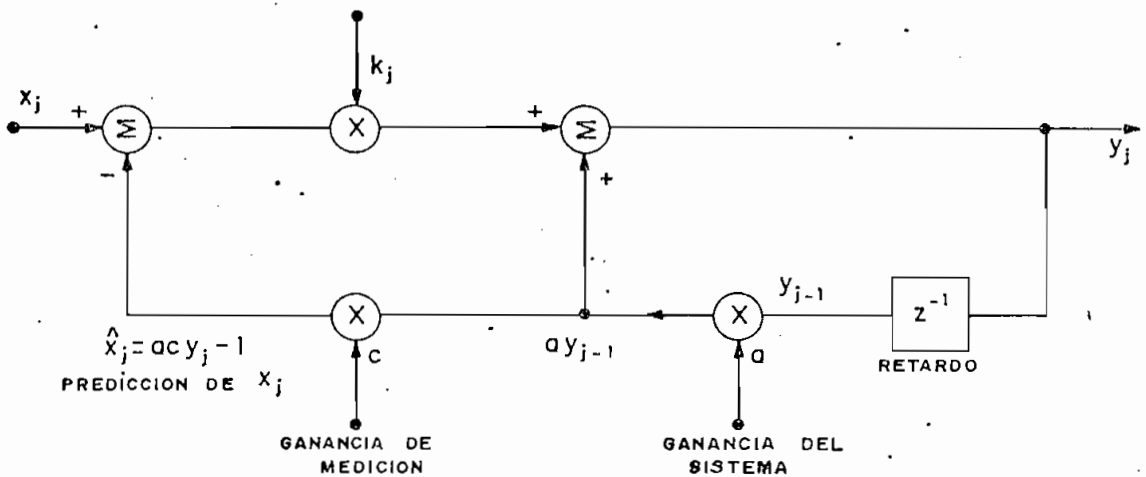


Fig. 3.3 Diagrama de bloques de un estimador reoursivo óptimo de primer orden o filtro escalar de Kalman.

3.1.1 a) Derivación de la Ganancia de Kalman

Luego de haber definido la forma del estimador recursivo óptimo de primer orden, resta obtener una expresión para la ganancia de Kalman. Para el efecto se procede de la siguiente forma:

$$\Gamma_j = E[\epsilon_j^2] = E[\epsilon_j(y_j - d_j)]$$

al sustituir y_j , ecuación (3.3), en ésta expresión se tiene:

$$\begin{aligned}\Gamma_j &= E[\epsilon_j[(b_j y_{j-1} + k_j x_j) - d_j]] && (3.20) \\ &= b_j E[\epsilon_j y_{j-1}] + k_j E[\epsilon_j x_j] - E[\epsilon_j d_j]\end{aligned}$$

donde los primeros términos del lado derecho de la ecuación son iguales a cero según (3.8) y (3.9). Consecuentemente se tendrá:

$$\Gamma_j = -E[\epsilon_j d_j] \quad (3.21)$$

Al usar la ecuación (3.2) para d_j y sustituir dicha expresión en el término de la derecha de la ecuación (3.21)

$$\begin{aligned}-E[\epsilon_j d_j] &= -\frac{1}{c} E[\epsilon_j(x_j - v_j)] \\ &= -\frac{1}{c} \{E[\epsilon_j x_j] - E[\epsilon_j v_j]\}\end{aligned}$$

de acuerdo a la ecuación (3.9), $E[\epsilon_j x_j] = 0$, de modo que:

$$-E[\epsilon_j d_j] = \frac{1}{c} E[\epsilon_j v_j] \quad (3.22)$$

sustituyendo esta expresión en la ecuación (3.21) queda:

$$\Gamma_j = \frac{1}{c} E[\epsilon_j v_j] \quad (3.23)$$

pero recuérdese que $\epsilon_j = y_j - d_j$, además si se usa la ecuación (3.3) para y_j se obtiene:

$$\begin{aligned} \Gamma_j &= \frac{1}{c} E[(b_j y_{j-1} + k_j x_j - d_j) v_j] \\ &= \frac{1}{c} \{b_j E[y_{j-1} v_j] + k_j E[x_j v_j] - E[d_j v_j]\} \end{aligned} \quad (3.24)$$

donde el término $E[y_{j-1} v_j] = 0$ por las razones expuestas anteriormente. Además $E[d_j v_j] = 0$, esto puede verificarse de la siguiente forma:

según la ecuación (3.1):

$$\begin{aligned} E[d_j v_j] &= E[(a d_{j-1} + g_{j-1}) v_j] \\ &= a E[d_{j-1} v_j] + E[g_{j-1} v_j] \end{aligned} \quad (3.25)$$

v_j es ruido blanco, y por lo tanto no correlacionado, de modo que cada uno de los términos del lado derecho de la ecuación (3.25) será nulo. En consecuencia: $E[d_j v_j] = 0$.

Por lo tanto la expresión (3.24) se reduce a:

$$\Gamma_j = \frac{k_j}{c} E[x_j v_j] \quad (3.26)$$

al reemplazar x_j , ecuación (3.2), en esta última expresión:

$$\begin{aligned}\Gamma_j &= \frac{k_j}{c} E[(cd_j + v_j) v_j] \\ &= \frac{k_j}{c} \{cE[d_j v_j] + E[v_j v_j]\}\end{aligned}$$

donde el primer término del lado derecho de la ecuación es cero, esto es $E[d_j v_j] = 0$, tal y como se determinó anteriormente, de modo que:

$$\Gamma_j = \frac{1}{c} k_j E[v_j^2]$$

pero $E[v_j^2] = \sigma_v^2$, y por lo tanto:

$$\Gamma_j = \frac{1}{c} k_j \sigma_v^2 \quad (3.27)$$

de modo que: $k_j = \frac{c \Gamma_j}{\sigma_v^2}$

Para resolver el problema completamente retornemos a la ecuación del error cuadrático medio:

$$\Gamma_j = E[(y_j - d_j)^2]$$

con y_j dado por la ecuación (3.19), de modo que:

$$\Gamma_j = E[(a y_{j-1} + k_j (x_j - a c y_{j-1})) - d_j]^2 \quad (3.28)$$

donde al sustituir x_j por la ecuación (3.2) y d_j por la ecuación (3.1) se tiene:

$$\begin{aligned}\Gamma_j &= E[ay_{j-1} + k_j(cd_j + v_j) - ack_jy_{j-1} - ad_{j-1} - g_{j-1}]^2 \\ &= E[ay_{j-1} + ck_j(ad_{j-1} + g_{j-1}) + k_jv_j - ack_jy_{j-1} - \\ &\quad - ad_{j-1} - g_{j-1}]^2\end{aligned}$$

agrupando términos:

$$\begin{aligned}\Gamma_j &= E[a(y_{j-1} - d_{j-1}) - ack_j(y_{j-1} - d_{j-1}) - \\ &\quad - (g_{j-1} - ck_jg_{j-1}) + k_jv_j]^2 \\ &= E[a\epsilon_{j-1} - ack_j\epsilon_{j-1} - (1 - ck_j)g_{j-1} + k_jv_j]^2 \\ &= E[a(1 - ck_j)\epsilon_{j-1} - (1 - ck_j)g_{j-1} + k_jv_j]^2 \quad (3.29)\end{aligned}$$

Al desarrollar la expresión (3.29), resulta que los valores esperados de los productos cruzados son cero, esto debido a que ϵ_{j-1} , g_{j-1} y v_j son independientes uno del otro, de manera que la ecuación (3.29) se reduce a:

$$\Gamma_j = a^2(1-ck_j)^2E[\epsilon_{j-1}]^2 + (1-ck_j)^2E[g_{j-1}]^2 + k_j^2E[v_j]^2 \quad (3.30)$$

pero $E[\epsilon_{j-1}]^2 = \Gamma_{j-1}$; $E[g_{j-1}]^2 = \sigma_g^2$; y, $E[v_j]^2 = \sigma_v^2$.
Por lo tanto:

$$\Gamma_j = a^2(1 - ck_j)^2 \Gamma_{j-1} + (1 - ck_j)^2 \sigma_g^2 + k_j^2 \sigma_v^2 \quad (3.31)$$

sustituyendo Γ_j dado en la ecuación (3.27) en esta última expresión se obtiene:

$$\frac{1}{c} k_j \sigma_v^2 = a^2(1 - ck_j)^2 \Gamma_{j-1} + (1 - ck_j)^2 \sigma_g^2 + k_j^2 \sigma_v^2$$

si de esta ecuación se despeja k_j , se obtiene la siguiente expresión para la ganancia de Kalman.

$$k_j = \frac{c[a^2 \Gamma_{j-1} + \sigma_g^2]}{\sigma_v^2 + c^2 a^2 \Gamma_{j-1} + c^2 \sigma_g^2} \quad (3.32)$$

Debe entonces determinarse primero k_j partiendo del conocimiento de Γ_{j-1} ; y, luego calcular Γ_j a partir de:

$$\Gamma_j = \frac{1}{c} \sigma_v^2 k_j \quad (3.33)$$

Las ecuaciones (3.19), (3.32) y (3.33) definen el estimador óptimo recursivo de primer orden.

3.1.1 b) Estimador recursivo óptimo de orden n

Normalmente en situaciones prácticas el modelo de señal autoregresivo de primer orden es insuficiente para caracterizar adecuadamente un proceso físico; y, es más probable que se requiera de un proceso AR de orden n.

Las ecuaciones para el estimador recursivo óptimo de primer orden se pueden modificar a fin de considerar filtros de mayor orden, para el efecto deberán sustituirse escalares por vectores de orden n.

Sea pues un proceso autoregresivo de segundo orden definido por:

$$d_j = a d_{j-1} + b d_{j-2} + g_{j-1} \quad (3.34)$$

se definen dos variables de estado $d1_j$ y $d2_j$ tal que:

$$\begin{aligned} d1_j &= d_j \\ d2_j &= d_{j-1} \end{aligned}$$

por lo tanto es posible escribir la ecuación (3.34) como un par de ecuaciones de estado, donde:

$$\begin{aligned}d1_j &= a d1_{j-1} + b d2_{j-1} + g_{j-1} \\d2_j &= d1_{j-1}\end{aligned}\tag{3.35}$$

al expresar las ecuaciones (3.35) en forma matricial se tiene:

$$\begin{bmatrix} d1_j \\ d2_j \end{bmatrix} = \begin{bmatrix} a & b \\ 1 & 0 \end{bmatrix} \begin{bmatrix} d1_{j-1} \\ d2_{j-1} \end{bmatrix} + \begin{bmatrix} g_{j-1} \\ 0 \end{bmatrix}$$

o lo que es lo mismo:

$$D_j = A D_{j-1} + G_{j-1}\tag{3.36}$$

Las ecuaciones están ahora en formato de vector, pero tienen la misma forma que para el caso escalar; en un desarrollo similar es posible llegar a las siguientes expresiones para el caso del estimador óptimo recursivo de orden n.

$$Y_j = A Y_{j-1} + K_j [X_j - C A Y_{j-1}]\tag{3.37}$$

$$K_j = \Gamma_{1j} C^T [C \Gamma_{1j} C^T + Z_j]^{-1}\tag{3.38}$$

$$\Gamma_{1j} = A \Gamma_{j-1} A^T + Q_{j-1}\tag{3.39}$$

$$\Gamma_j = \Gamma_{1j} - K_j C \Gamma_{1j}\tag{3.40}$$

donde la varianza del ruido de observación σ_v^2 (escalar) y la varianza del ruido del sistema σ_g^2 (escalar), se han reemplazado por las matrices Z_j y Q_j respectivamente.

$$Z_j = E[V_j V_j^T]$$

$$Q_j = E[G_j G_j^T]$$

Similarmente la ganancia del filtro k_j , se ha sustituido por una matriz K_j de orden $n \times n$.

La actualización de los coeficientes de ponderación del filtro adaptivo en el algoritmo RLS se realiza en base a un proceso de estimación tipo Kalman como se verá mas adelante.

3.1.2 DERIVACION DEL ALGORITMO ADAPTIVO RLS

3.1.2.1 ALGORITMO ADAPTIVO RLS PREVENTANEADO

El filtro adaptivo RLS es un filtro digital con una estructura recursiva, constituida por una línea de retardos, cuyos coeficientes de ponderación son ajustables, y cuya respuesta impulso se controla por el algoritmo adaptivo RLS, tal estructura se muestra en la figura 3.4.

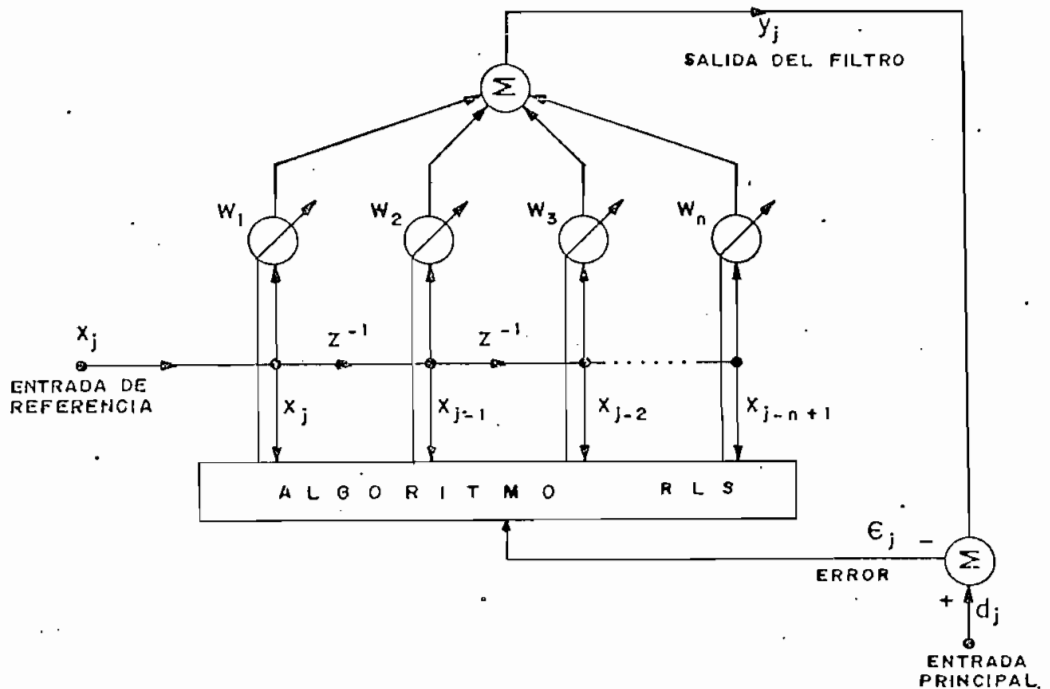


Fig. 3.4 Diagrama de un filtro adaptivo RLS.

Tal filtro en esencia es similar al filtro adaptivo LMS, pero difiere como se verá mas adelante en la ley adaptiva usada para actualizar los coeficientes del filtro.

Sea entonces d_j la entrada principal al sistema de cancelación de ruido en el instante de tiempo discreto j ; X_j la entrada al filtro FIR de orden n , y_j la salida del filtro y, sea ϵ_j la salida del sistema de cancelación de ruido; es evidente desde la figura 3.4 que se cumplirá:

$$\epsilon_j = d_j - w_1 x_j - w_2 x_{j-1} - \dots - w_n x_{j-n+1} \quad (3.41)$$

donde el vector de coeficientes del filtro al igual que en capítulo II será denotado por:

$$W^T = [w_1 \ w_2 \ \dots \ w_n] \quad (3.42)$$

A continuación se considerará un algoritmo diseñado para minimizar la suma de los cuadrados de la salida del cancelador, la cual viene dada por la siguiente expresión:

$$V_j = \sum_{s=1}^j \epsilon_s^2 \quad (3.43)$$

El vector de coeficientes del filtro óptimo para el que esta suma se minimiza es W^* , el cual se asume es constante o está variando muy lentamente con respecto a la velocidad de muestreo del filtro adaptivo, tal como se vió en el algoritmo LMS.

Trataremos primeramente con un problema de minimización puramente determinístico, y luego se extenderá su interpretación al caso estocástico. Para ver el problema más claramente se reescribirá (3.41) en forma matricial, tal que:

$$\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_j \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_j \end{bmatrix} - \begin{bmatrix} x_1 & 0 & \dots & 0 \\ x_2 & x_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & 0 & \dots \\ \dots & \dots & \dots & x_1 & \dots \\ x_j & x_{j-1} & \dots & x_{j-n+1} & \dots \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad (3.44)$$

donde n es el orden del filtro.

Es decir se ha asumido que $x_j = 0$ para $j < 1$; lo cual significa que los datos han sido preventaneados, esto es dichos datos se han multiplicado por una función ventana r_j , donde $r_j = 0$ para $j < 0$ y $r_j = 1$ para $j \geq 1$. Es posible escoger diversas funciones ventana, conduciendo cada una de ellas a algoritmos adaptivos diferentes; así por ejemplo, cuando los datos no son ventaneados, se obtiene la denominada forma no ventaneada o covarianza de estas ecuaciones, esto es:

$$\begin{bmatrix} \epsilon_n \\ \vdots \\ \epsilon_j \end{bmatrix} = \begin{bmatrix} d_n \\ \vdots \\ d_j \end{bmatrix} - \begin{bmatrix} x_n & \dots & \dots & x_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_j & \dots & \dots & x_{j-n+1} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \quad (3.45)$$

El presente trabajo de tesis estudiará el algoritmo RLS en la forma preventaneada; y, se introducirá posteriormente una innovación que dará lugar al denominado algoritmo preventaneado exponencialmente.

Escribiendo la ecuación (3.44) en forma matricial se tendrá¹:

$$\underline{\epsilon}_j = \underline{d}_j - [X_j] W \quad (3.46)$$

Minimizar la función de costo, V_j , dada en la ecuación (3.43) es equivalente a minimizar la norma al-cuadrado de $\underline{\epsilon}_j$ en la ecuación (3.46), puesto que:

$$V_j = ||\underline{\epsilon}_j||^2 = \underline{\epsilon}_j^T \underline{\epsilon}_j \quad (3.47)$$

El vector que minimiza ésta norma vendrá dado por:

$$W_j = [[X_j]^T [X_j]]^{-1} [X_j]^T \underline{d}_j \quad (3.48)$$

A fin de comprobarlo se procederá de la siguiente forma: de acuerdo a la ecuación (3.46) se verifica que:

$$\underline{\epsilon}_j = \underline{d}_j - [X_j] W$$

$$\underline{\epsilon}_j^T = [\underline{d}_j - [X_j] W]^T = \underline{d}_j^T - [X_j W]^T = \underline{d}_j^T - W^T [X_j]^T$$

Sustituyendo estas expresiones para $\underline{\epsilon}_j$ y $\underline{\epsilon}_j^T$ respectivamente, en la ecuación (3.47), la función de costo quedará de la forma:

$$V_j = [\underline{d}_j^T - W^T [X_j]^T] [\underline{d}_j - [X_j] W] \quad (3.49)$$

$$= \underline{d}_j^T \underline{d}_j - \underline{d}_j^T [X_j] W - W^T [X_j]^T \underline{d}_j + W^T [X_j]^T [X_j] W \quad (3.50)$$

A fin de obtener el vector de coeficientes del filtro W que minimiza la función de costo V_j , se derivará la ecua-

¹ donde $[X_j]$ señalará que se trata de una matriz a fin de diferenciarla de X_j vector; y, la notación en letras minúsculas subrayadas identificará a las variables como vectores.

ción (3.50) con respecto a W y luego se igualará a cero dicha derivada, esto es:

$$\begin{aligned} \frac{\delta V_j}{\delta W} &= \frac{\delta}{\delta W} [d_j^T d_j - d_j^T [X_j] W - W^T [X_j]^T d_j + W^T [X_j]^T [X_j] W] \\ &= - \frac{\delta}{\delta W} [W^T [X_j]^T d_j]^T - \frac{\delta}{\delta W} [W^T [X_j]^T d_j] + \frac{\delta}{\delta W} [W^T [X_j]^T [X_j] W] \end{aligned} \quad (3.51)$$

$W^T [X_j]^T d_j$ es un escalar y por lo tanto se verificará que: $[W^T [X_j]^T d_j]^T = W^T [X_j]^T d_j$, de esta manera la ecuación (3.51) quedará como sigue:

$$\frac{\delta V_j}{\delta W} = -2 \frac{\delta}{\delta W} [W^T [X_j]^T d_j] + \frac{\delta}{\delta W} [W^T [X_j]^T [X_j] W] \quad (3.52)$$

pero

$$\frac{\delta}{\delta W} [W^T [X_j]^T d_j] = [X_j]^T d_j \quad (3.53)$$

$$\frac{\delta}{\delta W} [W^T [X_j]^T [X_j] W] = 2 [X_j]^T [X_j] W \quad (3.54)$$

de modo que la ecuación (3.52), al igualarla a cero y tomando en consideración las expresiones anteriores (3.53 y 3.54), se convierte en :

$$-2 [X_j]^T d_j + 2 [X_j]^T [X_j] W = 0 \quad (3.55)$$

y por consiguiente se verificará:

$$[X_j]^T [X_j] W = [X_j]^T d_j \quad (3.56)$$

al premultiplicar ambos lados de la ecuación (3.56) por el término $[[X_j]^T [X_j]]^{-1}$ se tendrá:

$$W = [[X_j]^T [X_j]]^{-1} [X_j]^T d_j$$

lo cual confirma que efectivamente el vector de coeficientes del filtro que minimiza la función de costo es aquel dado por la ecuación (3.48).

A cada paso del tiempo se podría evaluar (3.48) a fin de obtener los coeficientes del filtro; esto sin embargo involucra una gran cantidad de cálculos, puesto que los coeficientes se determinan tomando en consideración todos los datos observados desde el inicio. Una forma más útil de este algoritmo se obtiene desarrollando un método recursivo para el cálculo de W.

El algoritmo de los mínimos cuadrados recursivos posibilita actualizar los coeficientes del filtro tomando en consideración toda la información contenida en el conjunto actual de coeficientes del filtro y, solamente es necesario una cantidad de cálculos incremental a cada paso de tiempo; esto es posible gracias a la aplicación de un proceso de estimación tipo Kalman como el descrito en la sección anterior.

A fin de derivar el algoritmo de mínimos cuadrados recursivos, considérese que se realiza una observación más al instante $j+1$, de modo que el vector de datos de entrada al filtro para este instante será:

$$X_{j+1} = [x_{j+1} \quad x_j \quad \dots \quad x_{j-n+2}] \quad (3.57)$$

Si en la ecuación (3.48) definimos como:

$$P_j = [[X_j]^T [X_j]]^{-1} \quad (3.58)$$

al considerar esta nueva observación la ecuación (3.58) se transformará en¹:

$$P_{j+1} = \left[\begin{array}{c|c} [[X_j]^T & X_{j+1} \end{array} \right] \left[\begin{array}{c} [X_j] \\ \hline X_{j+1}^T \end{array} \right]^{-1} \quad (3.59)$$

la cual es equivalente a la expresión:

$$P_{j+1} = [[X_j]^T[X_j] + X_{j+1} X_{j+1}^T]^{-1} \quad (3.60)$$

la matriz inversa en (3.60) puede determinarse mediante el lema de inversión de matrices, que señala lo siguiente:

lema de inversión de matrices

Sean ^DA, B y C matrices de dimensiones compatibles; y, A una matriz no singular, se cumplirá que:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1} DA^{-1} \quad (3.61)$$

prueba

A fin de demostrar que se cumple la ecuación (3.61) probaremos que:

$$[A + BCD] [A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1} DA^{-1}] = I$$

donde I es la matriz identidad. Desarrollando el producto término a término queda:

$$AA^{-1} + BCDA^{-1} - AA^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1} - BCDA^{-1}B[C^{-1} +$$

$$+DA^{-1}B]^{-1}DA^{-1} = I$$

¹ Las líneas segmentadas señalan que se ha añadido una columna o una fila a la matriz original, en la ecuación (3.59).

$$I + BCDA^{-1} - B[C^{-1}+DA^{-1}B]^{-1}DA^{-1} - BCDA^{-1}B[C^{-1}+DA^{-1}B]^{-1}DA^{-1} = I$$

$$I+BCDA^{-1}-BCC^{-1}[C^{-1}+DA^{-1}B]^{-1}DA^{-1}-BCDA^{-1}B[C^{-1}+DA^{-1}B]^{-1}DA^{-1} = I$$

agrupando términos

$$I+BCDA^{-1}-BC\{C^{-1}[C^{-1}+DA^{-1}B]^{-1} + DA^{-1}B[C^{-1}+DA^{-1}B]^{-1}\}DA^{-1} = I$$

$$I + BCDA^{-1} - BC\{[C^{-1} + DA^{-1}B][C^{-1} + DA^{-1}B]^{-1}\}DA^{-1} = I$$

en consecuencia

$$I + BCDA^{-1} - BCDA^{-1} = I$$

$$I = I$$

con lo cual quedaría demostrado que se cumple la igualdad (3.61).

Si aplicamos el lema de inversión de matrices a la ecuación (3.60); considerando como A,B,C, y D a los términos :

A =	$[X_{j+1}]^T[X_j]$	matriz de orden $n \times n$ y además no singular.
B =	X_{j+1}	vector columna de orden n
C =	1	matriz identidad de orden 1
D =	X_{j+1}^T	vector fila de orden n

como se observa las matrices son de dimensiones totalmente compatibles, de modo que se tiene:

$$P_{j+1} = P_j - P_j X_{j+1}[1 + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T P_j \quad (3.62)$$

Por otro lado tomando en consideración la ecuación (3.48) e incluyendo una nueva observación, el vector de coefi-

cientes de ponderación del filtro para el instante $j+1$ vendrá dado por:

$$W_{j+1} = P_{j+1} [X_{j+1}]^T d_{j+1} \quad (3.63)$$

la ecuación (3.63) puede expresarse en la forma:

$$\begin{aligned} W_{j+1} &= P_{j+1} [[X_j]^T d_j + X_{j+1} d_{j+1}] \\ &= P_{j+1} [[X_j]^T d_j] + P_{j+1} X_{j+1} d_{j+1} \end{aligned}$$

al sustituir en esta ecuación la expresión para P_{j+1} , ecuación (3.62), se tendrá:

$$\begin{aligned} W_{j+1} &= \{P_j - P_j X_{j+1} [1 + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T P_j\} \{ [X_j]^T d_j \} + \\ &\quad + P_{j+1} X_{j+1} d_{j+1} \\ &= P_j [X_j]^T d_j - \{P_j X_{j+1} [1 + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T\} \\ &\quad \{P_j [X_j]^T d_j\} + P_{j+1} X_{j+1} d_{j+1} \\ &= W_j - P_j X_{j+1} [1 + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T W_j + \\ &\quad + P_{j+1} X_{j+1} d_{j+1} \end{aligned} \quad (3.64)$$

Definiendo:

$$K_{j+1} = P_{j+1} X_{j+1} \quad (3.65)$$

de modo que al sustituir nuevamente P_{j+1} dado en la ecuación (3.62) en ésta última expresión se tendrá:

$$\begin{aligned} K_{j+1} &= \{P_j - P_j X_{j+1} [1 + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T P_j\} X_{j+1} \\ &= P_j X_{j+1} - P_j X_{j+1} [1 + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T P_j X_{j+1} \\ &= P_j X_{j+1} \{1 - [1 + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T P_j X_{j+1}\} \end{aligned}$$

$$\begin{aligned}
 &= P_J X_{J+1} \left[1 - \frac{X_{J+1}^T P_J X_{J+1}}{1 + X_{J+1}^T P_J X_{J+1}} \right] \\
 &= P_J X_{J+1} \left[\frac{1 + X_{J+1}^T P_J X_{J+1} - X_{J+1}^T P_J X_{J+1}}{1 + X_{J+1}^T P_J X_{J+1}} \right]
 \end{aligned}$$

$$K_{J+1} = P_J X_{J+1} [1 + X_{J+1}^T P_J X_{J+1}]^{-1} \quad (3.66)$$

usando entonces la expresión dada en (3.66) para K_{J+1} , y sustituyéndola en la ecuación (3.64) se obtiene:

$$W_{J+1} = W_J - K_{J+1} X_{J+1}^T W_J + P_{J+1} X_{J+1} d_{J+1}$$

pero $P_{J+1} = \frac{K_{J+1}}{X_{J+1}}$, en consecuencia:

$$W_{J+1} = W_J - K_{J+1} X_{J+1}^T W_J + \frac{K_{J+1}}{X_{J+1}} X_{J+1} d_{J+1}$$

lo que conduce finalmente a la siguiente expresión para W_{J+1} :

$$W_{J+1} = W_J + K_{J+1} [d_{J+1} - X_{J+1}^T W_J] \quad (3.67)$$

Nótese la similitud de la expresión (3.67) con que se obtuvo para el estimador recursivo de primer orden, ecuación (3.19), y así mismo con la expresión para el estimador óptimo recursivo de orden n , ecuación (3.37).

Las ecuaciones (3.62), (3.66) y (3.67) constituyen al algoritmo RLS.

3.1.2.2 ALGORITMO RLS PONDERADO EXPONENCIALMENTE

El algoritmo RLS tal y como se ha enunciado tiene una memoria infinita, pues los valores de los coeficientes del filtro son funciones de todas las entradas pasadas; es sin embargo a menudo muy útil introducir un "factor de olvido" dentro del algoritmo, de esta forma se da mayor peso o importancia a los datos más recientes y menor peso o importancia a los datos anteriores. Una alternativa para el factor de olvido es sustituir la función de costo, V_j , por una suma de los cuadrados de los errores ponderada exponencialmente, esto es:

$$\hat{v}_j = \sum_{s=1}^j \rho^{j-s} \epsilon_s^2 \quad (3.68)$$

donde ρ es una constante que determina la memoria efectiva del algoritmo; tal que $0 < \rho \leq 1$. Cuando $\rho = 1$ el algoritmo tendrá una memoria infinita y estaremos en el caso anteriormente visto del algoritmo preventaneado donde:

$$\hat{v}_j = \sum_{s=1}^j 1^{j-s} \epsilon_s^2 = \sum_{s=1}^j \epsilon_s^2$$

Cuando $\rho < 1$ el algoritmo tendrá una memoria efectiva de Ω puntos de datos, donde Ω estará dado por:

$$\Omega = \frac{-1}{\log \rho} \approx \frac{1}{1 - \rho} \quad (3.69)$$

a fin de comprender mejor lo expuesto anteriormente res-

pecto a la memoria efectiva del algoritmo, consideremos el caso donde $\epsilon_s^2=1$, en consecuencia se tendrá:

$$\hat{v}_j = \sum_{s=1}^j \phi^{j-s} = \phi^{j-1} + \phi^{j-2} + \dots + \phi^1 + \phi^0 \quad (3.70)$$

la ecuación (3.70) constituye una serie geométrica cuya suma será:

$$\hat{v}_j = \frac{1 - \phi^j}{1 - \phi} \quad (3.71)$$

se define la longitud de memoria efectiva del algoritmo como aquel valor de $n=j+1$ para el cual se cumple:

$$\frac{\hat{v}_n}{\hat{v}_\infty} \approx 0.9 \quad (3.72)$$

pero de la ecuación (3.71) se desprende que el valor de \hat{v}_j para $j=\infty$ será:

$$\hat{v}_\infty = \lim_{j \rightarrow \infty} \frac{1 - \phi^j}{1 - \phi} = \frac{1}{1 - \phi}, \text{ pues recuérdese que } \phi < 1$$

además de acuerdo a (3.71) también se cumplirá que:

$$\hat{v}_n = \frac{1 - \phi^n}{1 - \phi}$$

de modo que

$$\frac{\hat{v}_n}{\hat{v}_\infty} = 1 - \phi^n \quad (3.73)$$

por lo tanto, de acuerdo con (3.72) se tendrá:

$$1 - \phi^n = 0.9$$

$$\phi^n = 0.1$$

$$n \log \phi = \log(0.1) = -1$$

por lo tanto

$$n = \frac{-1}{\log \phi} \quad (3.74)$$

cuando ϕ es muy cercano a la unidad, $\log \phi \approx \phi - 1$; de modo que la memoria efectiva del algoritmo vendrá dada por:

$$n = \frac{-1}{\phi - 1} = \frac{1}{1 - \phi} \quad (3.75)$$

En consecuencia para valores de $\phi=1$ el algoritmo tendrá una memoria infinita tal como lo habíamos dicho ($n \rightarrow \infty$). Mientras ϕ se aproxime más a la unidad la memoria efectiva del algoritmo se irá incrementando, en tanto que valores pequeños de ϕ la memoria efectiva del algoritmo disminuirá.

Los típicos valores de ϕ están en el rango de 0.95 (para variaciones rápidas, es decir damos mayor peso a los datos más recientes y prácticamente olvidamos los datos anteriores) a 0.999 (para variaciones lentas, pues con tal valor de ϕ damos importancia a casi todos los datos pasados).

Si se toma el factor de olvido ϕ^{j-s} , de la ecuación (3.-68), éste constituye un término de la forma exponencial que da diferente peso a los datos. Consideremos que $j=100$ y analicemos éste factor ϕ^{j-s} para el 20-avo dato dependiendo del valor que adopte ϕ . Así:

- para $\rho=0.95$ tendremos:

$$\rho^{j-20} = (0.95)^{100-20} = 0.02$$

$$\text{la memoria efectiva ser\'a } \Omega = \frac{1}{1 - 0.95} = 20$$

- para $\rho=0.999$ tendremos:

$$\rho^{j-20} = (0.999)^{100-20} = 0.92$$

$$\text{la memoria efectiva ser\'a } \Omega = \frac{1}{1 - 0.999} = 1000$$

de donde se puede ver que para $\rho=0.95$ el peso que se da a los valores anteriores al dato 20 y al mismo dato 20 es pequeño, esto lo confirma la memoria efectiva del algoritmo, $\Omega=20$, que señala que únicamente son considerados los 20 últimos datos en el algoritmo despreciándose todos aquellos datos anteriores. En cambio para $\rho=0.999$ prácticamente se está dando importancia a casi todos los valores anteriores al dato 20, efectivamente la memoria del algoritmo es $\Omega=1000$, de manera que para el caso del ejemplo se tomarán en consideración todos los datos.

Para ver el efecto del factor de olvido sobre el algoritmo recursivo, analizemos el nuevo criterio del error, el mismo que puede expresarse como¹:

$$\hat{v}_j = \epsilon_j^T \Phi_j \epsilon_j \quad (3.76)$$

$$= \epsilon_j^T \Phi_j^{1/2} \Phi_j^{1/2} \epsilon_j \quad (3.77)$$

donde

¹ Puesto que Φ_j es una matriz diagonal, $\Phi_j^{1/2}$ seguirá siendo una matriz diagonal.

$$\Phi_J = \begin{bmatrix} \phi^{J-1} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \phi^{J-2} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \phi & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & 1 \end{bmatrix} \quad (3.78)$$

expandiendo (3.76) de acuerdo a (3.78) tendremos:

$$\hat{v}_J = [\epsilon_1 \epsilon_2 \dots \epsilon_J] \begin{bmatrix} \phi^{J-1} & & & & & \\ & \phi^{J-2} & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \phi \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \cdot \\ \cdot \\ \epsilon_{J-1} \\ \epsilon_J \end{bmatrix} \quad (3.79)$$

$$\hat{v}_J = [\epsilon_1 \phi^{J-1} \epsilon_2 \phi^{J-2} \dots \epsilon_{J-1} \phi \epsilon_J] \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \cdot \\ \cdot \\ \epsilon_J \end{bmatrix}$$

$$\hat{v}_J = \phi^{J-1} \epsilon_1^2 + \phi^{J-2} \epsilon_2^2 + \dots + \phi \epsilon_{J-1}^2 + \epsilon_J^2 \quad (3.80)$$

de modo que \hat{v}_J expresado como (3.76) es equivalente a (3.68).

Si ahora premultiplicamos la ecuación (3.46) por $\Phi_J^{1/2}$ tendremos:

$$\Phi_J^{1/2} \underline{\epsilon}_J = \Phi_J^{1/2} d_J - \Phi_J^{1/2} [X_J] W \quad (3.81)$$

la transpuesta de (3.81) será¹:

¹ Recuérdese que la transpuesta de una matriz diagonal es la misma matriz, por lo tanto $[\Phi_J^{1/2}]^T = \Phi_J^{1/2}$.

$$[\Phi_J^{1/2} \underline{e}_J]^T = \underline{d}_J^T \Phi_J^{1/2} - W^T [X_J]^T \Phi_J^{1/2} = \underline{e}_J^T \Phi_J^{1/2} \quad (3.82)$$

de modo que \hat{v}_J quedará de la forma:

$$\begin{aligned} \hat{v}_J &= \underline{e}_J^T \Phi_J^{1/2} \Phi_J^{1/2} \underline{e}_J \\ &= [\underline{d}_J^T \Phi_J^{1/2} - W^T [X_J]^T \Phi_J^{1/2}] [\Phi_J^{1/2} \underline{d}_J - \Phi_J^{1/2} X_J W] \end{aligned} \quad (3.83)$$

$$\begin{aligned} &= \underline{d}_J^T \Phi_J \underline{d}_J - \underline{d}_J^T \Phi_J X_J W - W^T [X_J]^T \Phi_J \underline{d}_J + \\ &\quad + W^T [X_J]^T \Phi_J X_J W \\ &= \underline{d}_J^T \Phi_J \underline{d}_J - [W^T [X_J]^T \Phi_J \underline{d}_J]^T - W^T [X_J]^T \Phi_J \underline{d}_J + \\ &\quad + W^T [X_J]^T \Phi_J X_J W \end{aligned}$$

pero $W^T [X_J]^T \Phi_J \underline{d}_J$ es escalar de modo que:

$$[W^T [X_J]^T \Phi_J \underline{d}_J]^T = W^T [X_J]^T \Phi_J \underline{d}_J$$

y por lo tanto

$$\hat{v}_J = \underline{d}_J^T \Phi_J \underline{d}_J - 2W^T [X_J]^T \Phi_J \underline{d}_J + W^T [X_J]^T \Phi_J X_J W \quad (3.84)$$

al derivar \hat{v}_J con respecto a W e igualar a cero dicha derivada, se obtiene aquel vector de coeficientes óptimo que minimiza la función de costo, es decir:

$$\frac{\delta \hat{v}_J}{\delta W} = -2 \frac{\delta}{\delta W} [W^T [X_J]^T \Phi_J \underline{d}_J] + \frac{\delta}{\delta W} [W^T [X_J]^T \Phi_J X_J W] = 0 \quad (3.85)$$

pero

$$\frac{\delta}{\delta W} [W^T [X_J]^T \Phi_J \underline{d}_J] = [X_J]^T \Phi_J \underline{d}_J \quad (3.86)$$

y

$$\frac{\delta}{\delta W} [W^T [X_j]^T \Phi_j X_j W] = 2 [X_j]^T \Phi_j X_j W \quad (3.87)$$

en consecuencia

$$-2 [[X_j]^T \Phi_j d_j] + 2 [[X_j]^T \Phi_j X_j W] = 0$$

$$[X_j]^T \Phi_j X_j W = [X_j]^T \Phi_j d_j$$

y por lo tanto, el vector de coeficientes óptimo vendrá dado por:

$$W = [[X_j]^T \Phi_j X_j]^{-1} [X_j]^T \Phi_j d_j \quad (3.88)$$

Para esta versión del algoritmo RLS ponderado exponencialmente se define P_j como:

$$P_j = [[X_j]^T \Phi_j X_j]^{-1} \quad (3.89)$$

Siguiendo un procedimiento similar al visto anteriormente, es posible llegar a la siguiente expresión para P_{j+1}

$$P_{j+1} = [\phi [[X_j]^T \Phi_j X_j] + X_{j+1} X_{j+1}^T]^{-1} \quad (3.90)$$

$$= [\phi P_j^{-1} + X_{j+1} X_{j+1}^T]^{-1} \quad (3.91)$$

aplicando nuevamente el lema de inversión de matrices a la ecuación (3.91), considerando a:

$$A = \phi P_j^{-1}$$

$$B = X_{j+1}$$

$$C = 1$$

$$D = X_{j+1}^T$$

se llega a :

$$P_{j+1} = \frac{P_j}{\phi} - \frac{P_j}{\phi} X_{j+1} [1 + X_{j+1}^T \frac{P_j}{\phi} X_{j+1}]^{-1} X_{j+1}^T \frac{P_j}{\phi}$$

la misma que conduce a la siguiente expresión para P_{j+1} :

$$P_{j+1} = \frac{P_j - P_j X_{j+1} [\phi + X_{j+1}^T P_j X_{j+1}]^{-1} X_{j+1}^T P_j}{\phi} \quad (3.92)$$

En forma similar al algoritmo RLS sin factor de olvido ($\phi=1$), se tiene para este caso que:

$$W_{j+1} = P_{j+1} [X_{j+1}]^T \Phi_{j+1} d_{j+1}$$

esta expresión se puede escribir en la forma:

$$W_{j+1} = P_{j+1} [\phi [X_j]^T \Phi_j d_j + X_{j+1} d_{j+1}] \quad (3.93)$$

siguiendo los mismos pasos que para el algoritmo RLS sin factor de olvido, es posible comprobar que la ecuación (3.67) se mantiene, esto es la actualización de los coeficientes de ponderación del filtro vendrá dada por una ecuación de la forma:

$$W_{j+1} = W_j + K_{j+1} [d_{j+1} - X_{j+1}^T W_j] \quad (3.94)$$

sin embargo ahora se usa la nueva definición de P_{j+1} para el cálculo de K_{j+1} , esto es:

$$K_{j+1} = P_{j+1} X_{j+1}$$

$$K_{j+1} = P_j X_{j+1} [\phi + X_{j+1}^T P_j X_{j+1}]^{-1} \quad (3.95)$$

3.1.2.3 INTERPRETACION ESTOCASTICA

El algoritmo RLS se derivó sin hacer ninguna consideración acerca de los datos de entrada, se estableció únicamente que minimizaría la suma de los errores al cuadrado a cada paso del tiempo, este hecho es por sí solo muy útil; sin embargo a menudo es deseable conocer más acerca del comportamiento del filtro. Para analizar más detalladamente las propiedades del filtro adaptivo RLS es necesario establecer la naturaleza de los estadísticos de los datos, es importante recordar, que el algoritmo realizará su función establecida, esto es la minimización de V_j ó \hat{v}_j para secuencias de entrada arbitrarias.

Consideraremos el caso para el cual los datos de entrada x_j corresponden a un proceso estocástico estacionario y ergódico de media cero y correlación $r(\tau)$, donde $r(\tau)$ está dado por:

$$r(\tau) = E[x_j x_{j-\tau}] \quad (3.96)$$

Si el proceso es ergódico los promedios en el tiempo pueden reemplazarse en el límite por los valores esperados, es decir:

$$r(\tau) = E[x_j x_{j-\tau}] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^{N-\tau} x_j x_{j+\tau} \quad (3.97)$$

donde N representa el número de datos muestreados.

De manera que si se toma en consideración que los datos del proceso satisfacen la ecuación (3.97) es posible demostrar que se cumple:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \{[X_j]^T [X_j]\} = R_{xx} = \begin{bmatrix} r_{xx}(0) & \dots & r_{xx}(n-1) \\ \vdots & \ddots & \vdots \\ r_{xx}(n-1) & \dots & r_{xx}(0) \end{bmatrix} \quad (3.98)$$

Como es evidente, R_{xx} , es la matriz de autocorrelación de los datos de entrada.

En forma análoga se tiene que:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \{[X_j]^T d_j\} = R_{xd} = \begin{bmatrix} r_{xd}(1) & \dots & r_{xd}(n) \\ \vdots & \ddots & \vdots \\ r_{xd}(n) & \dots & r_{xd}(1) \end{bmatrix} \quad (3.99)$$

Por consiguiente se cumplirá:

$$\lim_{N \rightarrow \infty} NP_j = \lim_{N \rightarrow \infty} \left[\frac{1}{N} [X_j]^T [X_j] \right]^{-1} = R_{xx}^{-1} \quad (3.100)$$

De manera que es posible determinar, en el límite, a que tiende el vector de coeficientes de ponderación del filtro, así pues¹:

$$\lim_{N \rightarrow \infty} W_j = \lim_{N \rightarrow \infty} \{[X_j]^T [X_j]\}^{-1} [X_j]^T d_j$$

¹ donde W_j corresponde a la expresión dada en la ecuación (3.48)

$$\begin{aligned}
 &= \lim_{N \rightarrow \infty} \left[\frac{1}{N} [X_j]^T [X_j] \right]^{-1} \left[\frac{1}{N} [X_j]^T d_j \right] \\
 &= R_{xx}^{-1} R_{xd} \quad (3.101)
 \end{aligned}$$

$$\lim_{N \rightarrow \infty} W_j = W^* = R_{xx}^{-1} R_{xd}$$

$$W^* = R_{xx}^{-1} R_{xd} \quad (3.102)$$

En otras palabras, en el límite, el vector de coeficientes de ponderación del filtro converge a la ecuación de Wiener-Hopf descrita en el capítulo II (ecuación 2.13). Por otro lado, de acuerdo a la ecuación (3.100) es evidente que la matriz P_j convenientemente normalizada converge al inverso de la matriz de autocorrelación.

3.1.3 EL FILTRO ADAPTIVO RLS

El filtro adaptivo que se implementará para la cancelación de una interferencia de banda ancha en una señal periódica, consta de: un filtro FIR como el filtro programable, y el algoritmo RLS para la actualización de los coeficientes de dicho filtro.

La señal de referencia necesaria para el proceso de cancelación adaptiva de ruido, se la obtiene como se estableció en los capítulos anteriores, a partir de la entrada principal, como una versión retardada de ésta. La señal de error que se realimenta al sistema, permite mediante la búsqueda del error mínimo, la determinación de los parámetros óptimos del filtro.

Lo expuesto anteriormente se sintetiza en el diagrama para el filtro RLS que consta en la figura 3.5.

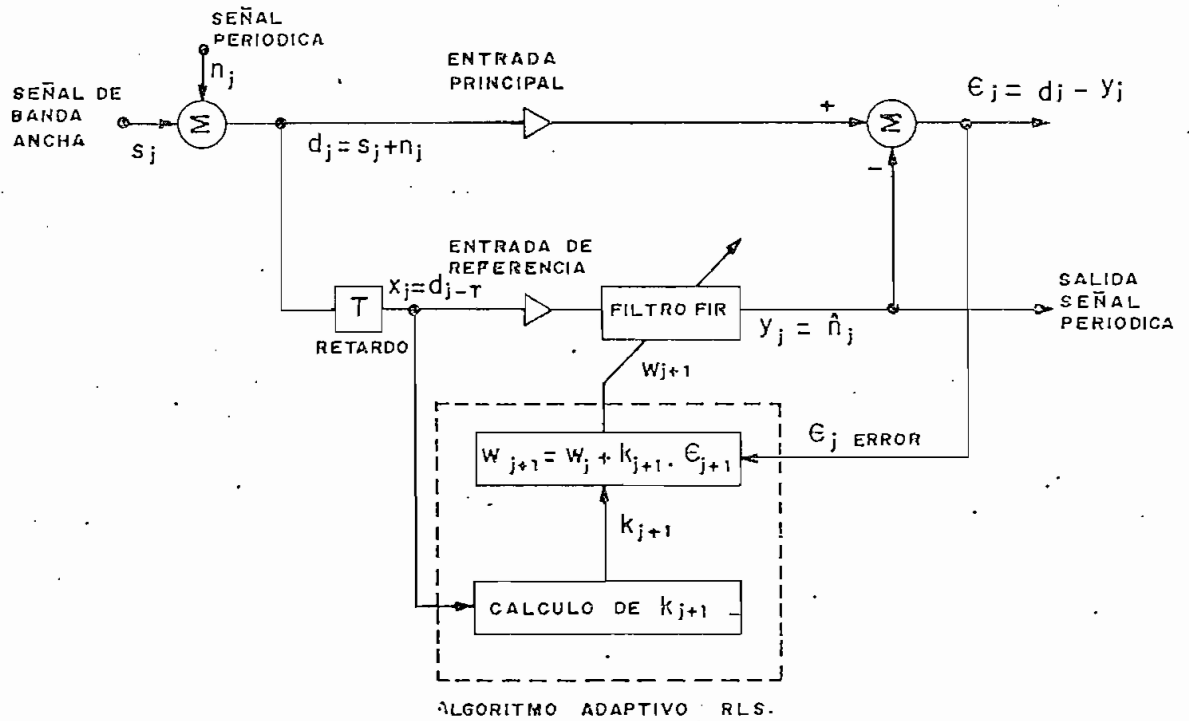


Fig. 3.5 Sistema de cancelación de interferencia de banda ancha en una señal periódica, mediante un filtro adaptivo RLS.

Al igual que en el algoritmo LMS, para el diseño de este tipo de filtro deben tomarse en consideración dos aspectos, así:

- deberá cumplirse con el teorema del muestreo, como se especifica en el capítulo II, sección 2.1.3.
- el orden del filtro está determinado por la resolución en frecuencia deseada, tal como se especifica en la sección 2.1.3.

3.2 ANALISIS DE CONVERGENCIA

La varianza de la salida del sistema de cancelación (error cuadrático medio) provee una medida útil del comportamiento del filtro adaptivo. La salida del sistema de cancelación (señal de error) para un vector de coeficientes arbitrario esta dada, según la ecuación (3.41), por:

$$\epsilon_j = d_j - X_j^T W \quad (3.103)$$

Tal ecuación puede descomponerse siempre en una suma del error de predicción óptimo, ϵ_j^* , (dado por el vector de coeficientes de ponderación óptimo) y otro término, así:

$$\epsilon_j = d_j - X_j^T W^* + X_j^T [W^{\hat{}} - W^*] \quad (3.104)$$

donde el error de predicción óptimo corresponde al término:

$$\epsilon_j^* = d_j - X_j^T W^*$$

Sacando el valor esperado del error al cuadrado se tiene:

$$E[\epsilon_j^2] = E[(\epsilon_j^*)^2] + E[(W - W^*)^T X_j X_j^T (W - W^*)] + 2 E[\epsilon_j^* X_j^T (W - W^*)] \quad (3.105)$$

donde el último término es cero, puesto que el error de predicción es no correlacionado con los datos pasados, según el principio de ortogonalidad. Entonces:

$$E[\epsilon_j^2] = \Gamma = \Gamma_{\min} + (W - W^*)^T R_{xx} (W - W^*) \quad (3.106)$$

La ecuación (3.106) puede usarse para evaluar la varianza de salida para cualquier vector de coeficientes fijo W , como era de esperarse la varianza de salida se minimizará sí y solamente sí $W = W^*$. Esta ecuación es análoga a la

ecuación (2.37) y constituye por tanto la expresión que permitirá dibujar la curva de aprendizaje.

La varianza de salida mínima Γ_{\min} puede evaluarse de la siguiente forma:

$$\begin{aligned} E[(\epsilon_j^*)^2] &= E[(d_j - X_j^T W^*)^2] \\ &= E[d_j^2] + E[(W^*)^T X_j X_j^T W^*] - 2E[(W^*)^T X_j d_j] \\ &= E[d_j^2] + (W^*)^T E[X_j X_j^T] W^* - 2(W^*)^T E[X_j d_j] \end{aligned} \quad (3.107)$$

de acuerdo con la expresiones vistas en el capítulo II (ecuaciones 2.8 y 2.9), $E[X_j X_j^T] = R_{xx}$ y $E[X_j d_j] = R_{xd}$, las cuales corresponden a la matriz de autocorrelación y al vector de correlación cruzada, respectivamente. Por lo tanto la ecuación (3.107) quedará:

$$\Gamma_{\min} = E[d_j^2] + (W^*)^T R_{xx} W^* - 2(W^*)^T R_{xd} \quad (3.108)$$

Usando la definición de W^* , ecuación (3.102), se tendrá:

$$\begin{aligned} \Gamma_{\min} &= E[d_j^2] + (W^*)^T R_{xx} W^* - 2(W^*)^T R_{xd} \\ &= E[d_j^2] - (W^*)^T R_{xd} \end{aligned}$$

la cual es una expresión análoga a la ecuación (2.36), desarrollada para el filtro adaptivo LMS (capítulo II).

3.3 IMPLEMENTACION DE PROGRAMAS

El desarrollo del algoritmo computacional correspondiente al método de los mínimos cuadrados recursivos consta de los siguientes pasos:

1.- Selección de Parámetros:

- Orden del filtro, n
- Número de retardos, T
- Factor de olvido, ϕ
- Parámetro de inicialización, σ

Al igual que en el algoritmo LMS, el orden del filtro máximo será 25.

El número de retardos que se elija igualmente deberá variar entre 1 y n .

Nuevamente como en el caso del algoritmo LMS, el número de muestras que puede manejar el programa es igual a 2000, por las razones expuestas en la sección 2.3 del capítulo II.

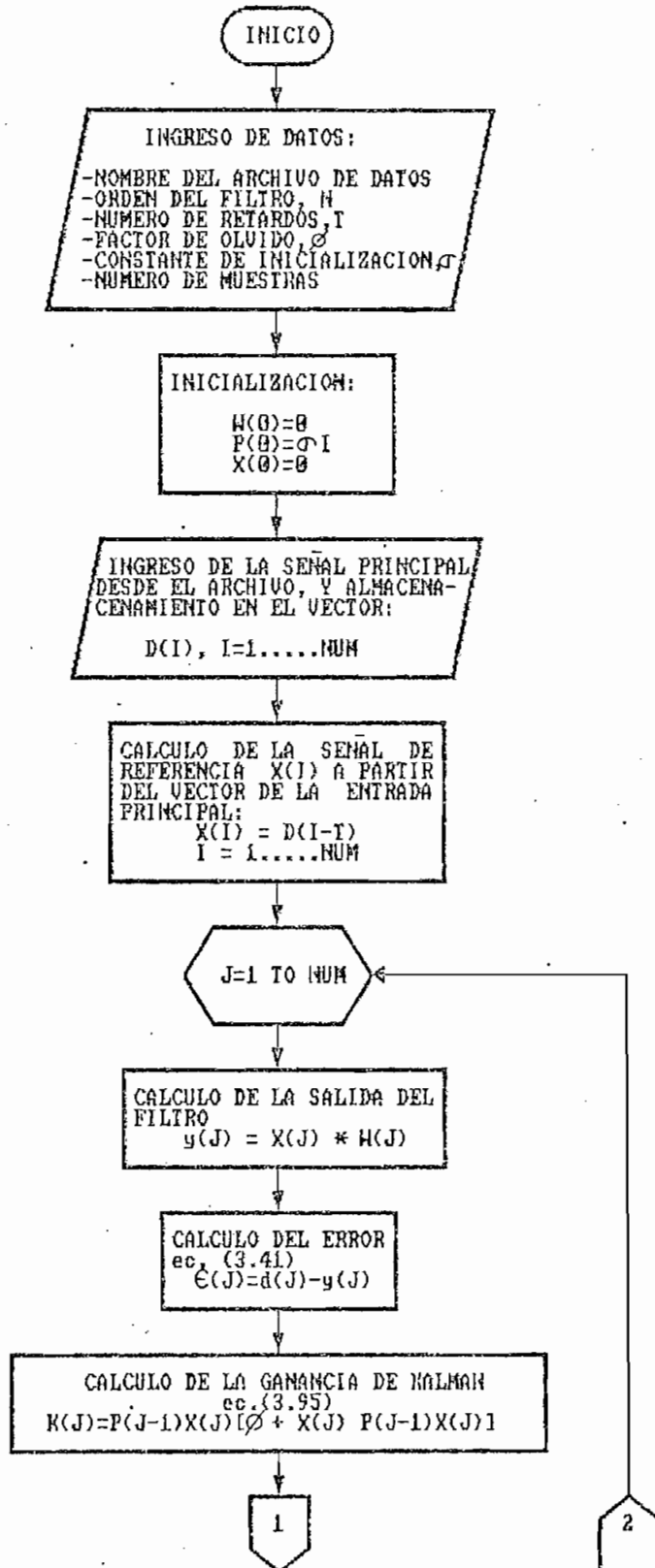
El factor de olvido, ϕ , deberá seleccionarse dependiendo de las características de la señal a procesarse, así pues si ésta es de variaciones rápidas es adecuado un valor de ϕ muy cercano a la unidad (0.999), en tanto que para señales de variaciones lentas un factor de olvido conveniente sería un valor menor (0.95).

2.- Inicialización:

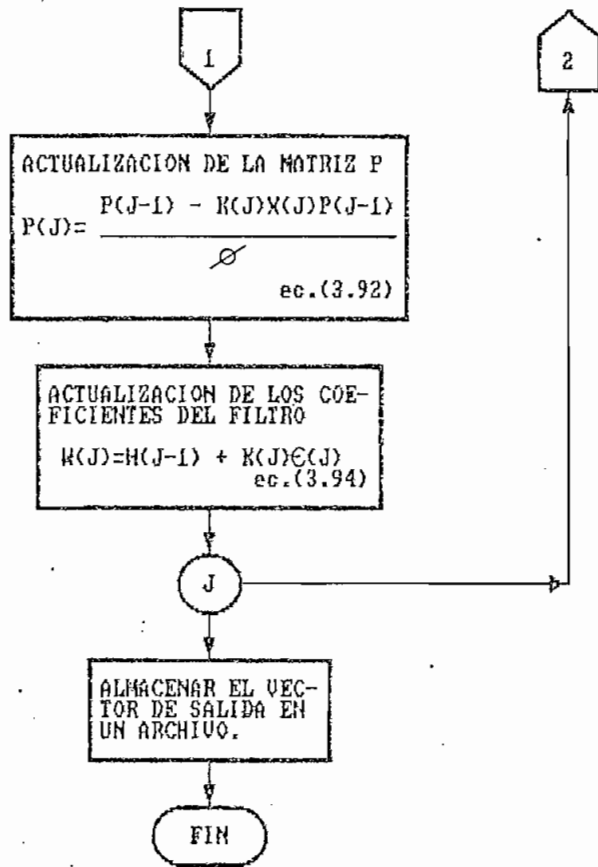
Se debe inicializar adecuadamente la matriz P_0 , el valor inicial de P puede obtenerse bien sea evaluando $[X_0^T X_0]^{-1}$ a partir de un bloque inicial de datos, o simplemente haciendo $P_0 = \sigma I$, donde σ es un número positivo grande que puede variar entre 10^2 y 10^4 , e I corresponde a la matriz identidad. Esta última alternativa se escogió para la inicialización de P_0 .

Al igual que en el algoritmo LMS, el vector de coeficientes de ponderación del filtro inicial se escogió igual a

ROUTINA DEL ALGORITMO RLS



...CONTINUACION



CAPITULO IV

RESULTADOS Y CONCLUSIONES

4.1 RESULTADOS EXPERIMENTALES

A fin de comprobar la validez de los programas implementados para simular la cancelación de una interferencia de banda ancha en una señal periódica, se presentan a continuación ejemplos ilustrativos con los cuales se intentará cubrir la mayor parte de las posibilidades que brindan los programas, los mismo que constituirán la base para en el numeral 4.2 realizar un análisis comparativo de los algoritmos desarrollados, y en el numeral 4.3 extractar las conclusiones correspondientes.

4.1.1 SEÑAL A PROCESARSE

4.1.1 a) Señal Periódica

La cual se genera en la forma expuesta en la sección 1.4 , a continuación se presentan dos ejemplos de los resultados obtenidos de la simulación de dicha señal periódica.

Ejemplo # 1

En este ejemplo se genera una onda sinusoidal de frecuencia 50 Hz. y amplitud unitaria, la frecuencia de muestreo seleccionada es de 1000 Hz.; con un tiempo de muestreo de 0.3 seg., dando un total de 300 muestras. Adjunto a la forma de onda de la señal generada se presenta el correspondiente análisis espectral.

Ejemplo # 2

En el cual se genera una onda periódica constituida por 3 sinusoides, así: primera senoide ($F_1 = 50$ Hz., $A = 1$), segunda senoide ($F_2 = 100$ Hz., $A = 1$) y, tercera senoide ($F_3 = 300$ Hz., $A = 1$), muestreada a 1000 Hz., durante 0.3 seg. En este caso también se realizó el análisis espectral que muestra claramente las componentes de frecuencia de la señal considerada.

4.1.1 b) Ruido blanco y coloreado

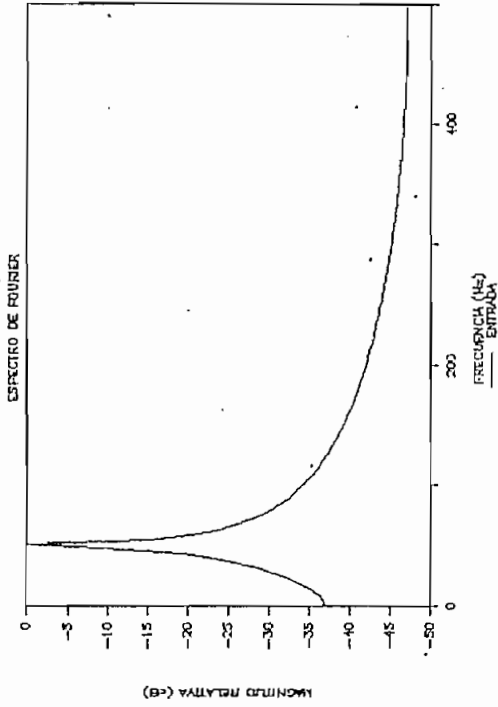
Ejemplo # 3

Muestra una señal de ruido blanco generada con varianza 0.5 y muestreada a 1000 Hz. El análisis espectral correspondiente refleja las características del ruido.

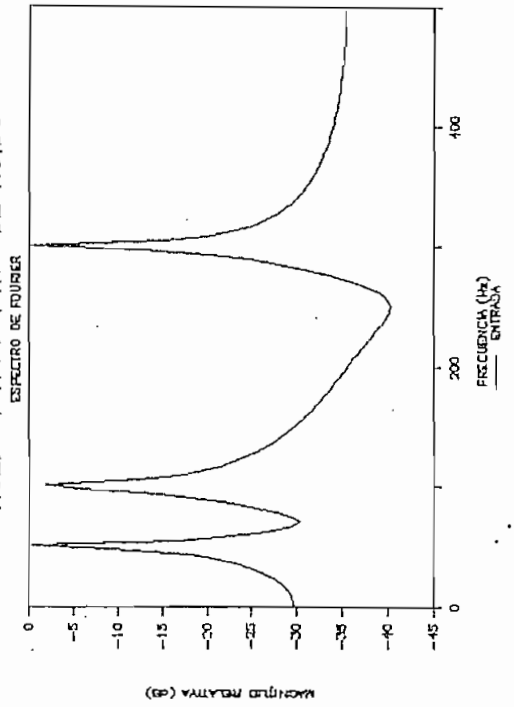
Ejemplo # 4, 5 y 6

Constituyen señales de ruido coloreado obtenidas a partir de muestras de ruido blanco (varianza 0.5, muestradas a 1000 Hz. durante 1.1 seg.), filtradas por filtros Butterworth pasabajos ($N = 10$, $F_c = 300$ Hz.), pasaltos ($N = 10$, $F_c = 300$ Hz.) y, pasabanda ($N = 10$, $F_1 = 100$ Hz. y $F_2 = 300$ Hz.) respectivamente. En el análisis espectral que se adjunta se muestran las frecuencias predominantes para cada uno de los tipos de ruido coloreado generados. Así para el ejemplo # 4 hay predominio de las bajas frecuencias, en el ejemplo # 5 en cambio son las altas frecuencias las significativas, en tanto que para el ejemplo # 6 la señal de ruido presenta solamente una banda de frecuencias componentes.

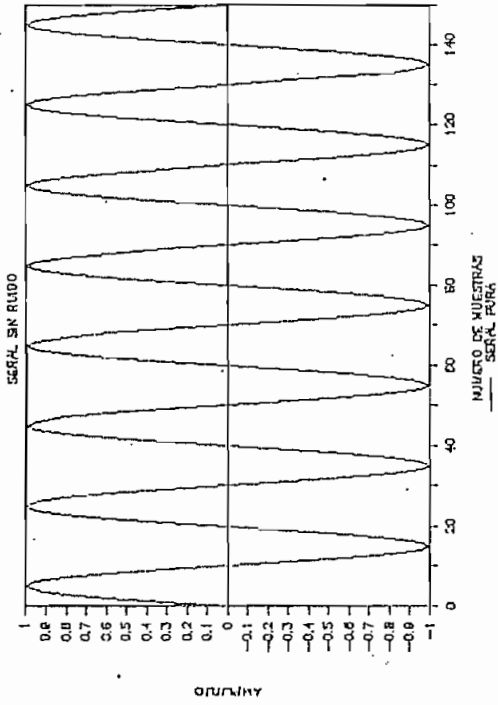
CANCELACION ADAPTIVA DE RUIDO



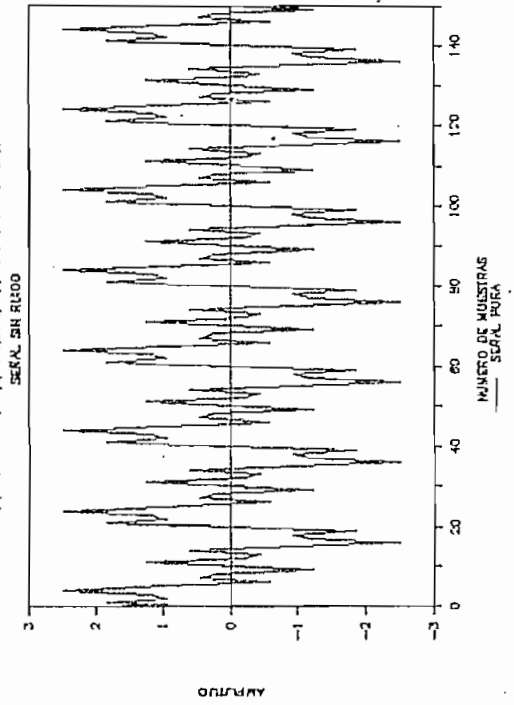
CANCELACION ADAPTIVA DE RUIDO



CANCELACION ADAPTIVA DE RUIDO



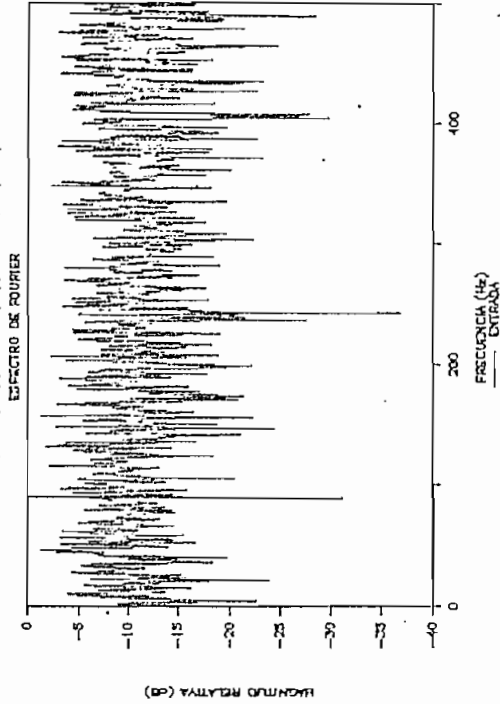
CANCELACION ADAPTIVA DE RUIDO



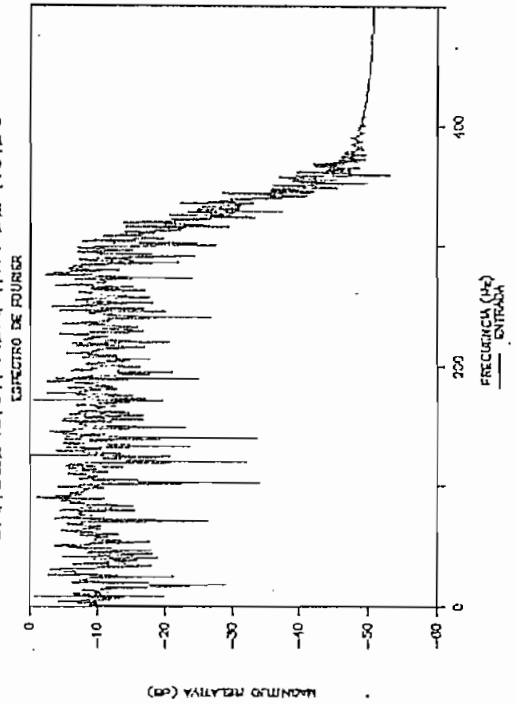
Ejemplo # 1
Señal sinusoidal.

Ejemplo # 2
Suma de 3 sinusoides.

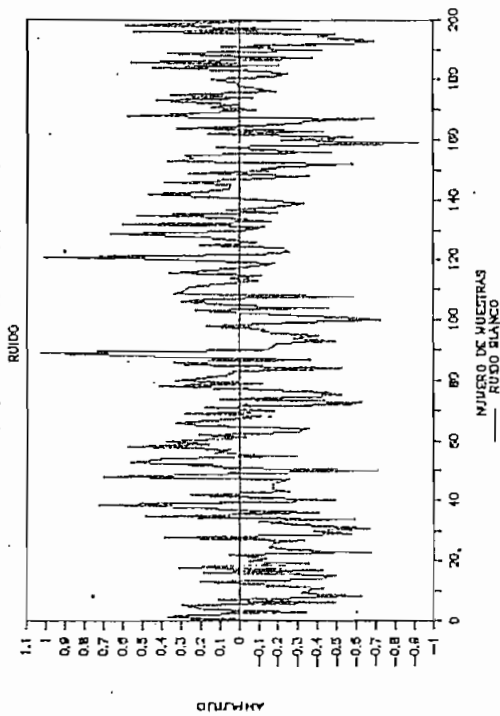
CANCELACION ADAPTIVA DE RUIDO



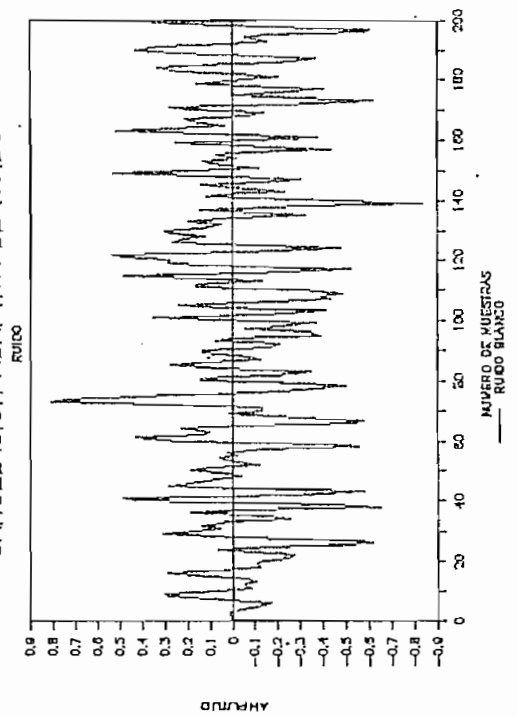
CANCELACION ADAPTIVA DE RUIDO



CANCELACION ADAPTIVA DE RUIDO



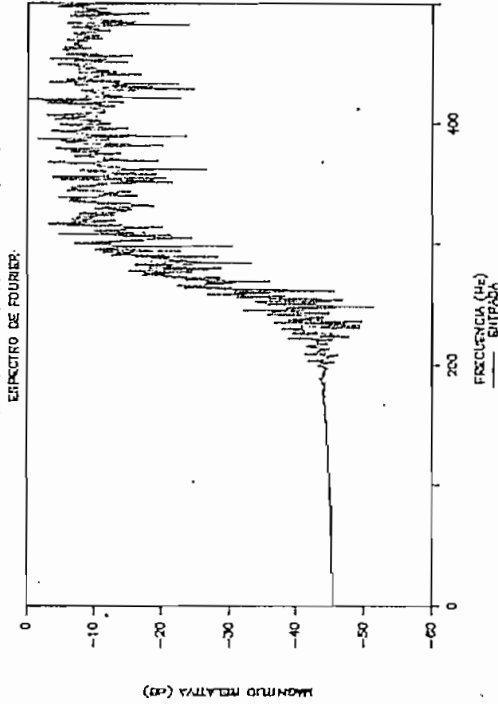
CANCELACION ADAPTIVA DE RUIDO



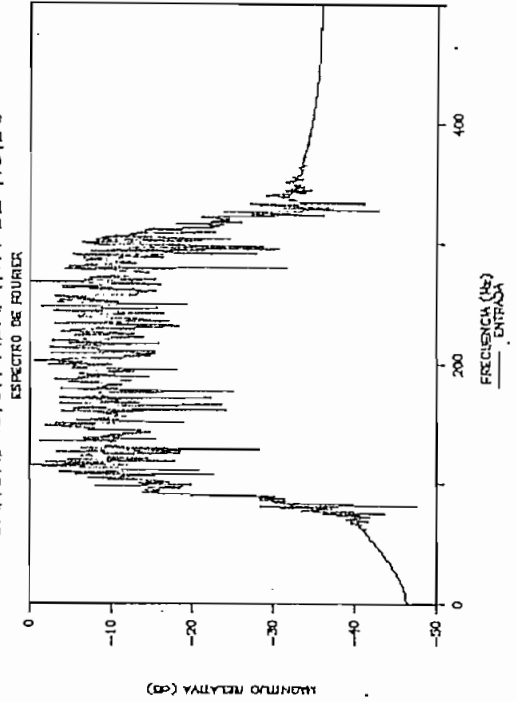
Ejemplo # 3
Ruido Blanco

Ejemplo # 4
Ruido Coloreado (Pasabajos)

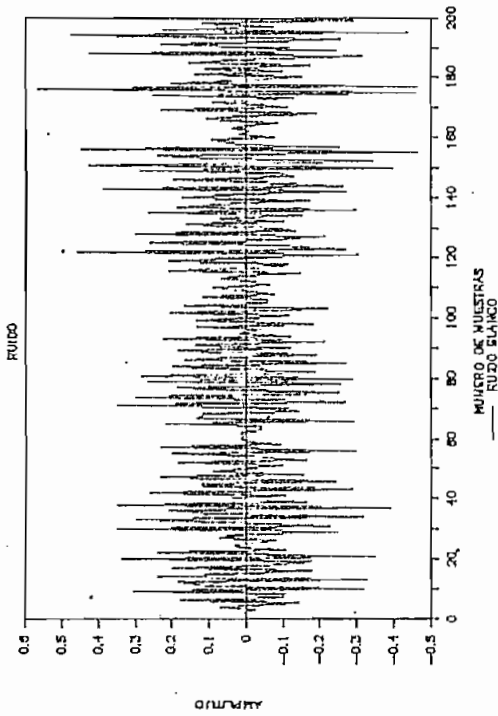
CANCELACION ADAPTIVA DE RUIDO



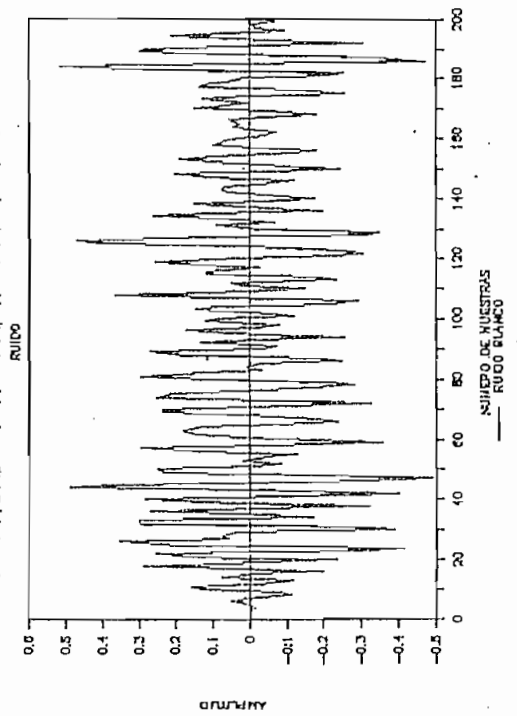
CANCELACION ADAPTIVA DE RUIDO



CANCELACION ADAPTIVA DE RUIDO



CANCELACION ADAPTIVA DE RUIDO



Ejemplo # 5

Ruido Coloreado (Pasaaltos)

Ejemplo #6

Ruido Coloreado (Pasabanda)

4.1.2 RESULTADOS SOBRE EL FILTRO ADAPTIVO LMS

En los ejemplos siguientes # 7, 8 y 9 se tratará de analizar el comportamiento del cancelador del ruido cuando la señal de entrada es de diferente naturaleza, así:

- señal contaminada con ruido blanco (ejemplo # 7)
- señal contaminada con ruido coloreado (ejemplo # 8)
- señal de potencia menor a la del ruido (ejemplo #9)

Ejemplo # 7

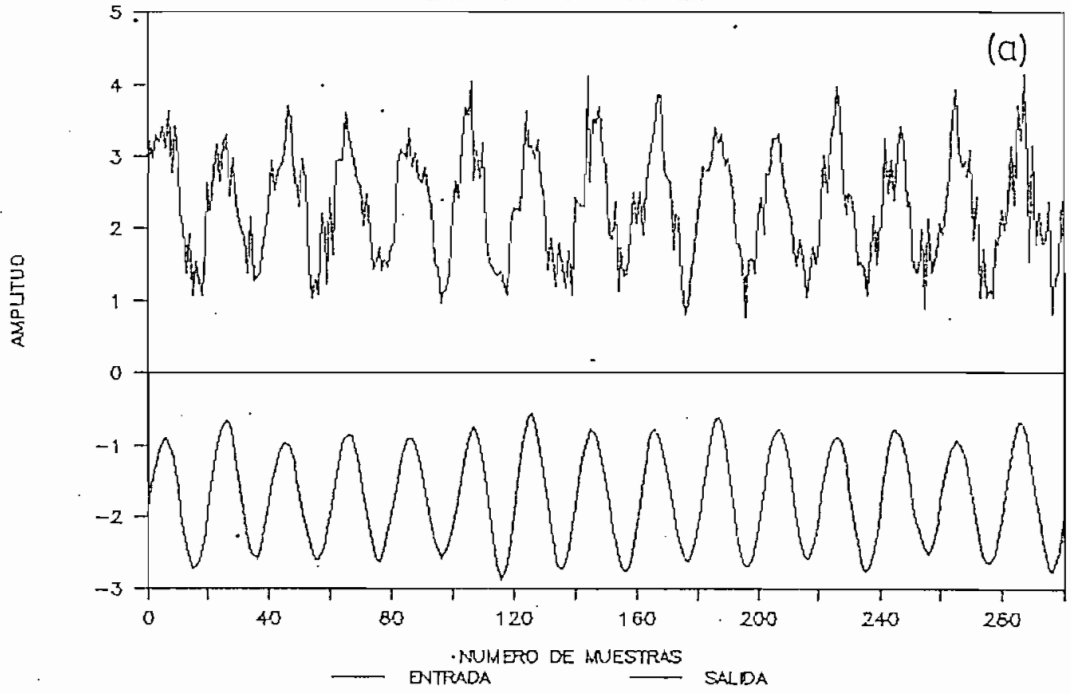
La señal a procesarse en este caso es una senoide de frecuencia igual a 500 Hz., amplitud unitaria, muestreada a 10 KHz. durante 0.1 seg, la cual está contaminada por una señal de banda ancha (ruido blanco de varianza 0.5). El filtro adaptivo utilizado fue de orden 16, y la señal de referencia se la obtuvo como una versión retardada de la entrada principal, la longitud del retardo fue 8 (mitad del orden del filtro), y el parámetro μ se seleccionó igual a $\mu_{\text{crítico}}/50^1$.

En este ejemplo al igual que en los ejemplos # 8 y # 9, se adjunta el análisis espectral correspondiente tanto de la señal de entrada al cancelador como de la señal de salida del mismo.

¹ El $\mu_{\text{crítico}}$ se toma como $1/3 \text{traza}[R_{xx}]$, lo cual no se contradice con la ecuación (2.32) del capítulo II, por el contrario un valor de μ menor al dado por dicha ecuación garantiza aún más la convergencia, este resultado se ha tomado de estudios recientes sobre el análisis de convergencia del algoritmo LMS [4].

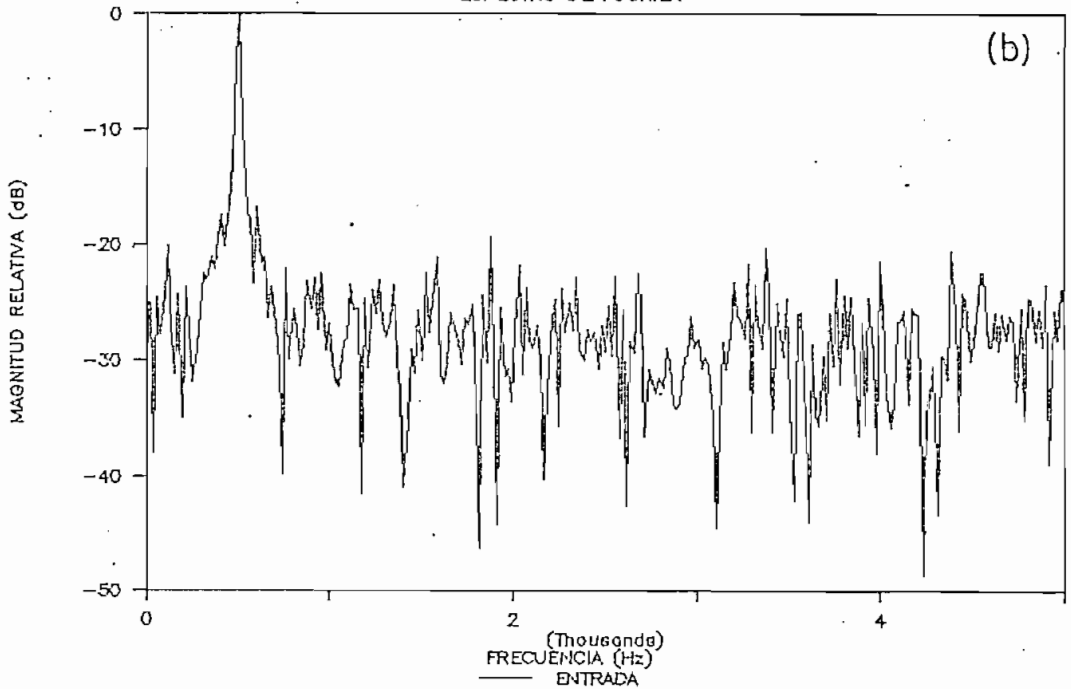
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

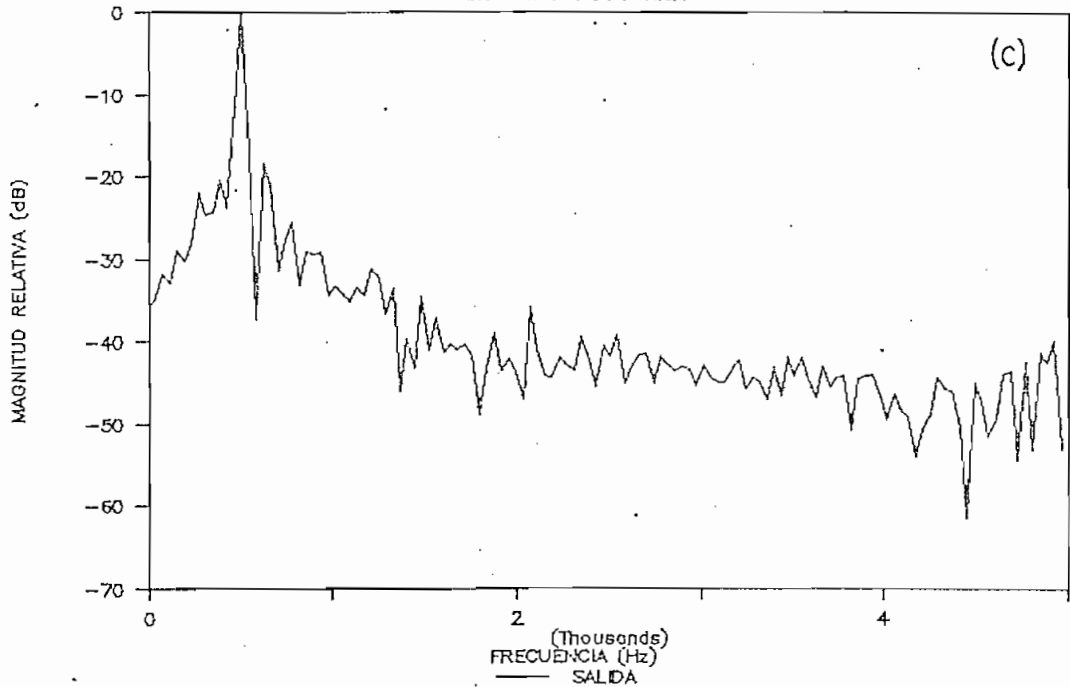


Ejemplo # 7

Procesamiento de una señal contaminada con ruido blanco. (a) Comparación entrada/salida del sistema. (b) Análisis espectral de la señal de entrada.

CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



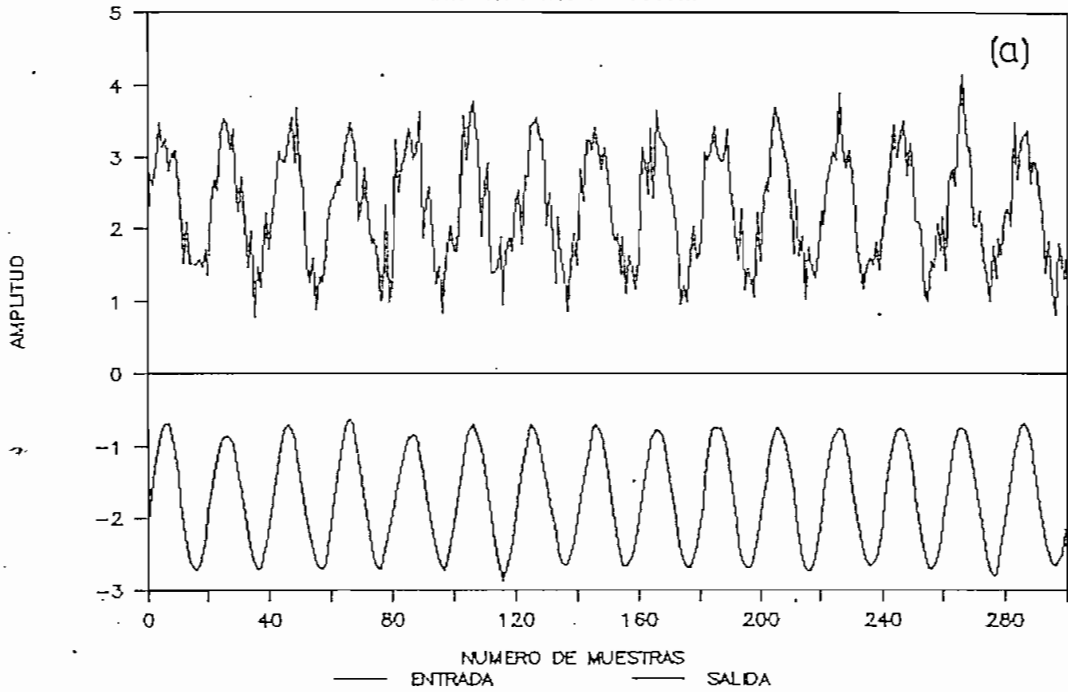
Ejemplo # 7 (c) Análisis espectral de la señal de salida.

Ejemplo # 8

La señal a procesarse en este caso es una senoide de iguales características a las del ejemplo # 7 ($f = 500$ Hz., amplitud = 1, frecuencia de muestreo = 10 KHz., tiempo de muestreo = 0.1 seg.); sin embargo el ruido que contamina la señal es ruido coloreado (filtrado con un Butterworth pasabajos, $F_c = 700$ Hz., $N = 10$). El filtro adaptivo es idéntico al del ejemplo # 7.

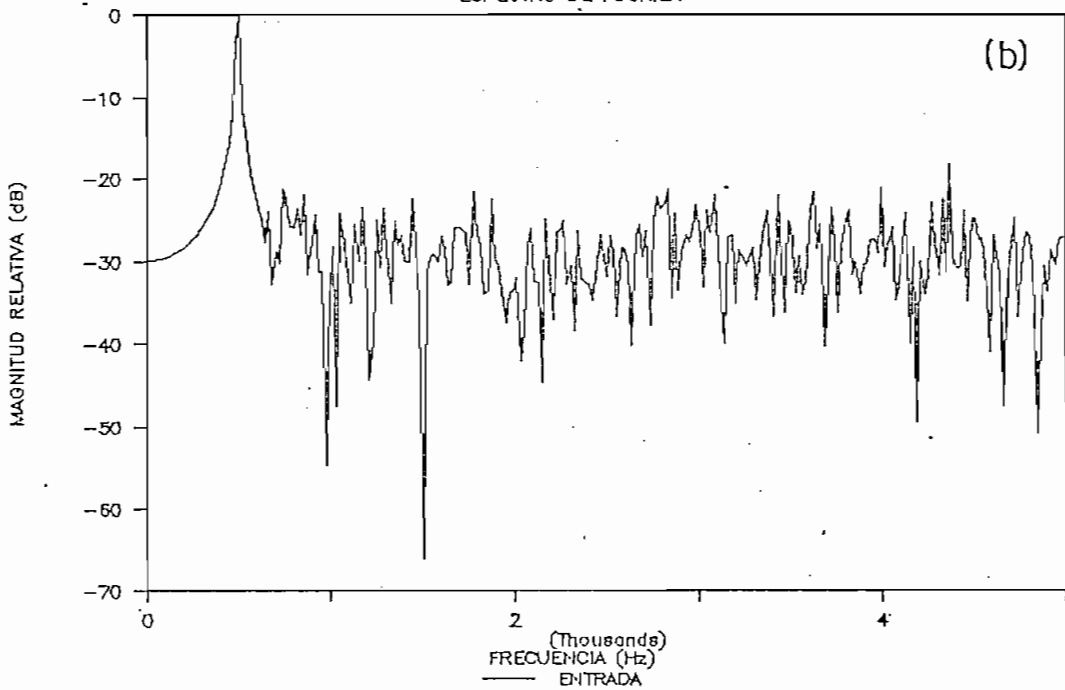
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

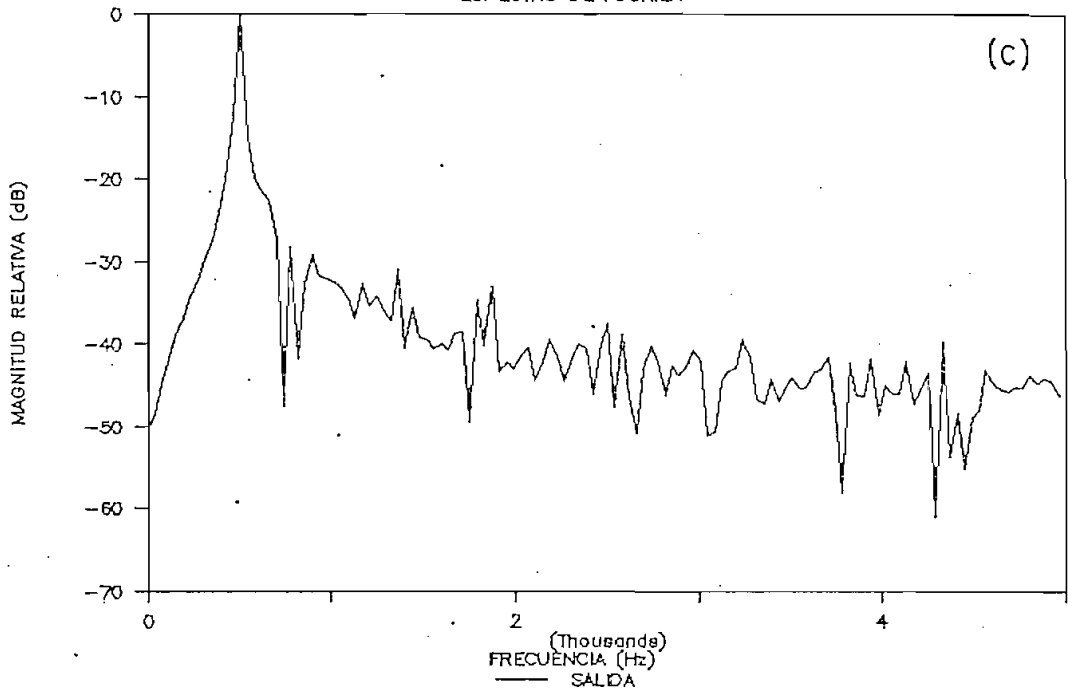


Ejemplo # 8

Procesamiento de una señal periódica contaminada con ruido coloreado. (a) Comparación entrada/salida del sistema. (b) Análisis espectral de la señal de entrada.

CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



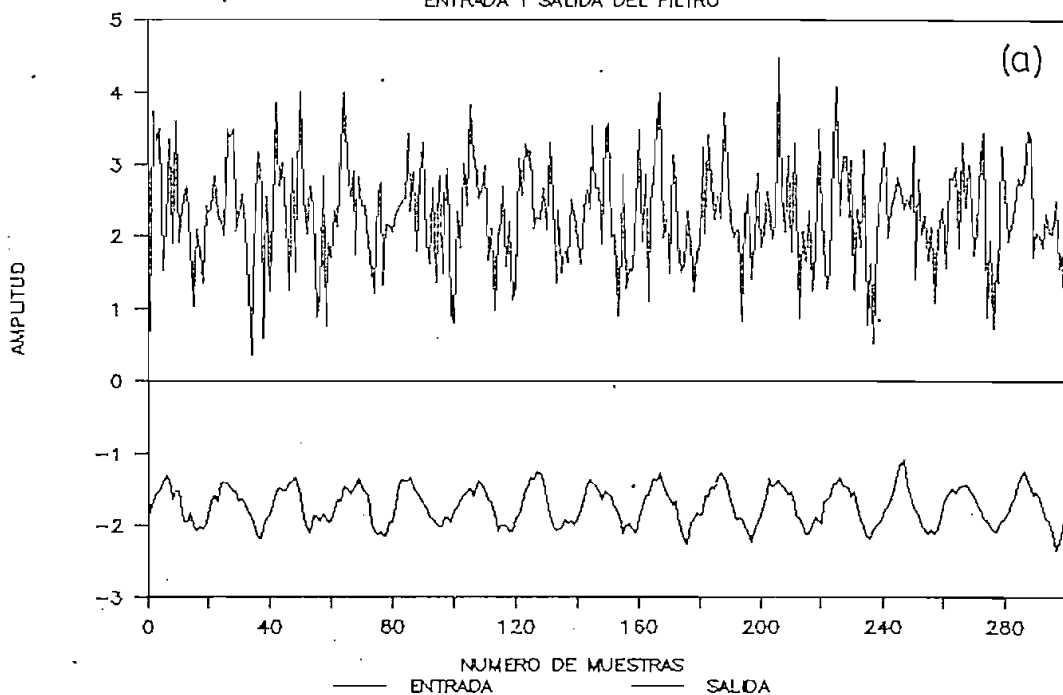
Ejemplo # 8 (c) Análisis espectral de la señal de salida.

Ejemplo # 9

Se trata de analizar en este ejemplo el comportamiento del cancelador de ruido frente a una señal periódica de potencia menor a la del ruido; así pues la onda periódica que se procesará es idéntica a la del ejemplo # 8, pero la amplitud de la señal en este caso es 0.5. El ruido coloreado se obtuvo de muestras de ruido blanco de varianza 2, filtradas con un Butterworth idéntico al del ejemplo # 8. El filtro adaptivo utilizado es el mismo del ejemplo # 8.

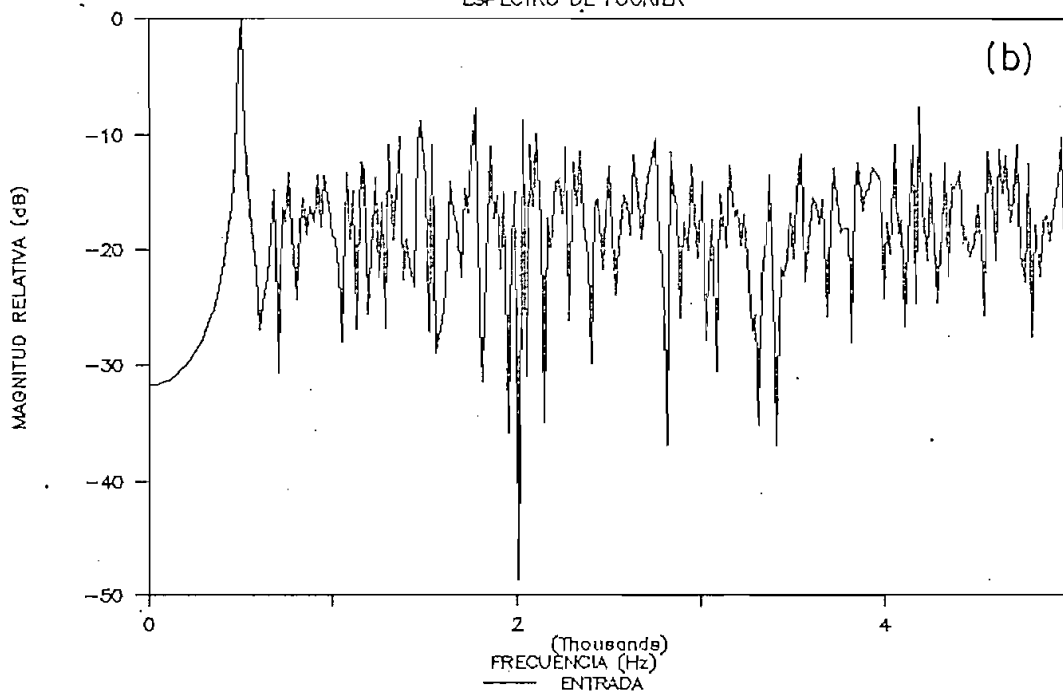
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

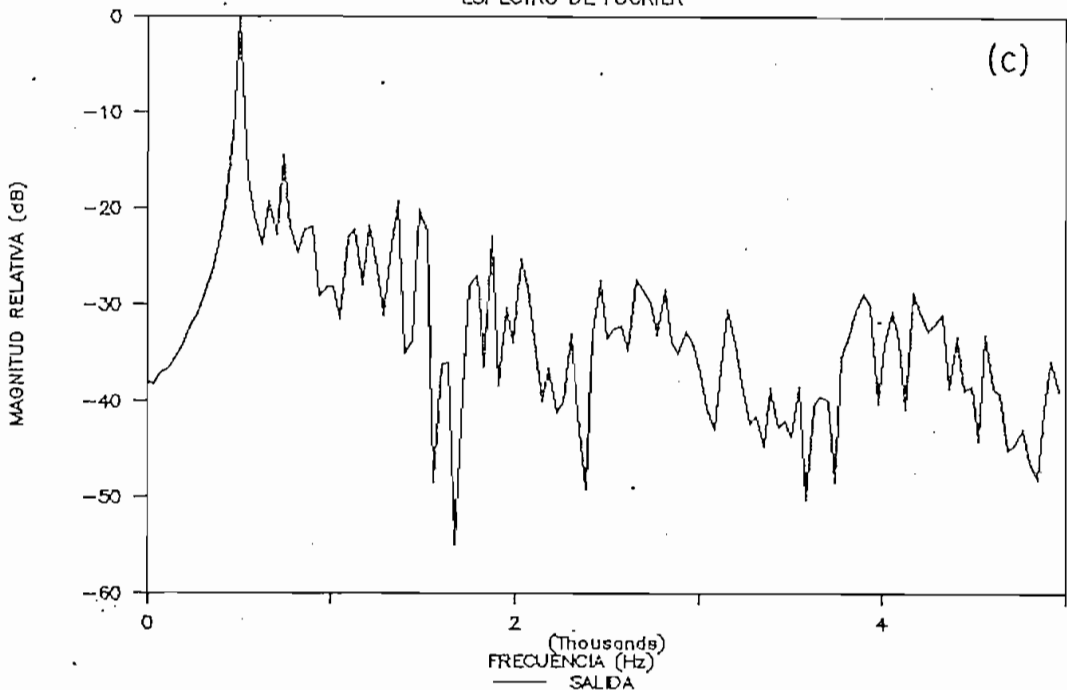


Ejemplo # 9

Procesamiento de una señal periódica de bajo nivel contaminada con ruido coloreado.
(a) Comparación entrada/salida del sistema.
(b) Análisis espectral de la señal de entrada.

CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



Ejemplo # 9 (c) Análisis espectral de la señal de salida.

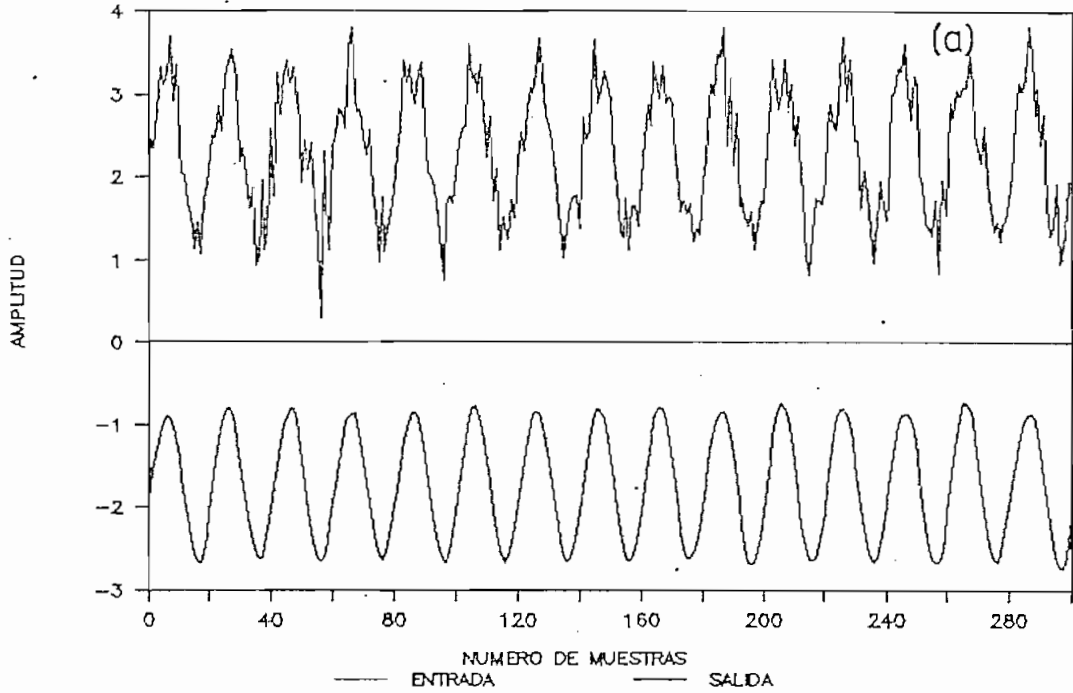
Ejemplo # 10, 11, 12 y 13

Se trata de analizar en estos ejemplos la influencia del parámetro μ en la convergencia del algoritmo LMS, y en el error en estado estable que se genera luego de la convergencia. La señal a procesarse es la misma del ejemplo # 8. El filtro adaptivo para los cuatro ejemplos es igual (orden 16, longitud del retardo 8); pero se ha variado en cada uno de los ejemplos el valor del parámetro de convergencia μ , así: ejemplo # 10 $\mu = \mu_0/100$, ejemplo # 11 $\mu = \mu_0/25$, ejemplo # 12 $\mu = \mu_0$, y ejemplo # 13 $\mu = 3.5\mu_0$.

Todos estos ejemplos se acompañan del análisis espectral tanto de la señal de entrada como de la de salida del cancelador y, de la curva de aprendizaje correspondiente.

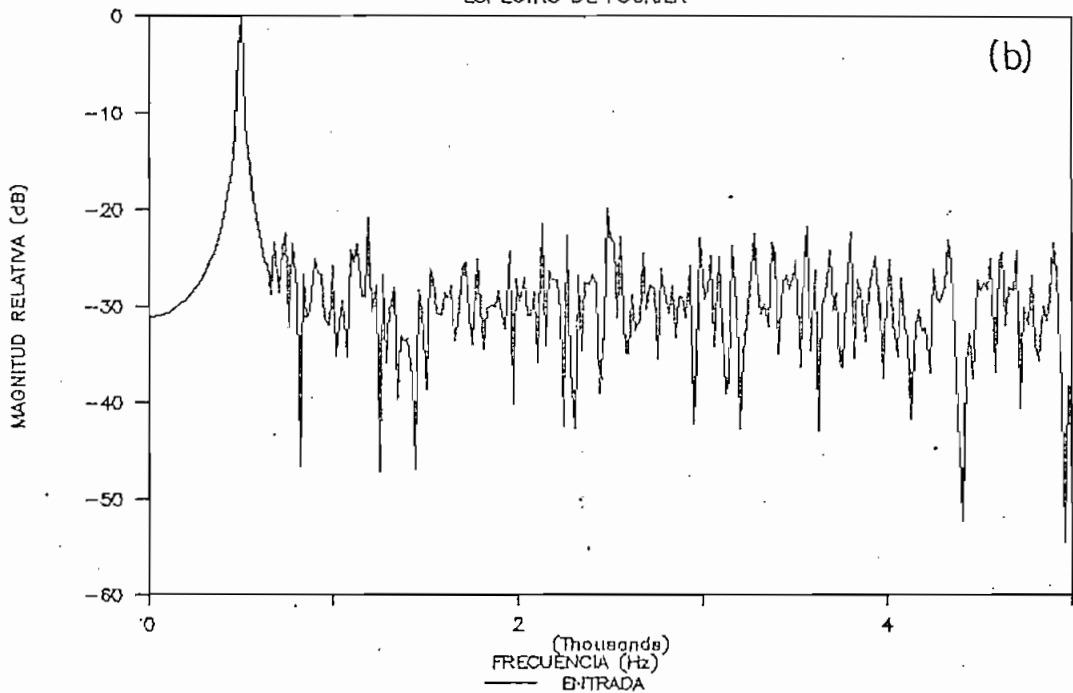
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

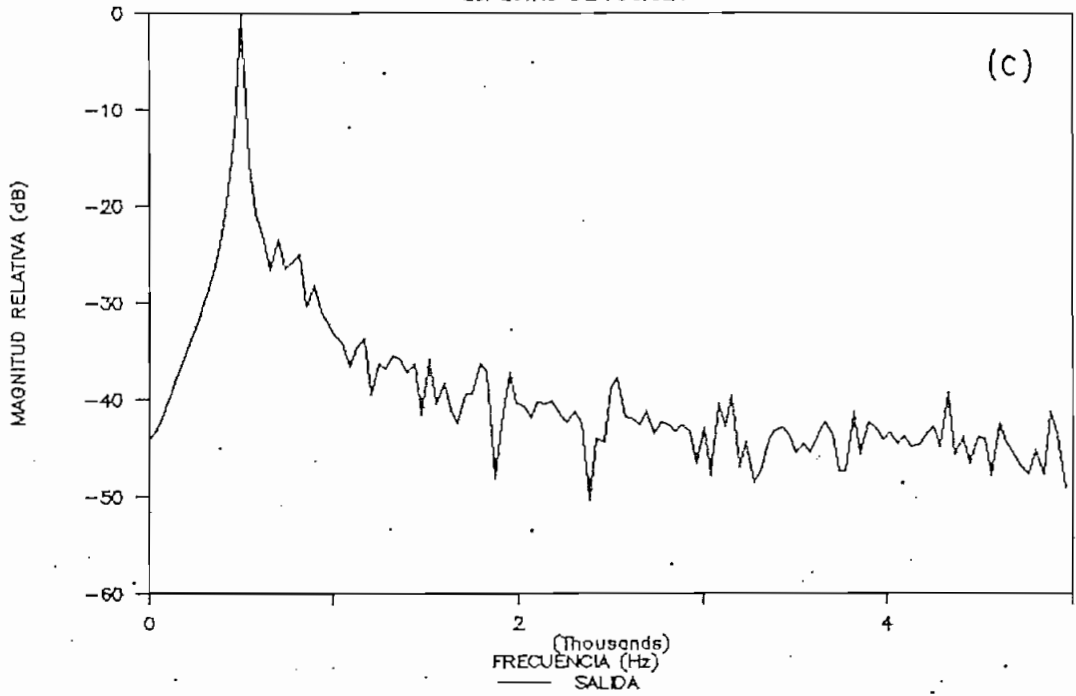
ESPECTRO DE FOURIER



Ejemplo # 10 Procesamiento de una señal periódica contaminada con ruido coloreado. $\mu = \mu_0/100$. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de entrada.

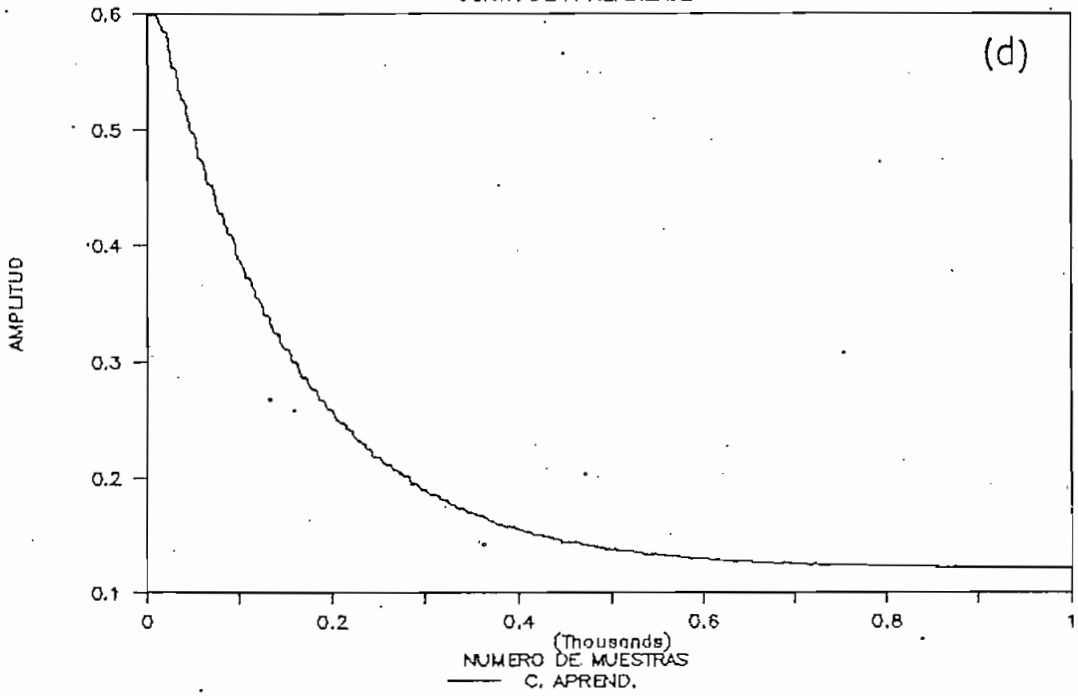
CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE

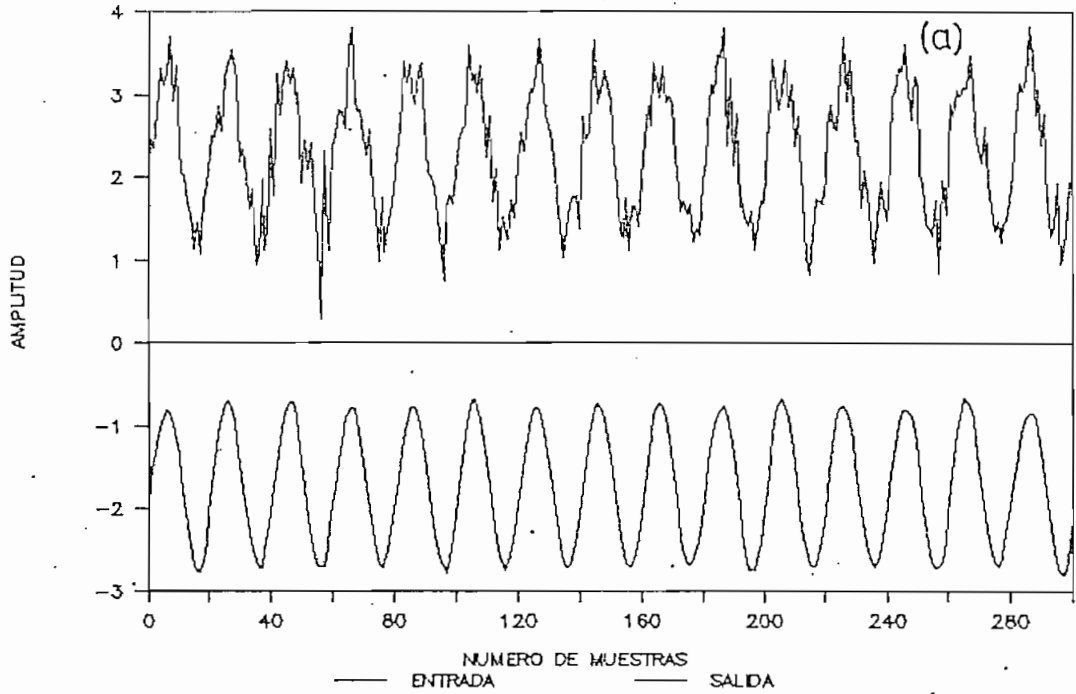


Ejemplo # 10

(c) Análisis espectral de la señal de salida. (d) Curva de aprendizaje.

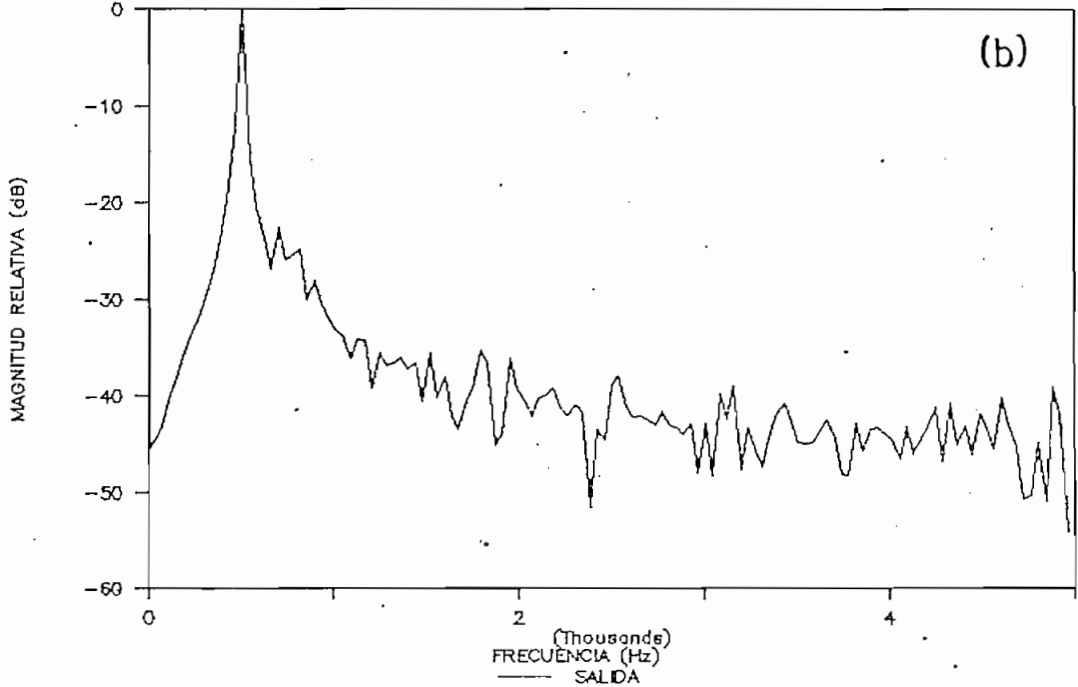
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

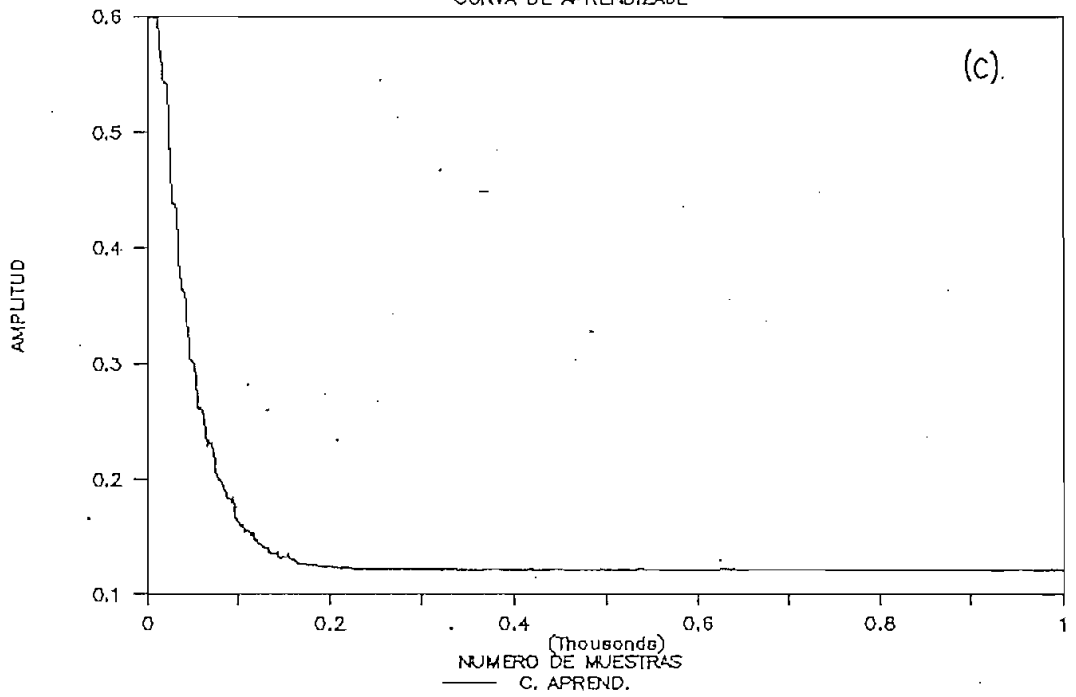
ESPECTRO DE FOURIER



Ejemplo # 11 Procesamiento de una señal contaminada con ruido coloreado. $\mu = \mu_c/25$. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

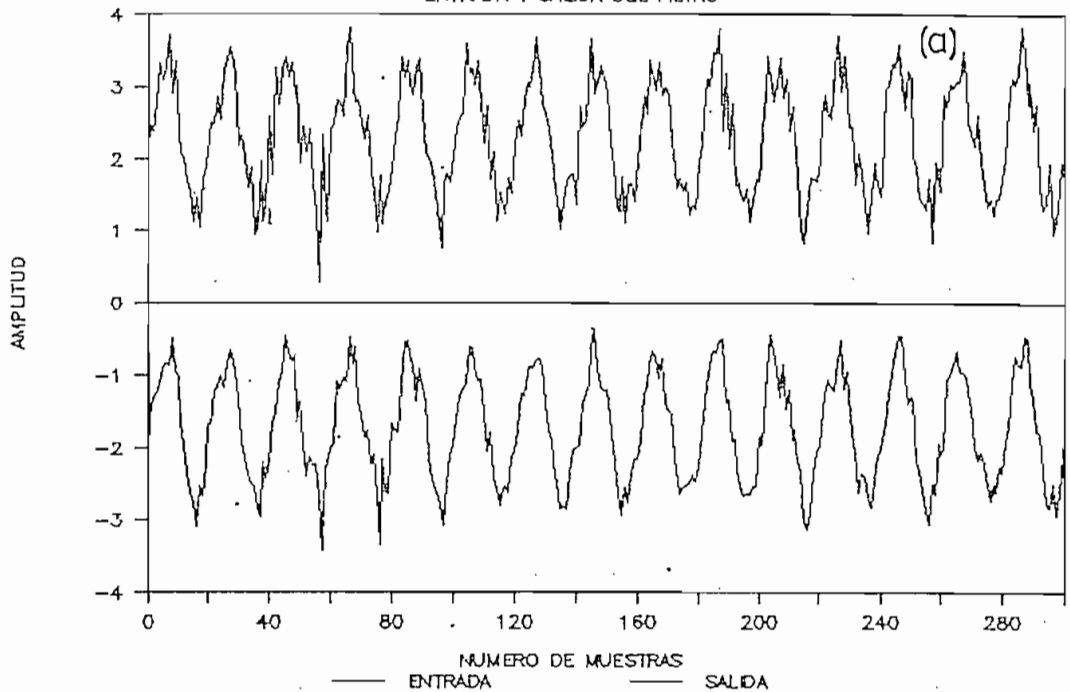
CURVA DE APRENDIZAJE



Ejemplo # 11 (c) Curva de aprendizaje.

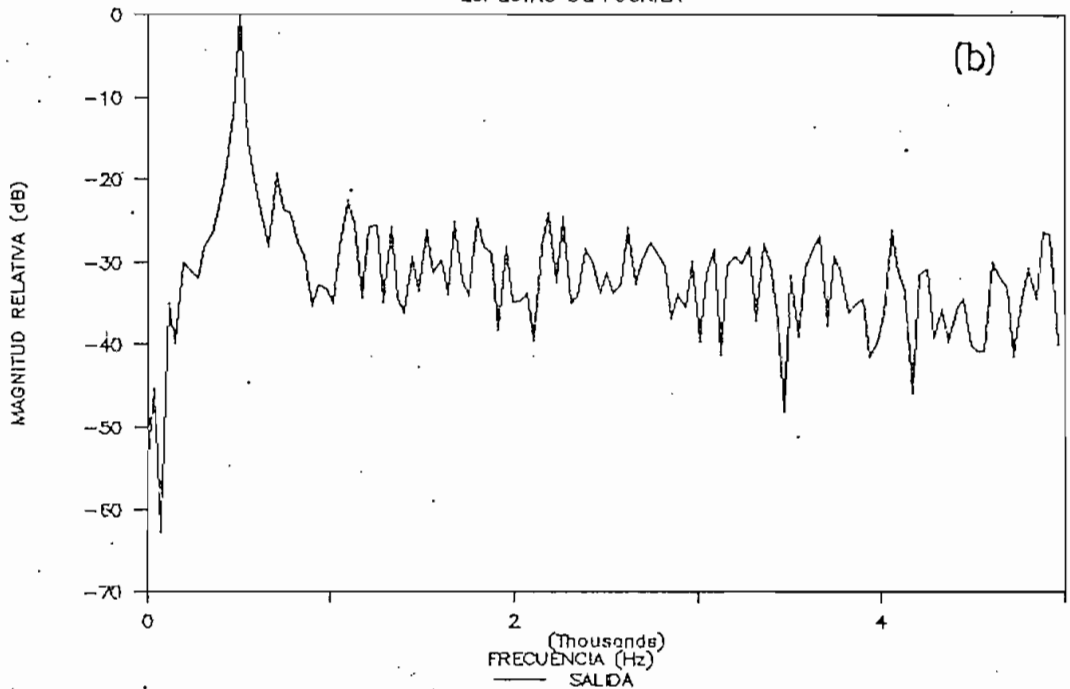
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

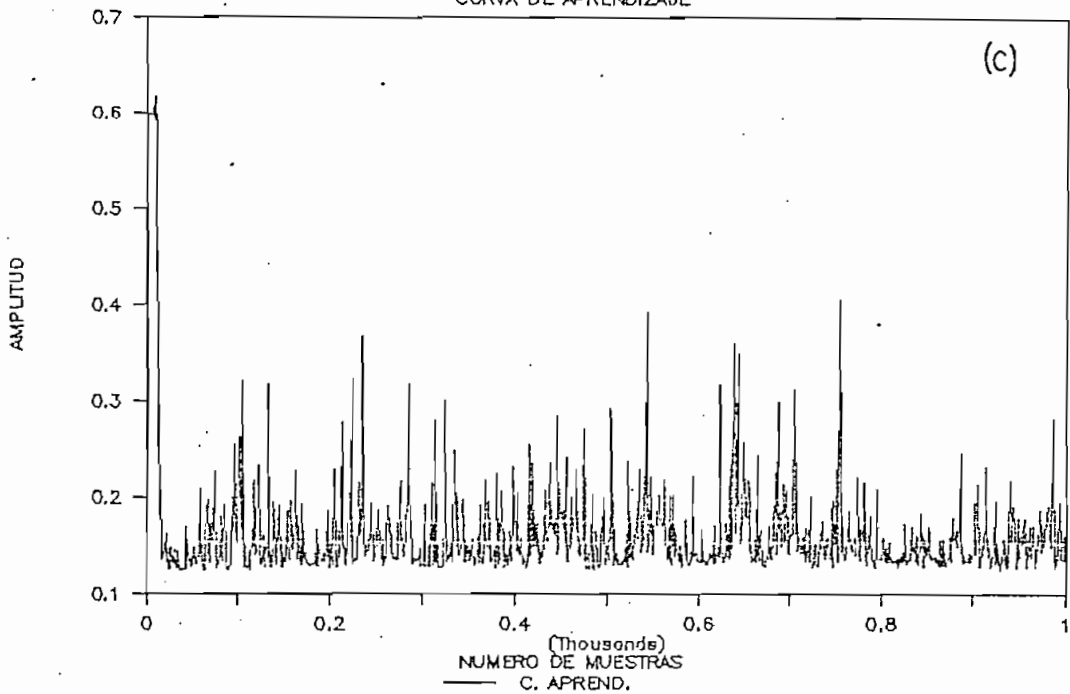
ESPECTRO DE FOURIER



Ejemplo # 12 Procesamiento de una señal contaminada con ruido coloreado. $\mu = \mu_0$. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE

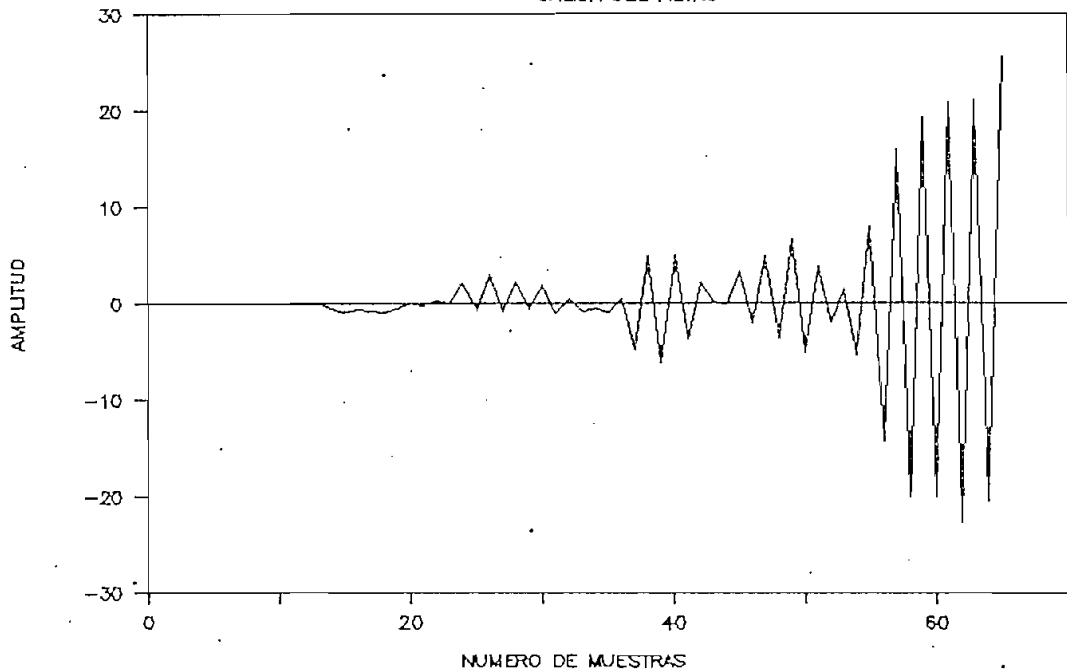


Ejemplo # 12 (c) Curva de aprendizaje.

En el caso del ejemplo # 13 debe hacerse una aclaración, el parámetro de convergencia μ se seleccionó igual a $35\mu_{critico}$, tal valor lógicamente produce divergencia, esto lo muestra el gráfico del ejemplo # 13, el cual corresponde a la señal de salida del filtro, y donde no aparece la señal de entrada en razón de la gran diferencia de escalas existente entre estas dos señales.

CANCELACION ADAPTIVA DE RUIDO

SALIDA DEL FILTRO



Ejemplo # 13 Procesamiento de una señal contaminada con ruido coloreado. $\mu = 3.5\mu_e$.

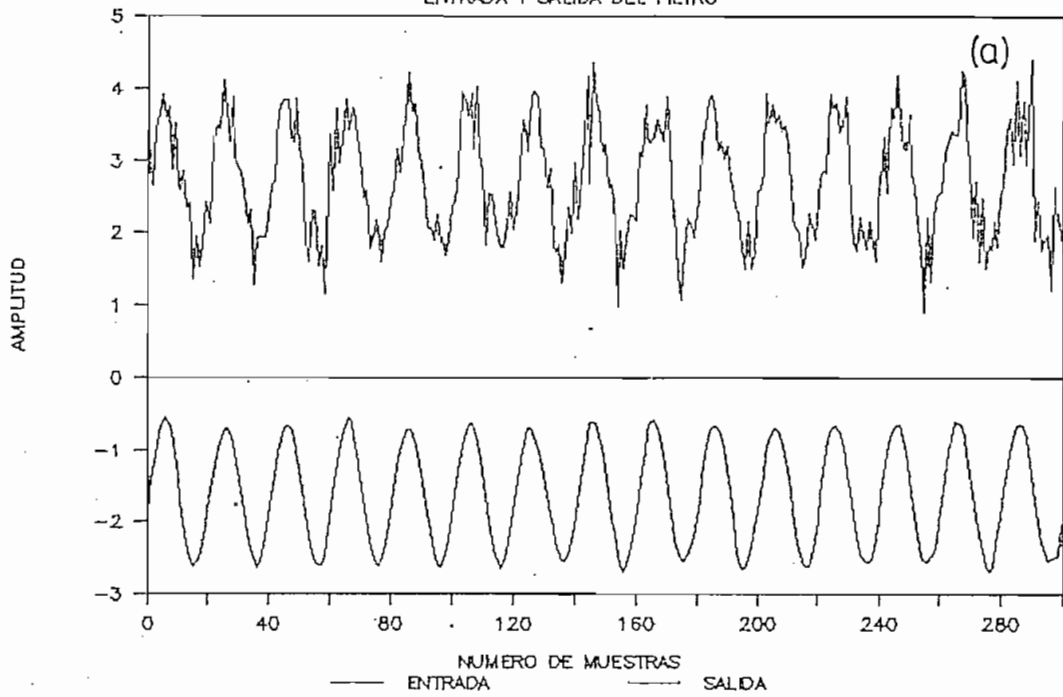
Ejemplos # 14, 15 y 16

Con estos ejemplos se pretende analizar la influencia del orden del filtro en la calidad de la señal procesada. La señal a tratarse es la misma que se usó en los ejemplos anteriores (ejemplos 10,11,12 y 13). El filtro utilizado fue de diferente orden para cada caso, con la correspondiente variación en la longitud del retardo; así pues para el ejemplo # 14 se trabajó con $N = 16$, $T = 8$ (T es el número de retardos); para el ejemplo # 15 con $N = 10$, $T = 5$; y para el ejemplo # 16 con $N = 6$, $T = 3$. En todos los casos el parámetro de convergencia fue igual a $\mu_e/50$.

Adicionalmente se obtuvo para cada uno de los ejemplos las curvas de aprendizaje y el análisis espectral respectivo.

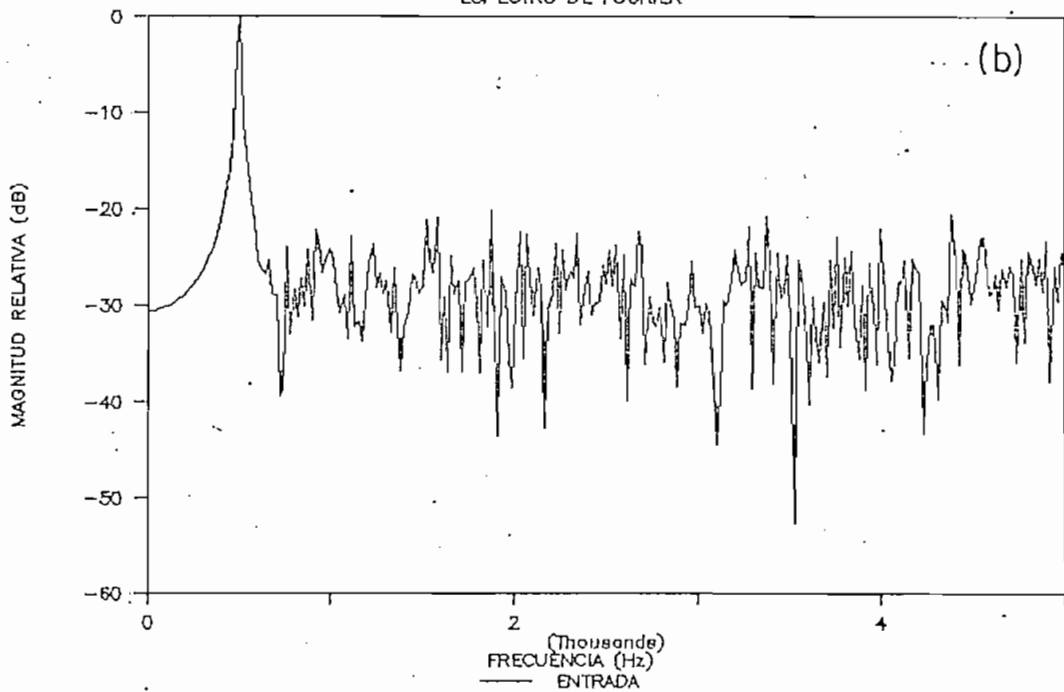
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

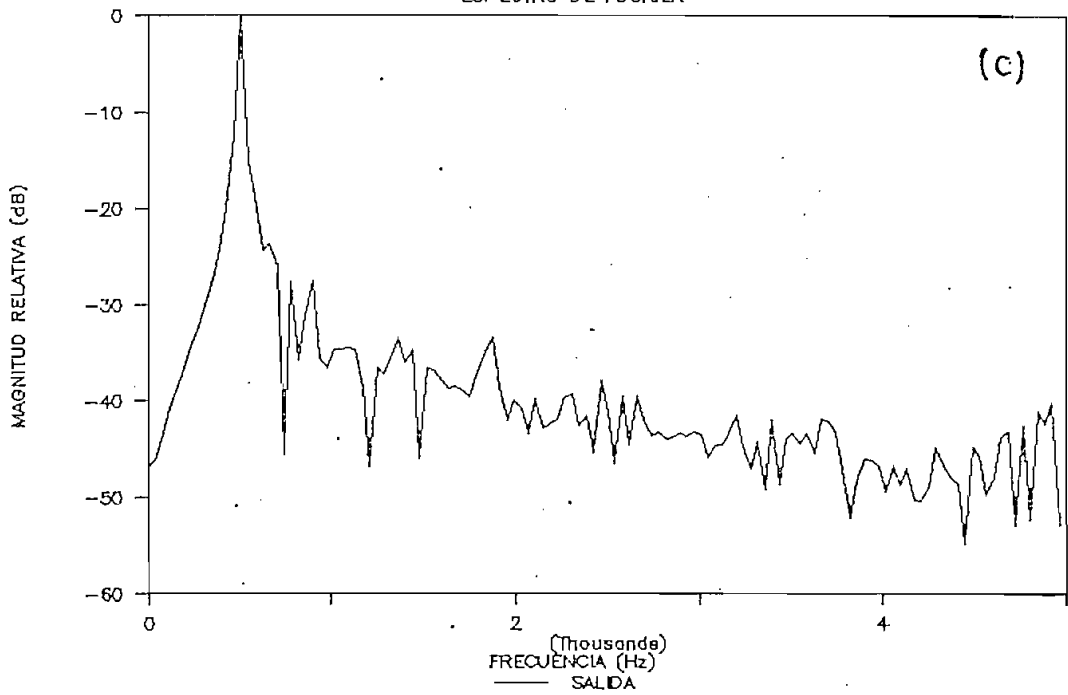
ESPECTRO DE FOURIER



Ejemplo # 14 Procesamiento de una señal contaminada con ruido coloreado. $N=16$, $T=8$. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de entrada.

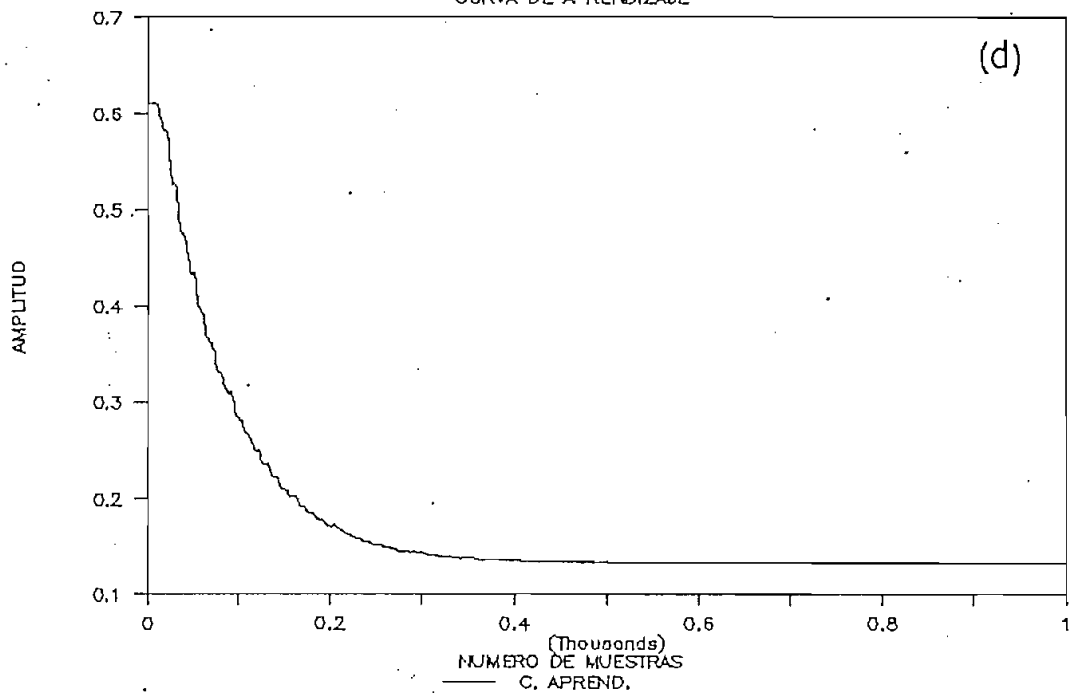
CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



CANCELACION ADAPTIVA DE RUIDO

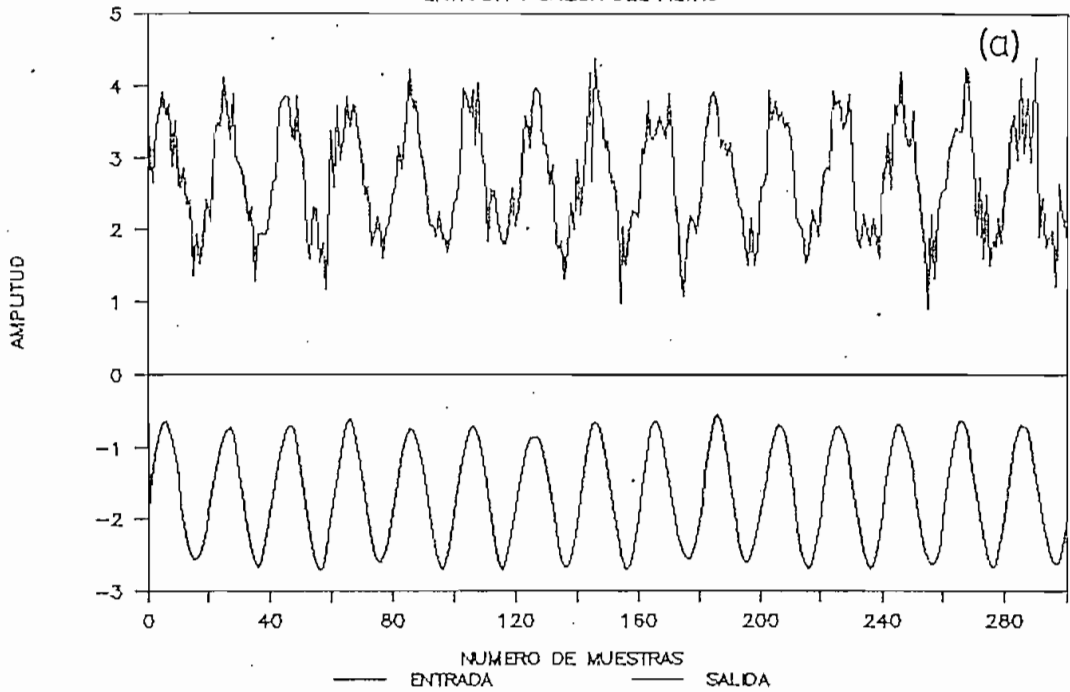
CURVA DE APRENDIZAJE



Ejemplo # 14 (c) Análisis espectral de la señal de salida. (d) Curva de aprendizaje.

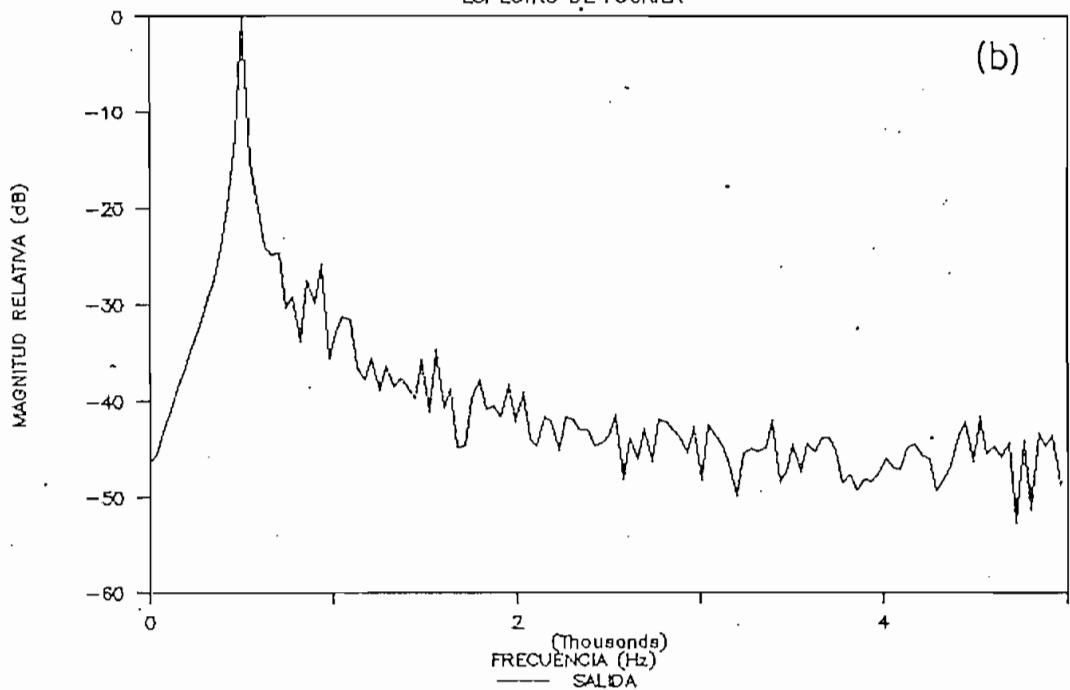
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

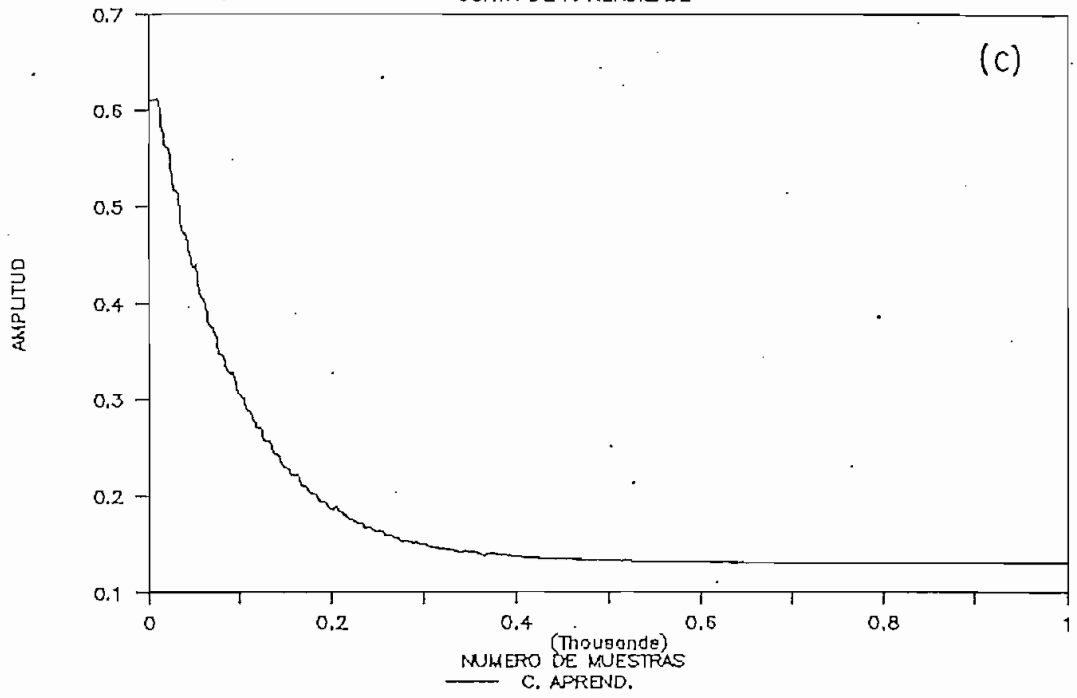
ESPECTRO DE FOURIER



Ejemplo # 15 Procesamiento de una señal contaminada con ruido coloreado. $N = 10$, $T = 5$. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

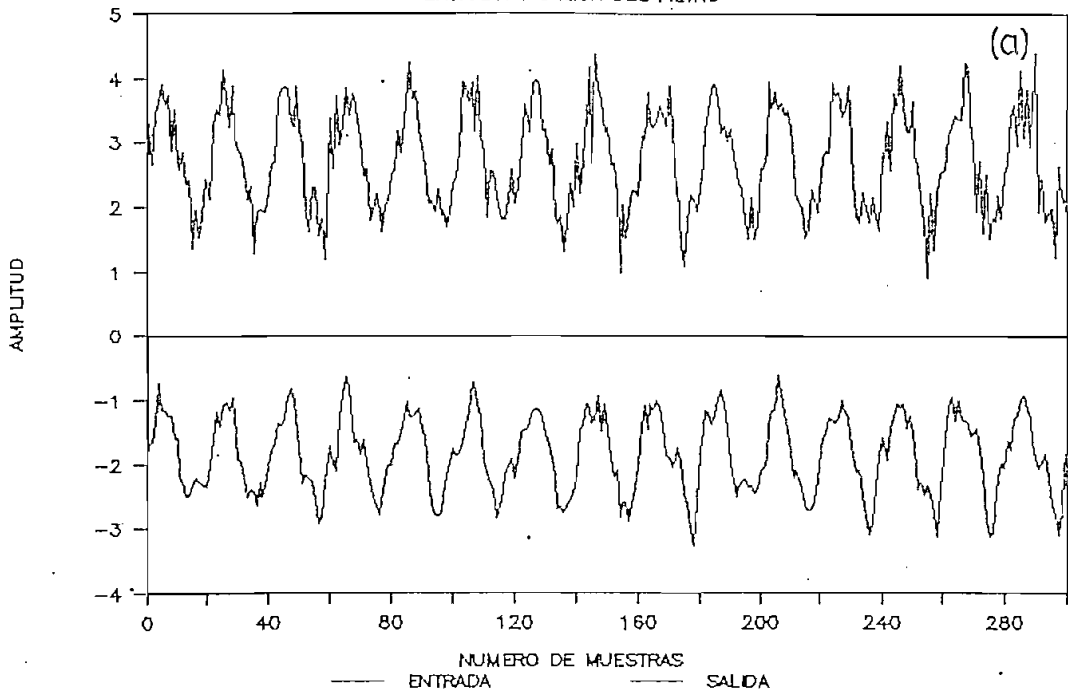
CURVA DE APRENDIZAJE



Ejemplo # 15 (c) Curva de aprendizaje.

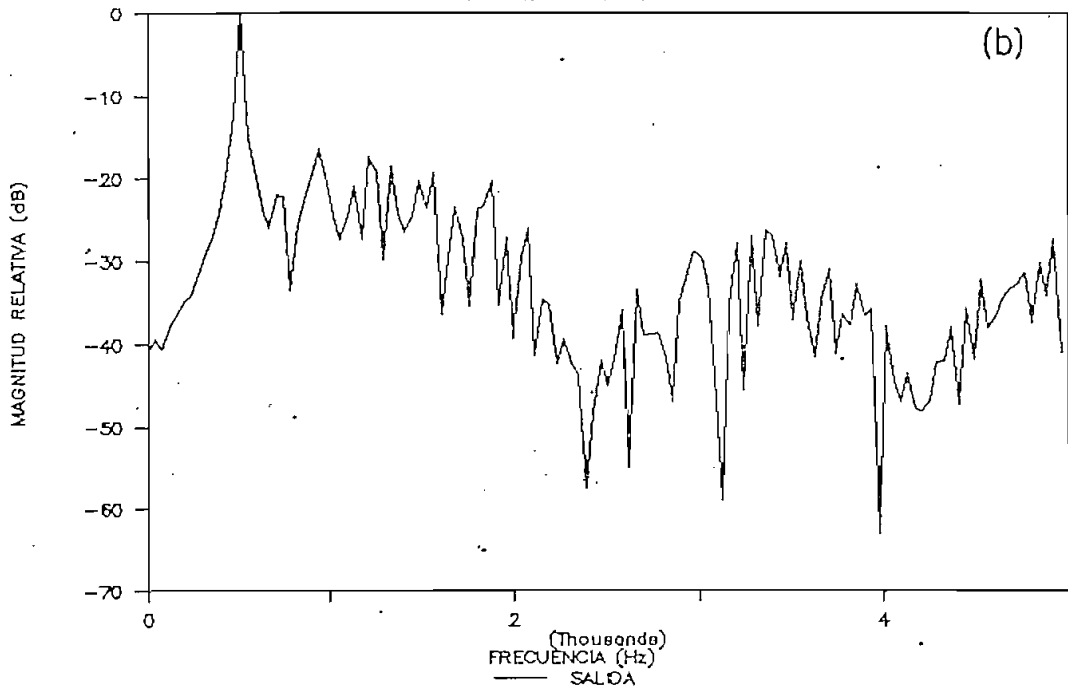
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

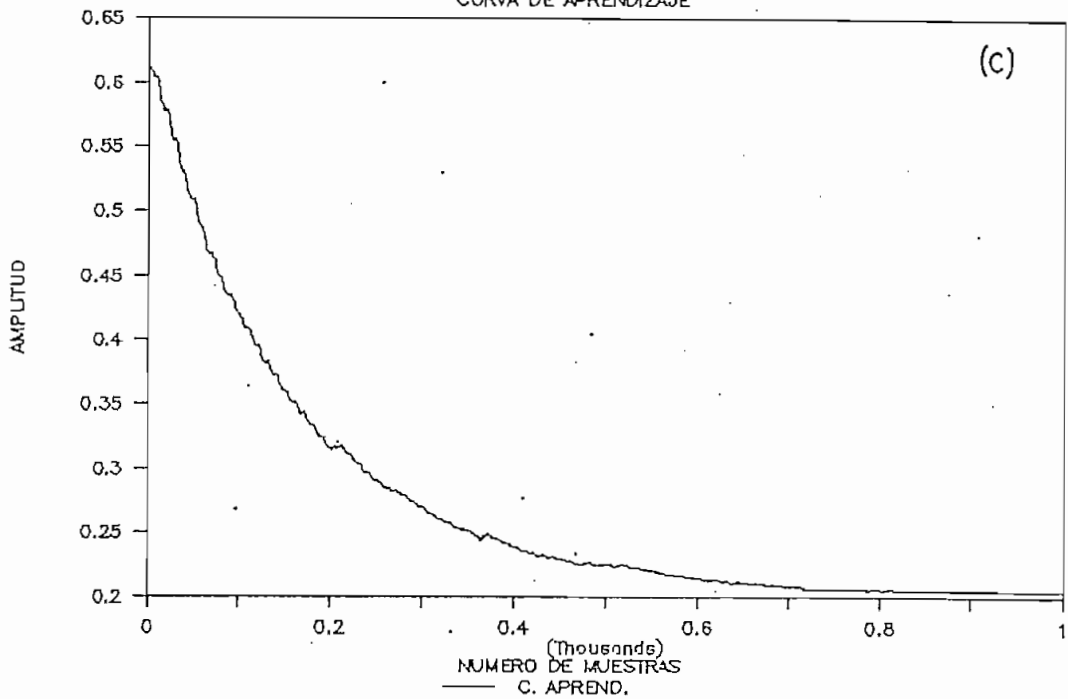


Ejemplo # 16

Procesamiento de una señal periódica contaminada con ruido coloreado. $N = 6$, $T = 3$.
(a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE

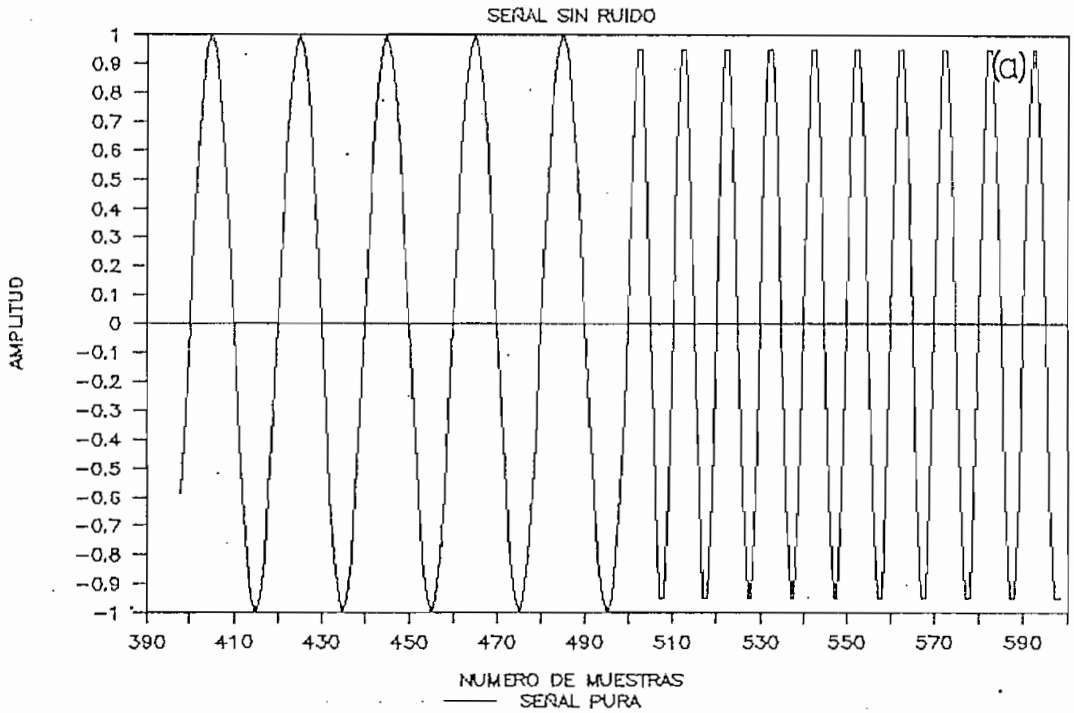


Ejemplo # 16 (c) Curva de aprendizaje.

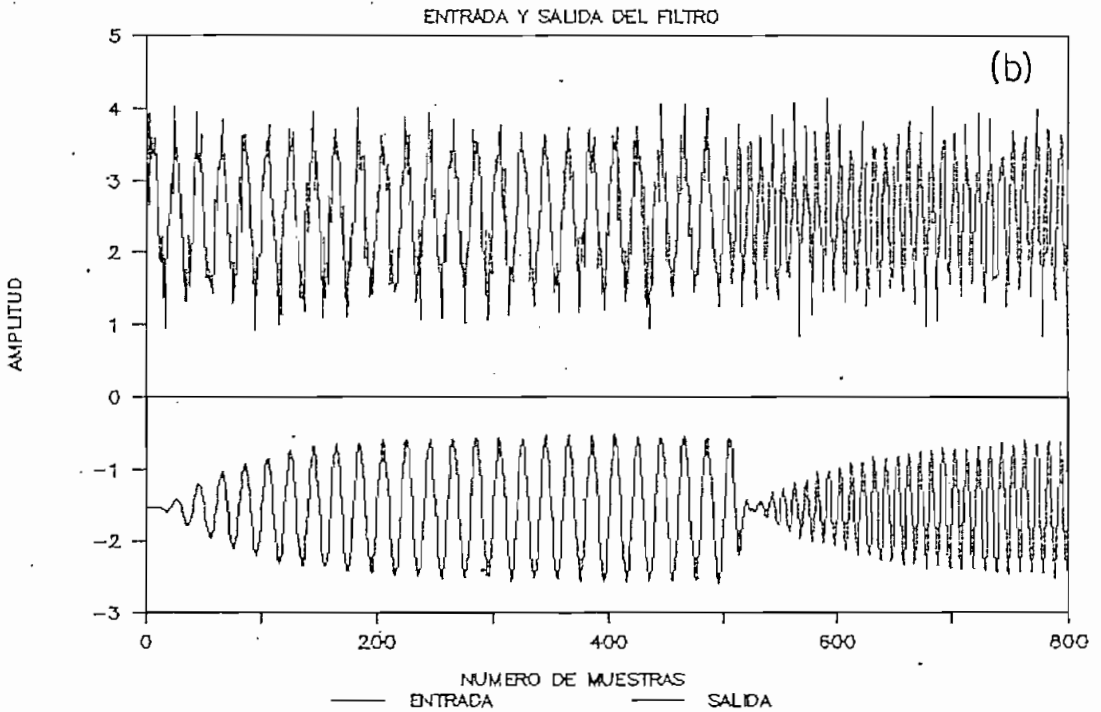
Ejemplo # 17

Con este ejemplo se trata de comprobar si el filtro realmente se ajusta frente a cambios en las características de la señal. Se procesó entonces una señal conformada por dos bloques: el primer bloque constituido por una señal de 500 Hz. y amplitud unitaria, el segundo bloque por una señal de 1000 Hz. y amplitud así mismo unitaria; la frecuencia de muestreo seleccionada fue de 10 KHz. y el tiempo de muestreo de 0.05 seg. para cada bloque; a esta señal se le sumó ruido blanco (varianza 0.5) filtrado con un Butterworth pasaaltos ($F_c = 1500$ Hz. y $N = 10$). El filtro adaptivo para este ejemplo fue de orden 16 y $T = 8$, con un parámetro de convergencia $\mu = \mu_0/25$.

CANCELACION ADAPTIVA DE RUIDO



CANCELACION ADAPTIVA DE RUIDO



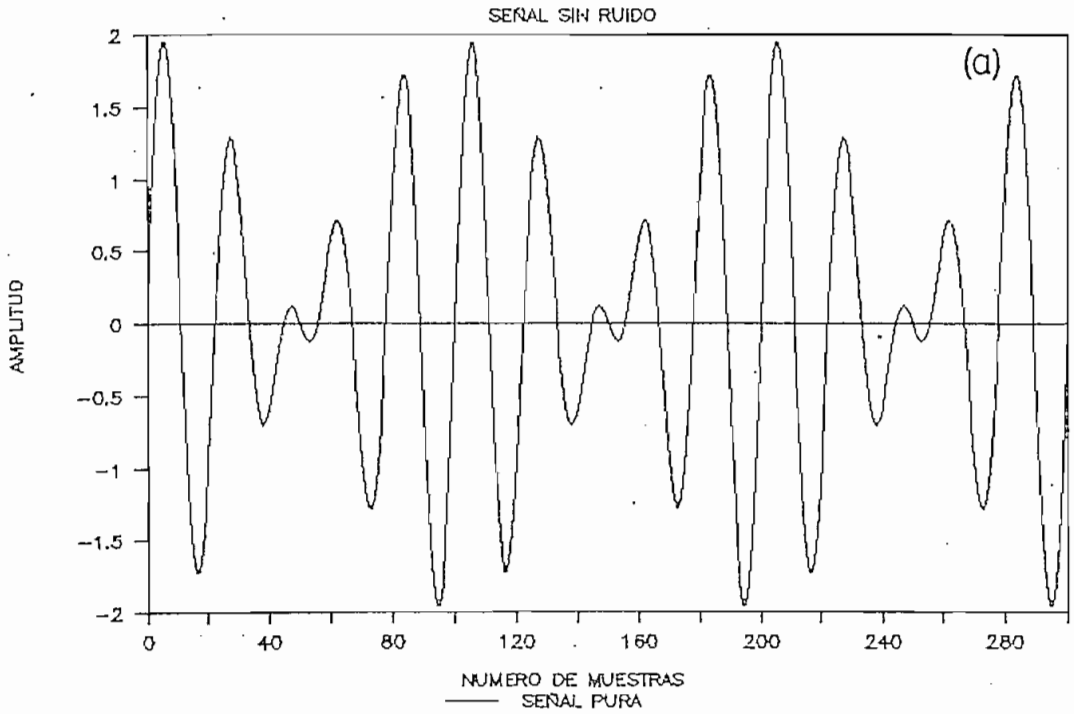
Ejemplo # 17 Procesamiento de una señal cuya frecuencia varía. (a) Señal a procesarse. (b) Comparación entrada/salida.

Ejemplo # 18

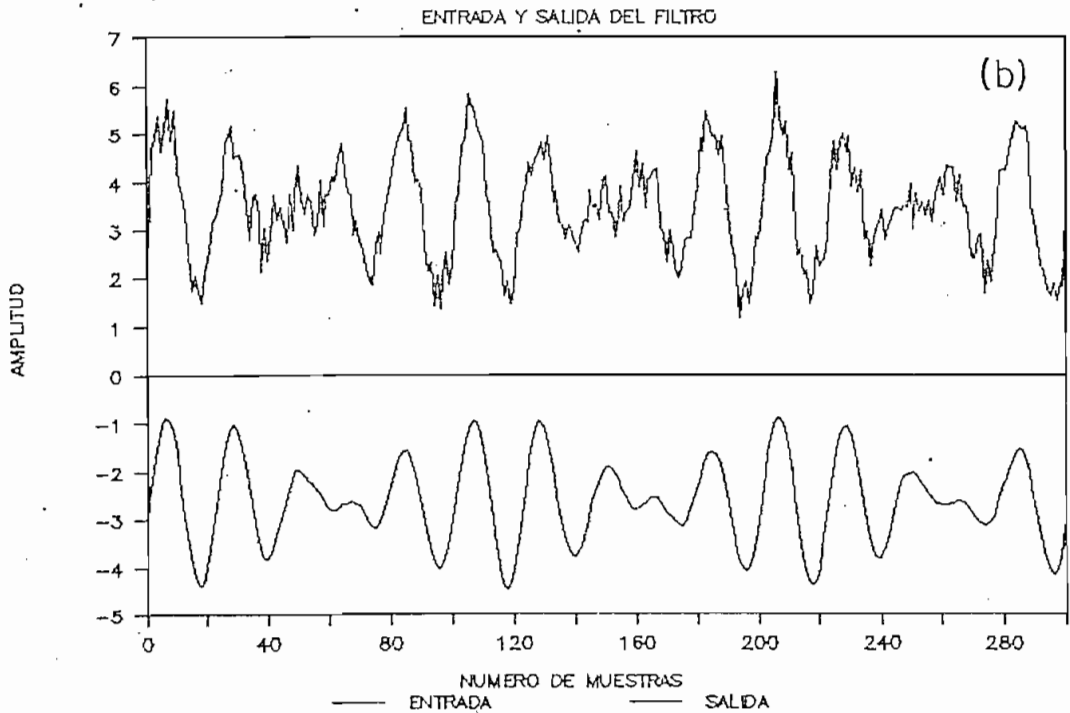
Finalmente para concluir con este grupo de ejemplos respecto al filtro adaptivo LMS, se procesa una señal periódica constituida por la suma de 2 sinusoides; se seleccionó una senoide con $f = 400$ Hz., amplitud unitaria, y otra senoide con $f = 500$ Hz. y amplitud unitaria; la señal se muestreó a 10 KHz. durante 0.1 seg.. El ruido contaminador fue ruido blanco (varianza 0.5) filtrado con un Butterworth pasaaltos ($F_c = 700$ Hz., $N = 10$). El filtro adaptivo seleccionado fue de orden 16, con una longitud de retardo igual a 8, y el parámetro de convergencia con el que se trabajó fue igual a $\mu_c/50$.

Los gráficos correspondientes mostrarán la señal periódica pura (constituida por las dos sinusoides en mención), una comparación de la señal contaminada con el ruido coloreado y la señal procesada, los espectros correspondientes tanto de la señal de entrada como de la de salida del cancelador, y la curva de aprendizaje respectiva.

CANCELACION ADAPTIVA DE RUIDO



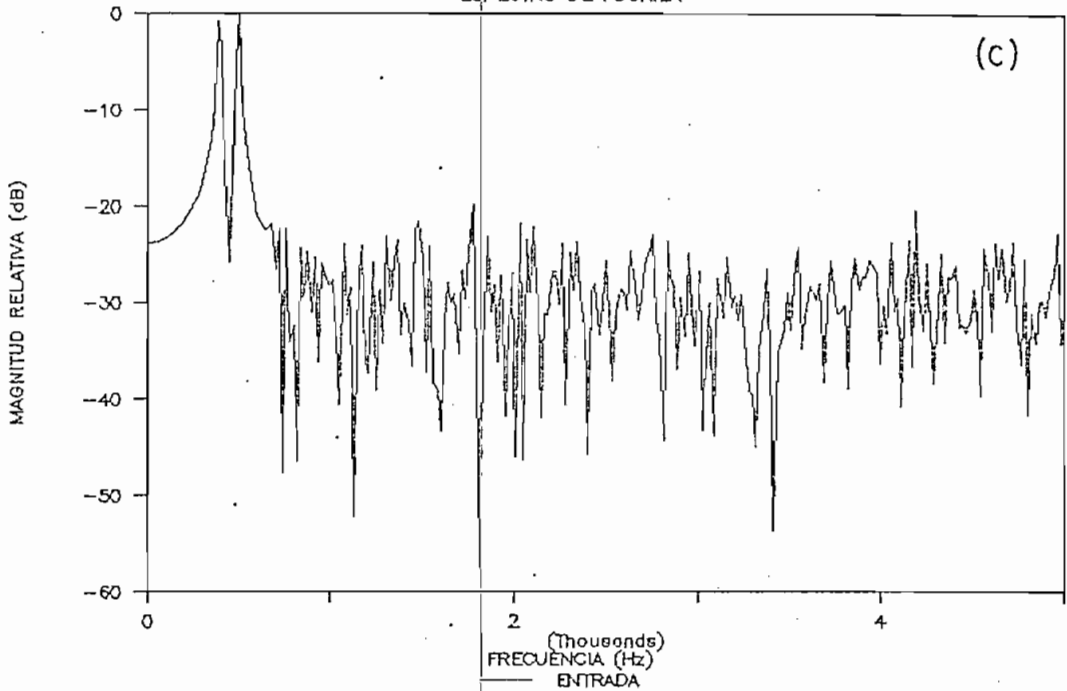
CANCELACION ADAPTIVA DE RUIDO



Ejemplo # 18 Procesamiento de una señal periódica constituida por la suma de dos sinusoides, contaminada con ruido coloreado. (a) Señal periódica libre de ruido. (b) Comparación entrada/salida.

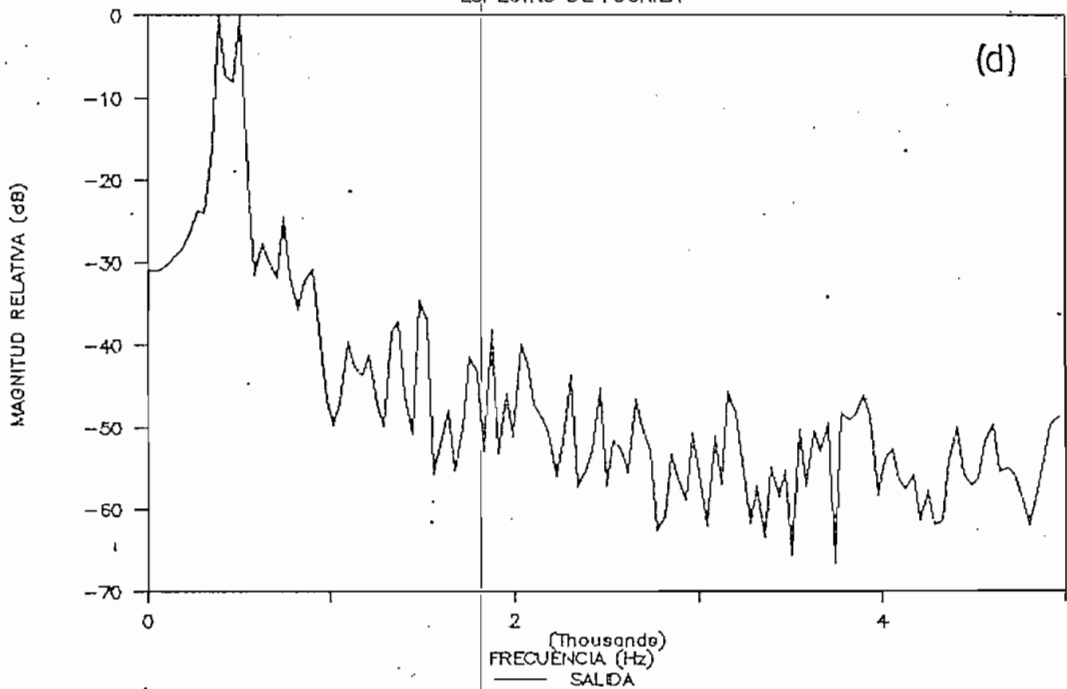
CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

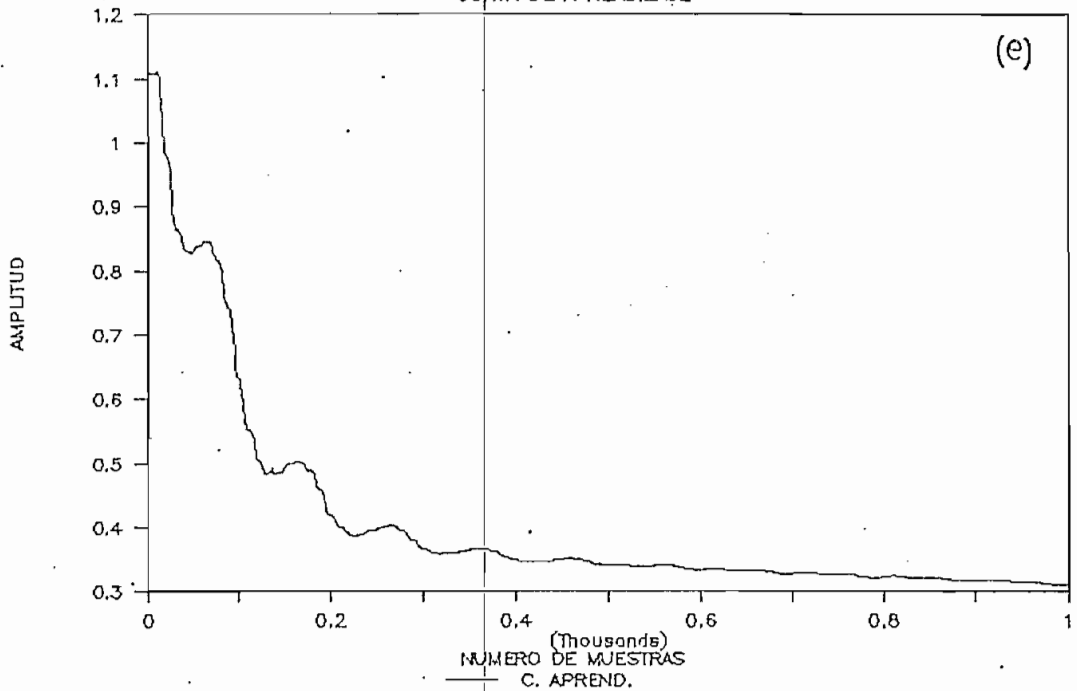


Ejemplo # 18

Procesamiento de una señal periódica constituida por la suma de 2 sinusoides. (c) Análisis espectral de la señal de entrada. (d) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE



Ejemplo # 18 (e) Curva de aprendizaje.

De todos estos ejemplos desarrollados para el algoritmo LMS se desprende:

* Cuando se opera con una señal constituida por una onda sinusoidal mas ruido blanco aditivo (ejemplo 7), se observa que en la señal filtrada existe presencia de una componente de frecuencia parásita, esto se ve mas claramente en el gráfico (c) correspondiente a este ejemplo, donde se presenta el análisis espectral de la señal de salida del cancelador. Al trabajar con ruido blanco, el cual contiene todas las componentes de frecuencia, su espectro se mezcla con el espectro de la señal (una sola frecuencia en este caso), de manera que para que el filtro adaptivo deje pasar sólo la frecuencia de la señal deberá generar un notch a dicha frecuencia, al parecer la selectividad del notch en tales circunstancias no es óptima dejando filtrar frecuencias cercanas a las de la señal. Efectivamente, como se observa en el ejemplo # 8, donde se tiene como interferencia ruido coloreado de frecuencias por encima de la de

señal, de manera que los espectros no se cruzan, al producirse el filtrado, la señal a la salida no contiene ninguna componente adicional.

* El ejemplo # 9 constituye el caso donde una señal periódica se encuentra contaminada con ruido coloreado cuya potencia es superior a la de la señal, en tal caso el cancelador trata de eliminar el ruido, y es notorio desde los análisis espectrales respectivos el nivel de rechazo al ruido que se consigue. De manera que el sistema adaptivo de cancelación es deseable en tales casos.

* Con los ejemplos 10, 11, 12 y 13, se pretende realizar un análisis de la influencia del valor de μ en la velocidad de convergencia y en el error en estado estable; se realizaron pruebas con valores de μ pequeños (ejemplo 10), donde la velocidad de convergencia es lenta, pero la señal filtrada prácticamente carece de ruido, cuando el valor de μ se aumenta cerca del límite superior la adaptación se produce mucho más rápido, pero la señal filtrada contiene así mismo más ruido. Se comprobó que efectivamente un valor de μ por encima del valor crítico produce divergencia en el algoritmo (ejemplo # 13).

* En los ejemplos 14, 15 y 16, se analiza la influencia del orden del filtro, estos resultados demuestran que efectivamente a medida que el orden del filtro se incrementa, mejora su respuesta, sin embargo superado un determinado orden del filtro tal mejora es prácticamente imperceptible.

* El ejemplo 17 analiza el reajuste del sistema adaptivo frente a cambios en la frecuencia de la señal periódica, de los resultados obtenidos se concluye que las características de frecuencia son transparentes al filtro, de manera que puede operar con un desconocimiento total de la

frecuencia de la señal periódica, adaptándose a cambios que se pueden producir en su valor de manera que luego de la fase de reajuste entrega nuevamente la señal libre de ruido.

* Finalmente el ejemplo 18 constituye el análisis de una señal constituida por la suma de dos sinusoides, con ruido blanco coloreado aditivo; el sistema de cancelación de ruido devuelve la señal a la salida del mismo libre de ruido, sin embargo se produce una alteración en la fase, esto se debe a que el filtro adaptivo implementado en el presente trabajo de tesis no tiene características de fase lineal, en consecuencia la señal a su salida presenta distorsiones de fase como se muestra en los gráficos (b) y (c) del ejemplo 18.

4.1.3 RESULTADOS SOBRE EL FILTRO ADAPTIVO RLS

A fin de poder analizar el comportamiento del filtro adaptivo RLS se desarrollaron varios ejemplos, la señal a procesarse en cada caso era la misma de los ejemplos realizados para el filtro adaptivo LMS, esto con la finalidad de poder establecer comparaciones entre los dos métodos.

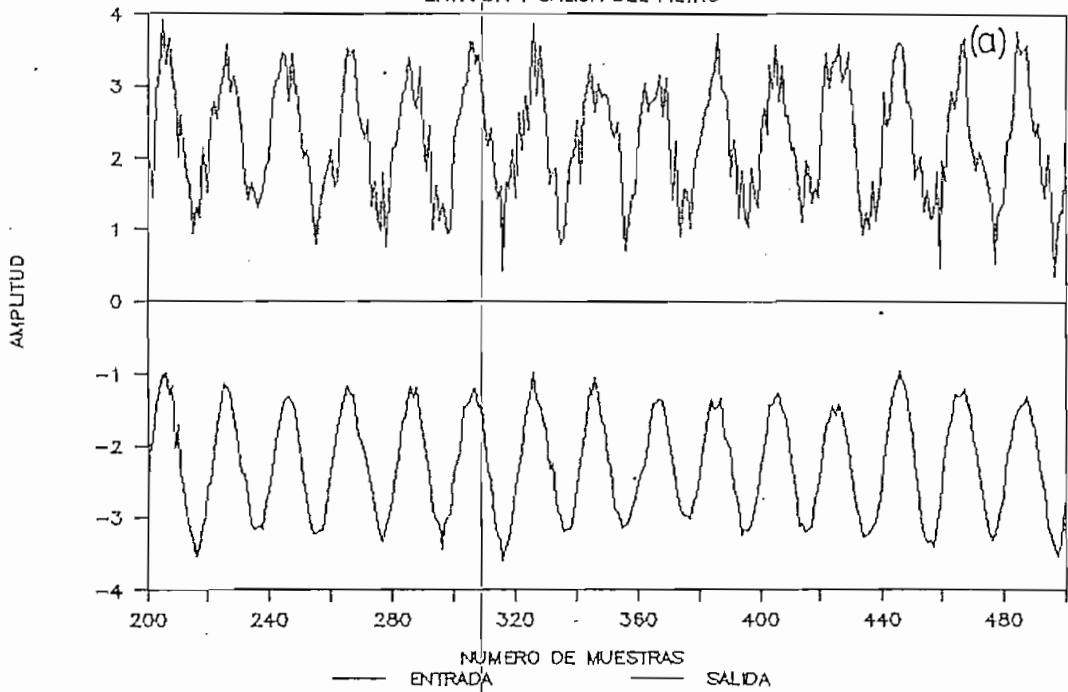
Con los ejemplos 19, 20 y 21 se intenta analizar el comportamiento del filtro adaptivo RLS cuando la señal a procesarse esta contaminada con ruido blanco, ruido coloreado y ruido coloreado de alto nivel, respectivamente. Las señales que se procesarán son las mismas que en los ejemplos 7, 8 y 9, pero muestreadas únicamente durante 0.05 seg.

Ejemplo # 19

El filtro adaptivo seleccionado fue de orden 16, $T = 8$, $\sigma = 10^2$, y $\rho = 0.998$.

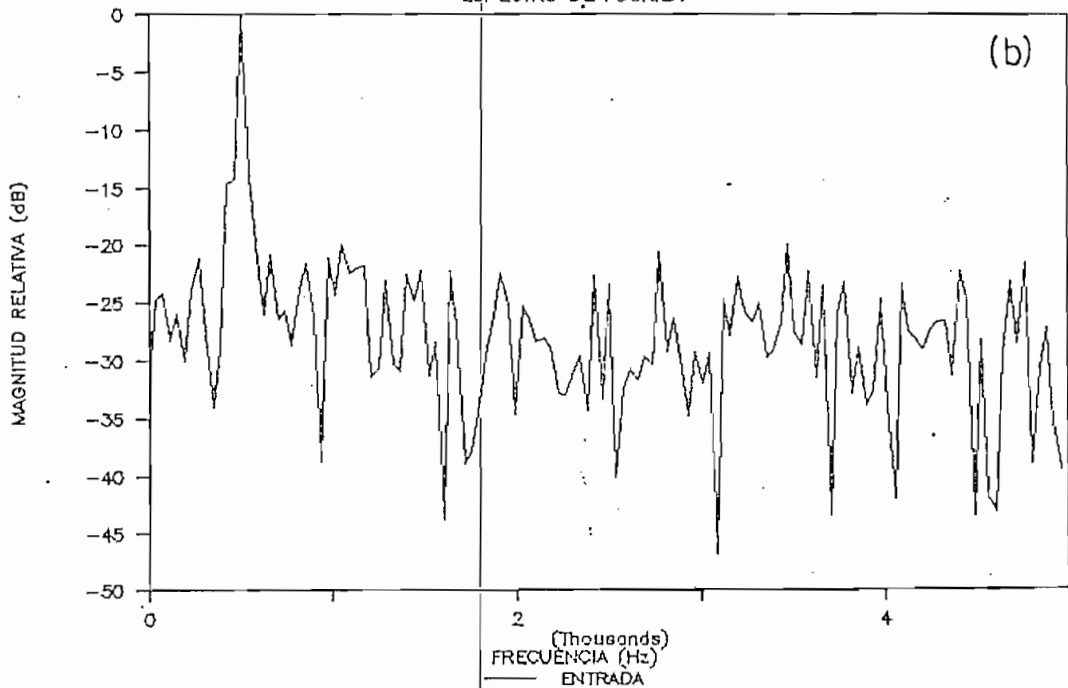
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

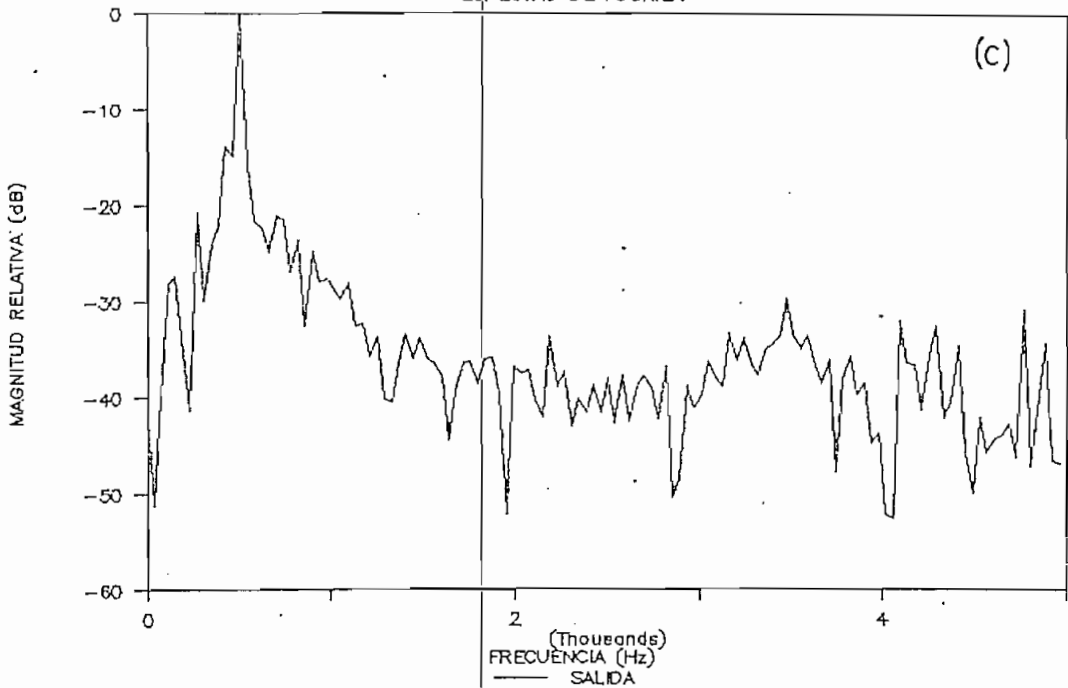


Ejemplo # 19

Procesamiento de una señal periódica contaminada con ruido blanco. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de entrada.

CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



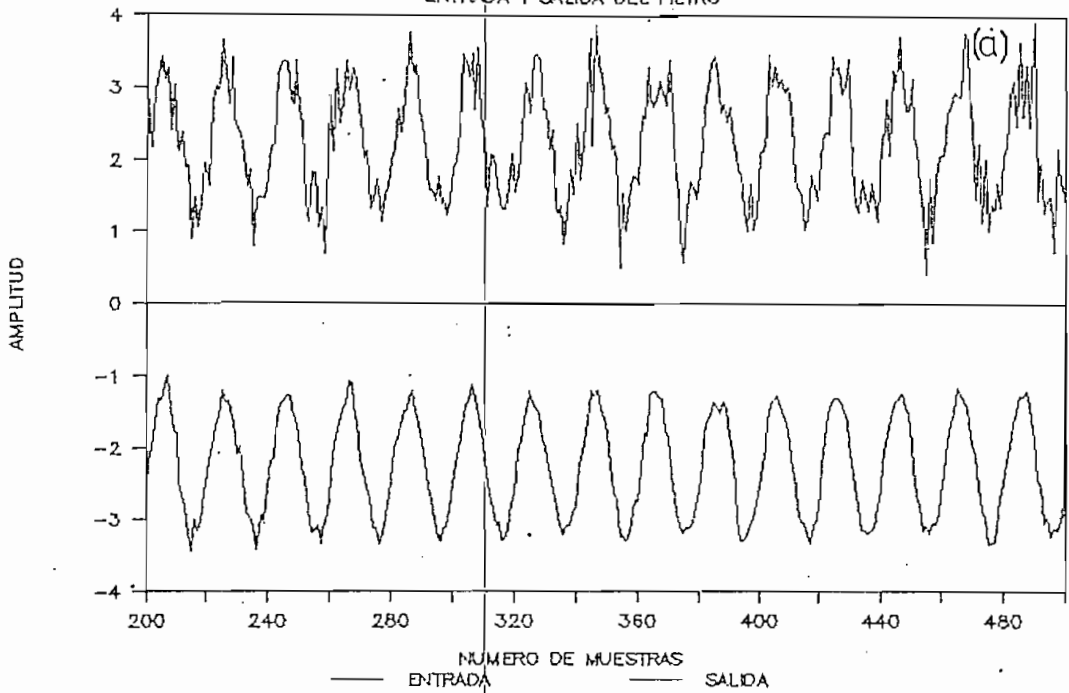
Ejemplo # 19 (c) Análisis espectral de la señal de salida.

Ejemplo # 20

Se trata de observar con este ejemplo el comportamiento del filtro adaptivo RLS cuando la señal a procesarse es una onda periódica idéntica a la del ejemplo # 19, contaminada con ruido coloreado obtenido a partir de una señal de ruido blanco con varianza 0.5, filtrada con un pasabajos de orden 10 y $F_c = 700$ Hz. Este es un ejemplo idéntico al # 8, desarrollado para el caso del filtro adaptivo LMS. El filtro adaptivo RLS para este caso es el mismo que se usó en el ejemplo # 19.

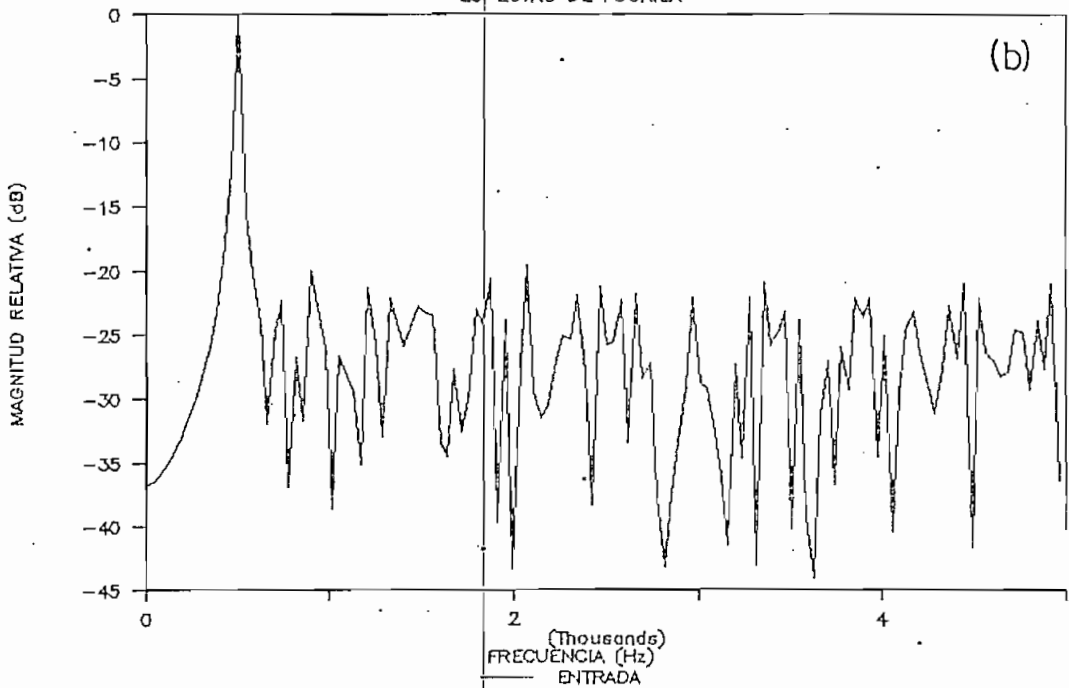
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

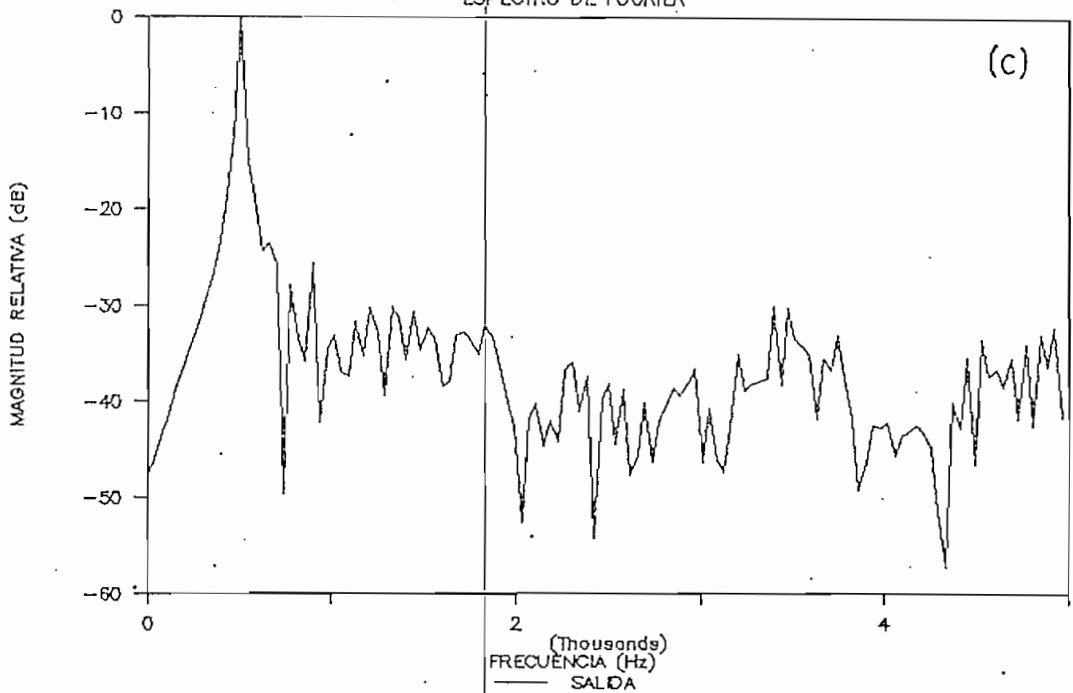


Ejemplo # 20

Procesamiento de una señal periódica contaminada con ruido coloreado. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de entrada.

CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



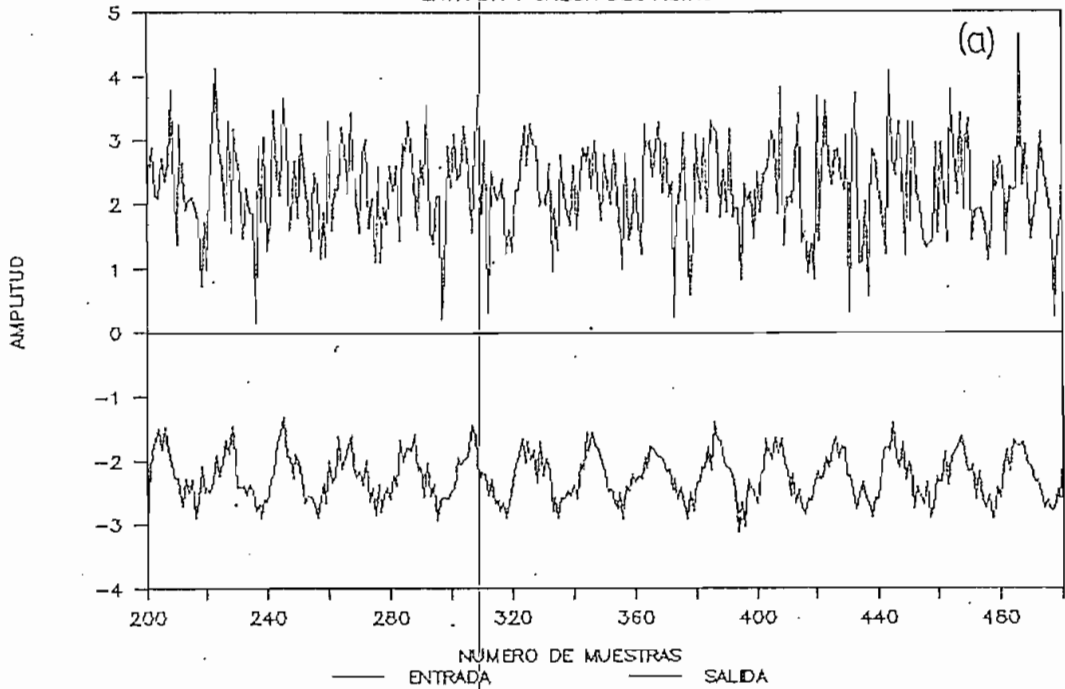
Ejemplo # 20 (c) Análisis espectral de la señal de salida.

Ejemplo # 21

Con este ejemplo se pretende estudiar el comportamiento del cancelador de ruido RLS cuando procesa una señal de bajo nivel en ruido (potencia del ruido mayor que la potencia de la señal), en forma similar a como se realizó con el filtro LMS en el ejemplo # 9. El filtro adaptivo seleccionado para este caso fue el mismo que se usó en el ejemplo # 20.

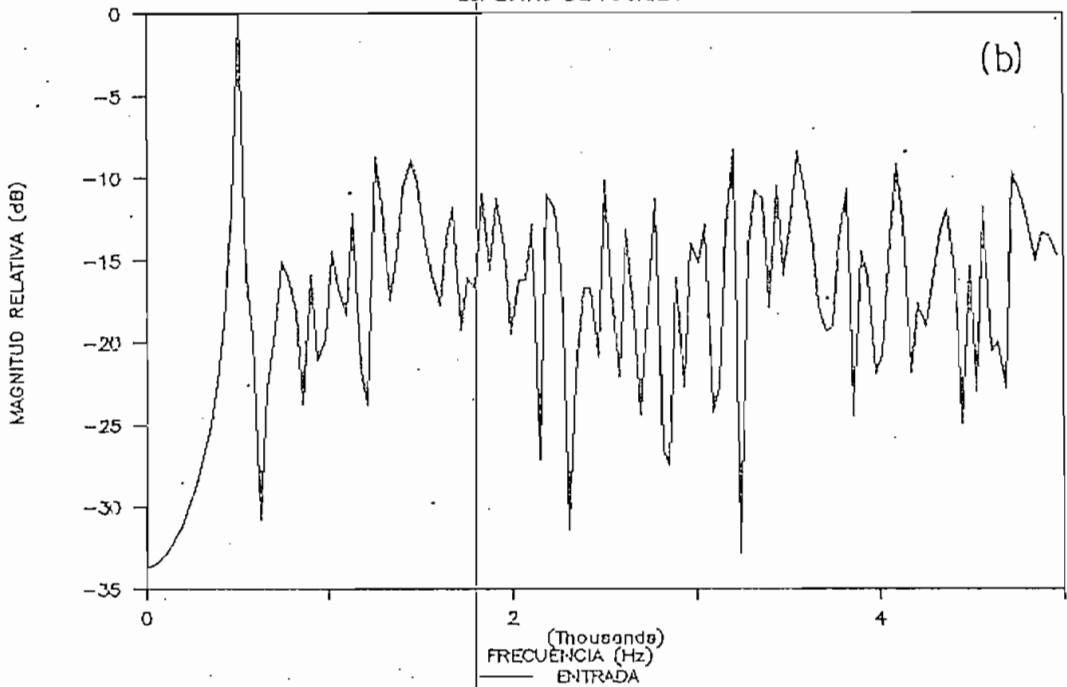
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

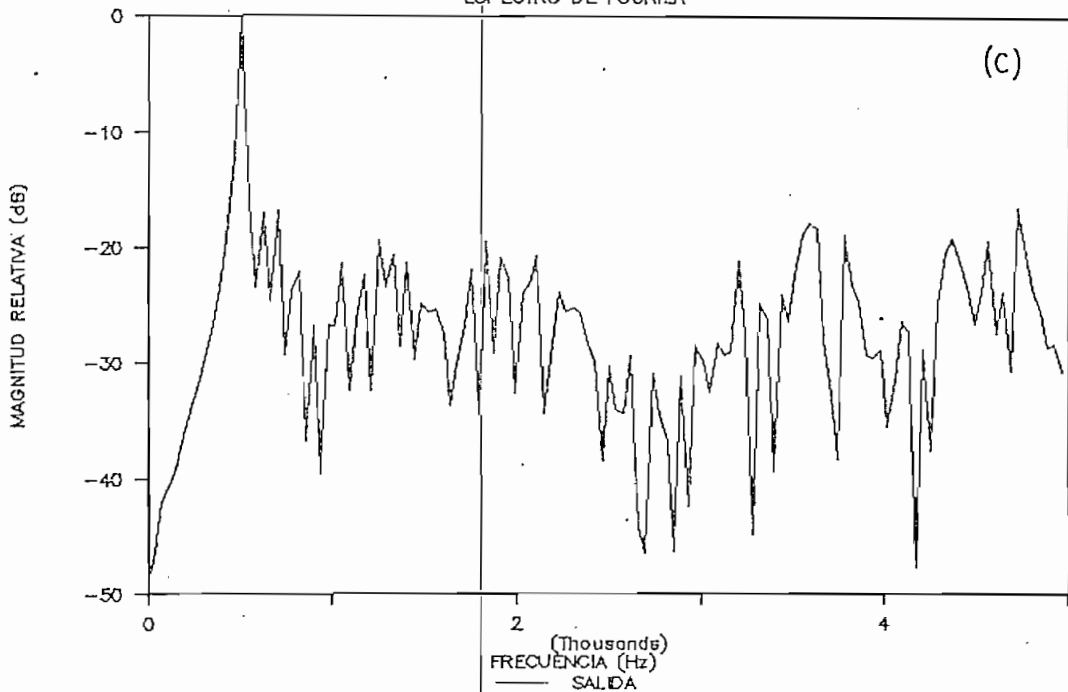
ESPECTRO DE FOURIER



Ejemplo # 21 Procesamiento de una señal periódica de bajo nivel en ruido coloreado. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de entrada.

CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



Ejemplo # 21 (c) Análisis espectral de la señal de salida.

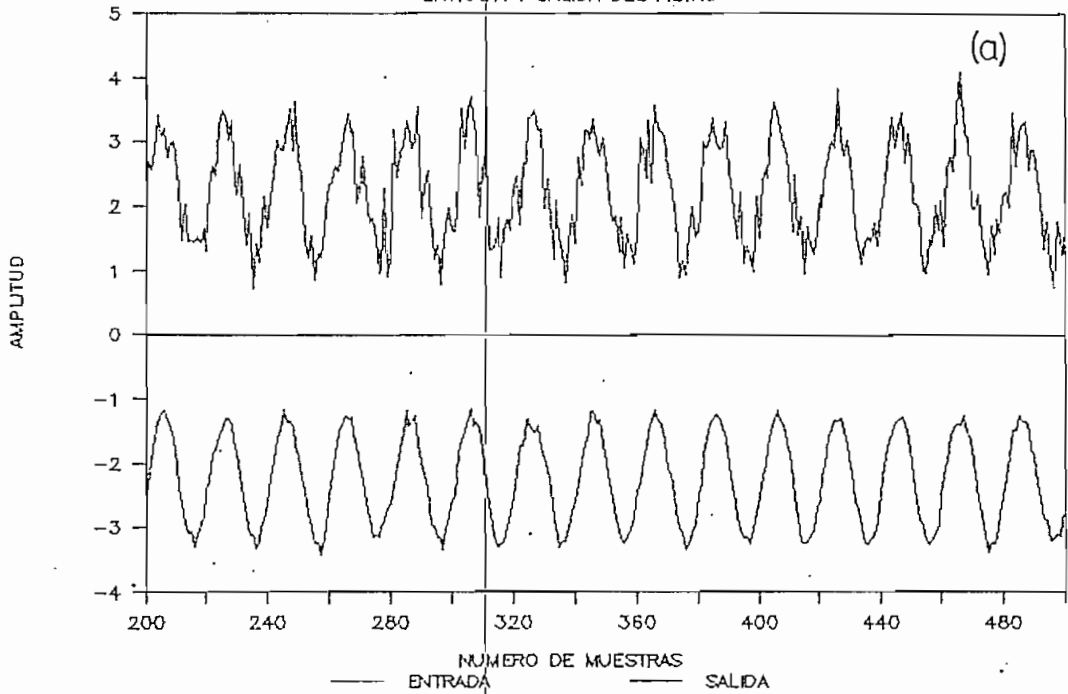
Ejemplos # 22, 23 y 24

Se trata de analizar en estos ejemplos el efecto del factor de olvido, ρ , tanto en el comportamiento (calidad de la señal procesada) como en la convergencia del algoritmo. La señal que se procesa es la misma del ejemplo # 20, igualmente los parámetros del filtro adaptivo son los mismos; difiriendo cada uno de los ejemplos en el factor de olvido utilizado, así para el ejemplo # 22 $\rho = 1$, para el ejemplo # 23 $\rho = 0.995$ y, para el ejemplo # 24 $\rho = 0.95$.

Para cada uno de los ejemplos se obtuvo tanto la curva de aprendizaje, como el análisis espectral correspondiente.

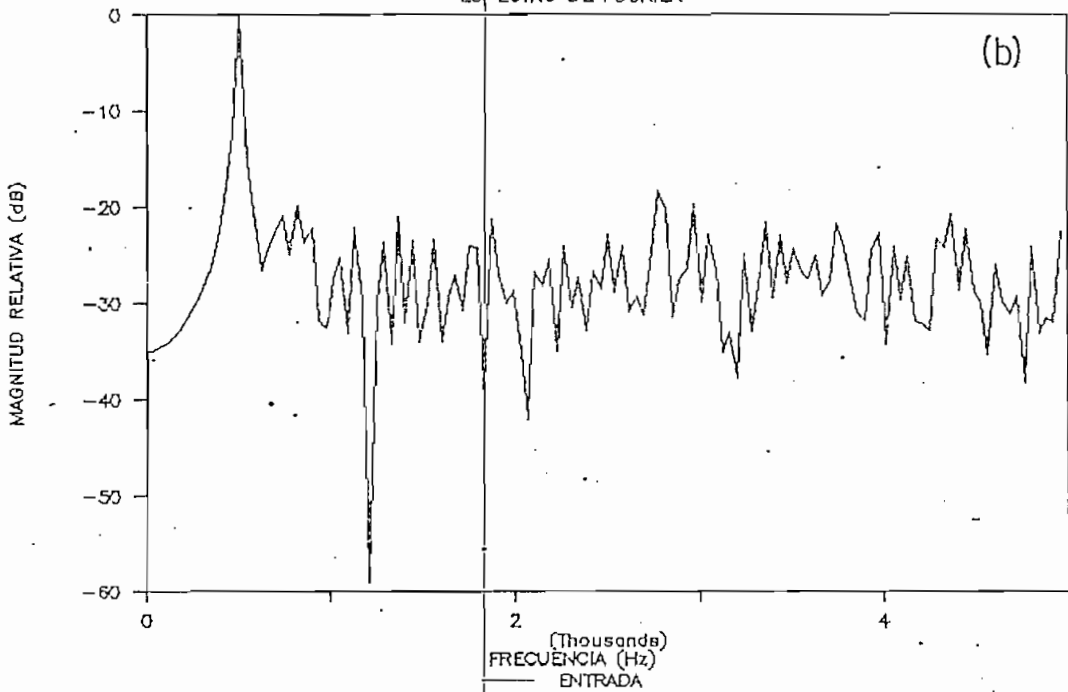
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

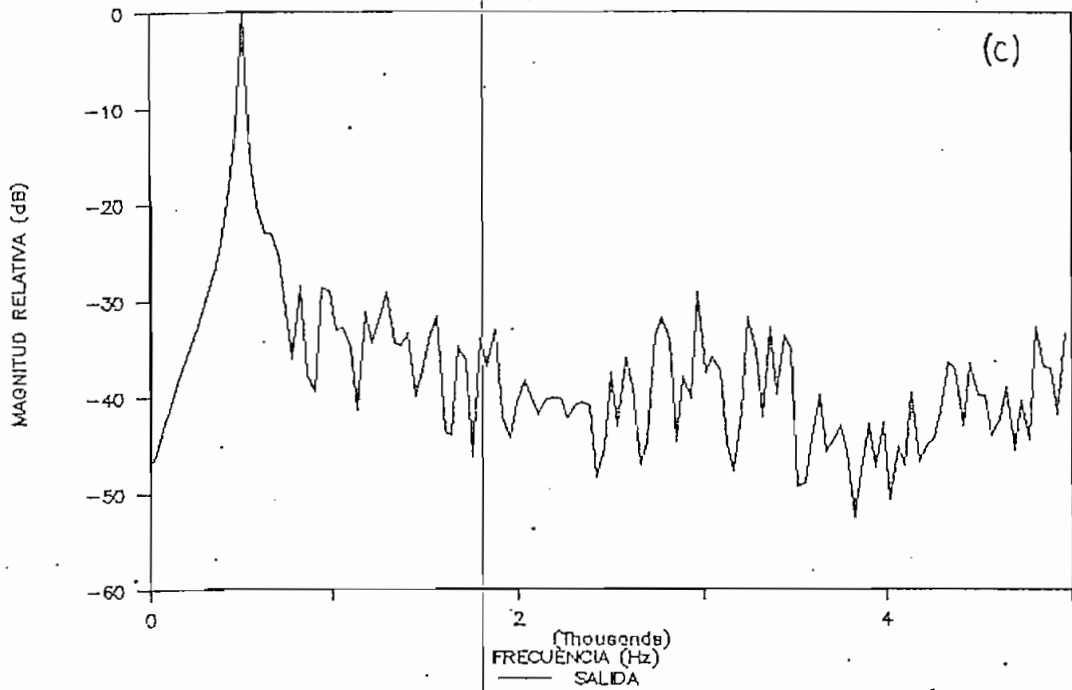


Ejemplo # 22

Procesamiento de una señal periódica contaminada con ruido coloreado. $\rho = 1$. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de entrada.

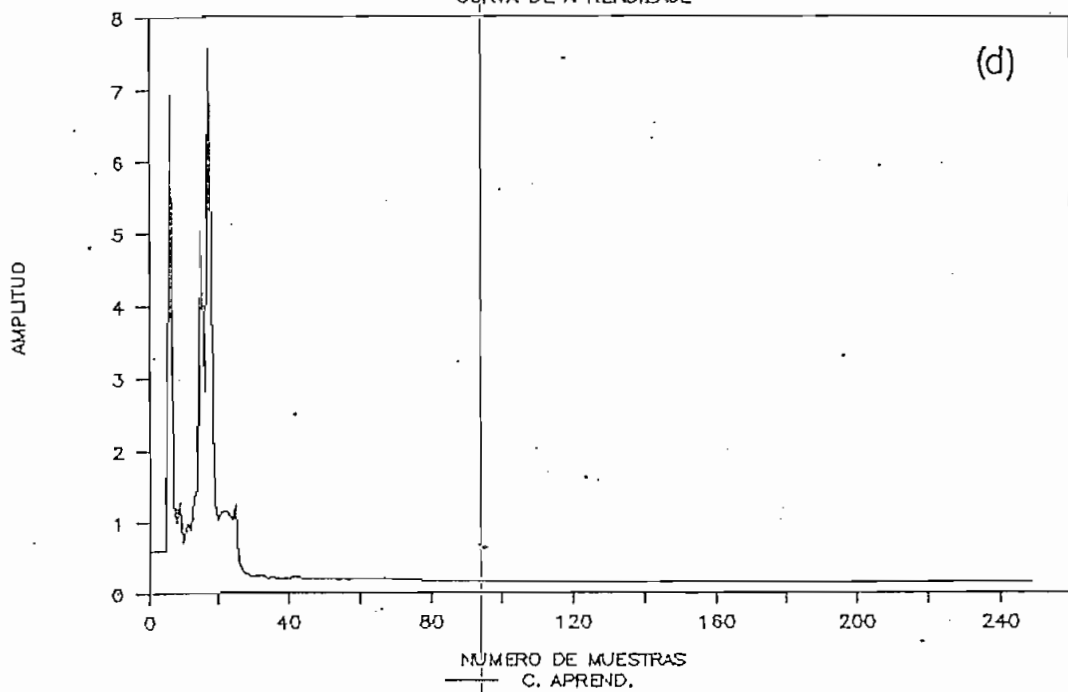
CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



CANCELACION ADAPTIVA DE RUIDO

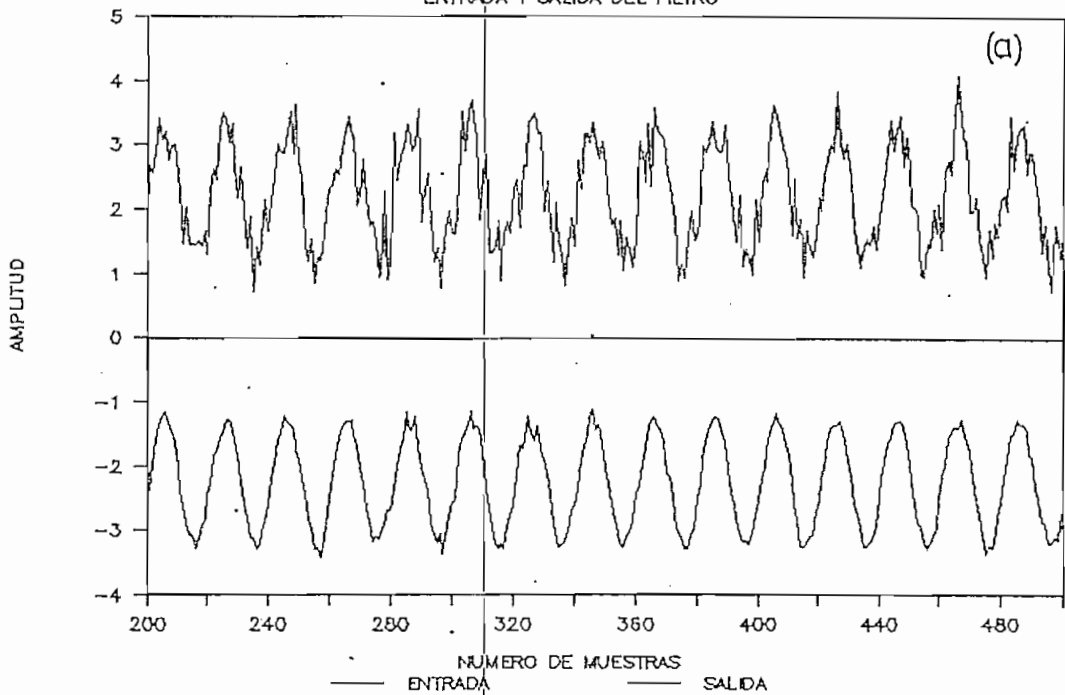
CURVA DE APRENDIZAJE



Ejemplo # 22 (c) Análisis espectral de la señal de salida. (d) Curva de aprendizaje.

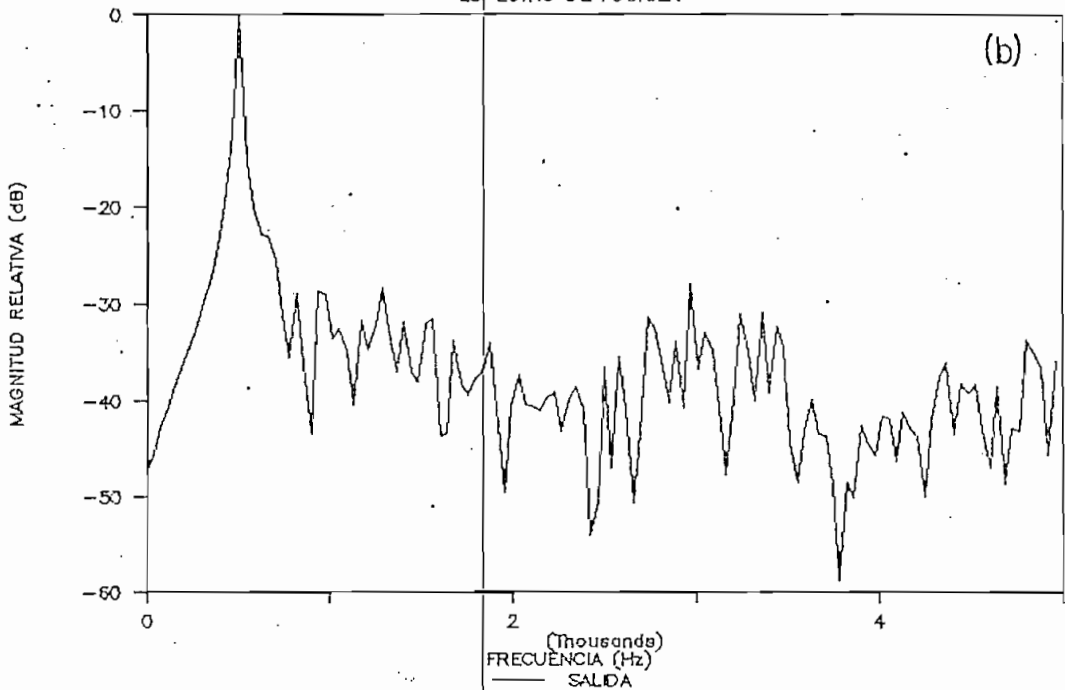
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

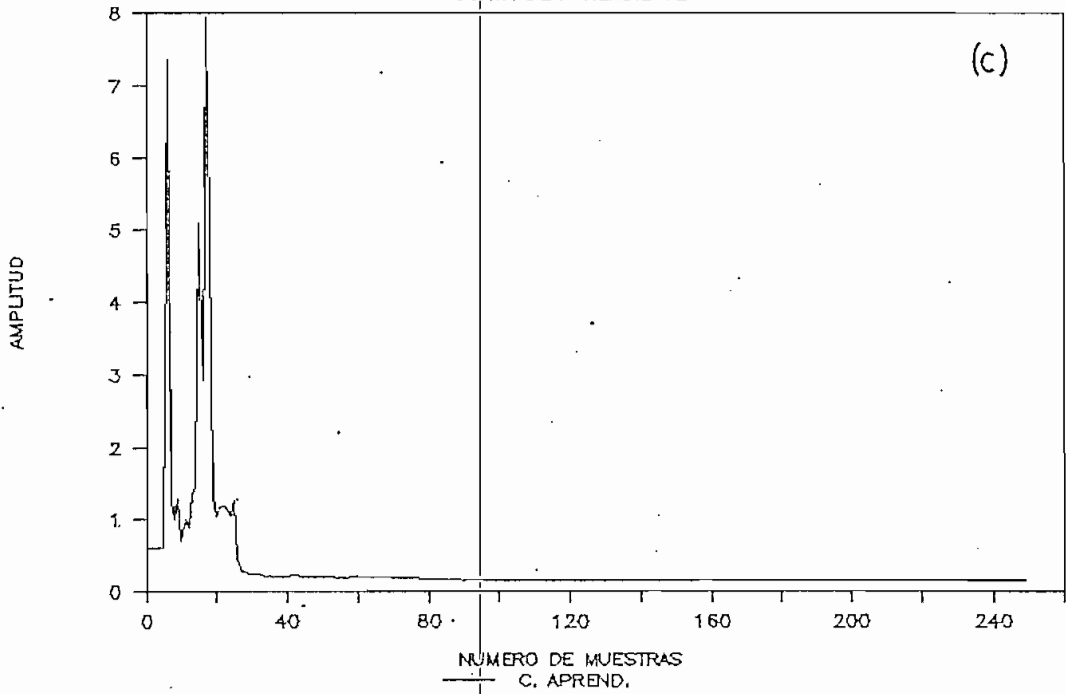


Ejemplo # 23

Procesamiento de una señal periódica contaminada con ruido coloreado. $\rho = 0.995$.
(a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

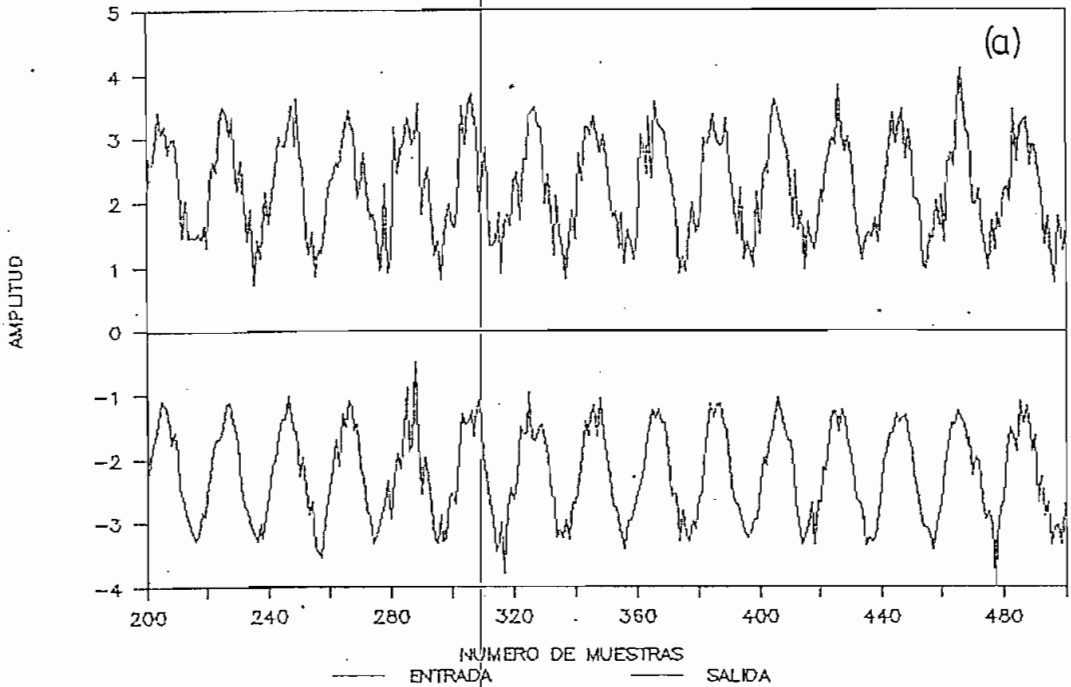
CURVA DE APRENDIZAJE



Ejemplo # 23 (d) Curva de aprendizaje.

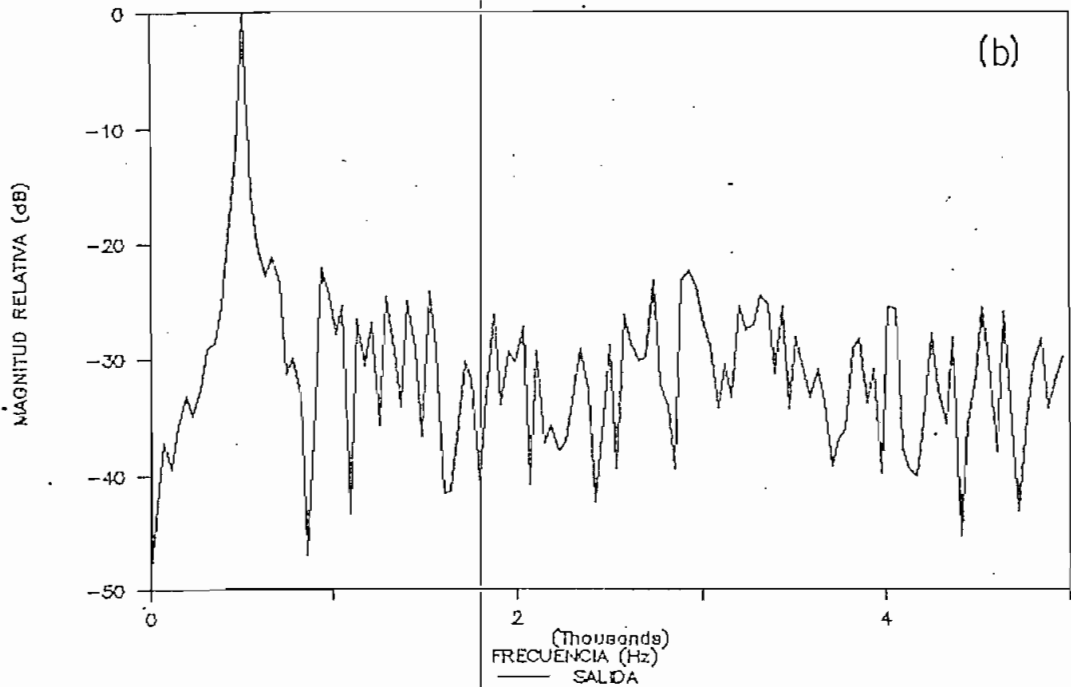
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

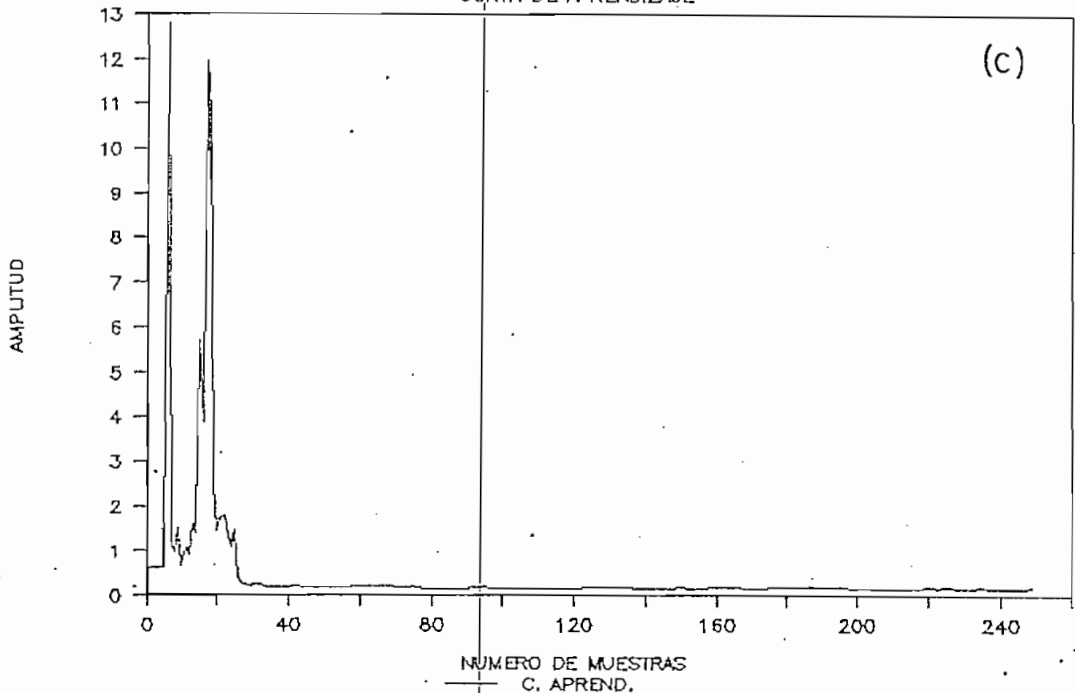
ESPECTRO DE FOURIER



Ejemplo # 24 Procesamiento de una señal periódica contaminada con ruido coloreado. $\rho = 0.95$. (a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE



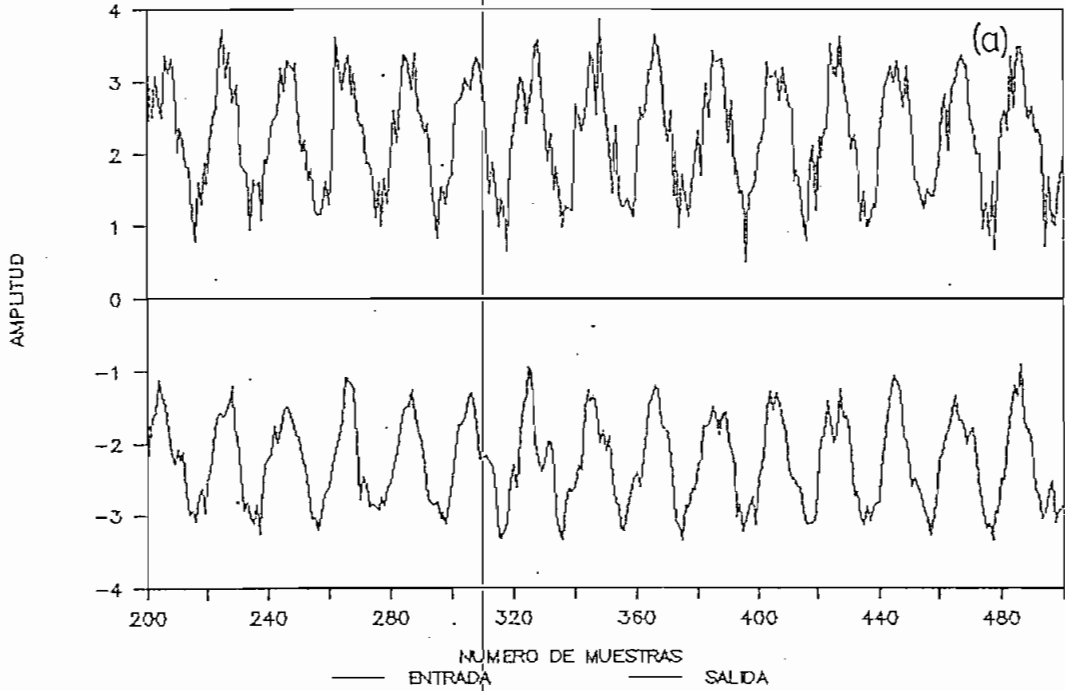
Ejemplo # 24 (c) Curva de aprendizaje.

Ejemplos # 25, 26 y 27

Al igual que para el algoritmo LMS se trata de analizar con estos ejemplos la influencia del orden del filtro en la calidad de la señal filtrada. La señal a procesarse es la misma de los ejemplos anteriores (22, 23 y 24), difiriendo en cada caso el orden del filtro usado y la correspondiente longitud de retardos, así: para el ejemplo # 25 el orden del filtro fue 6 y el número de retardos 3, para el ejemplo # 26 $N=10$, $T=5$ y, para el ejemplo # 27 $N=16$, $T=8$. El factor de olvido fue el mismo en todos los casos, $\rho=0.998$, y la constante de inicialización, $\sigma=10^2$, en todos los ejemplos.

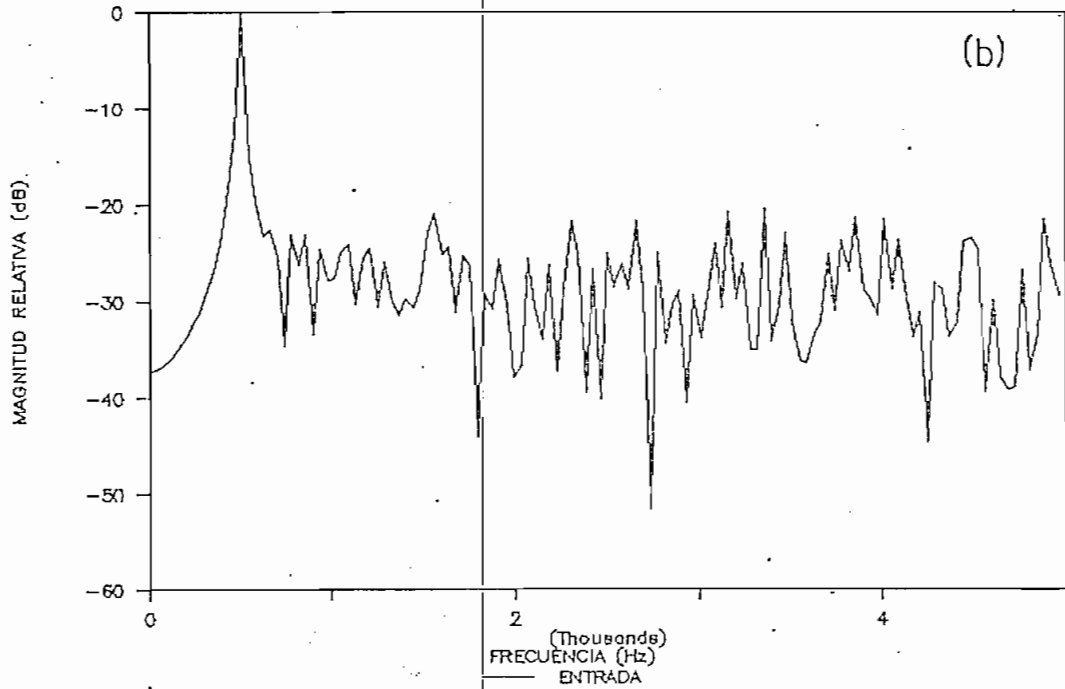
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

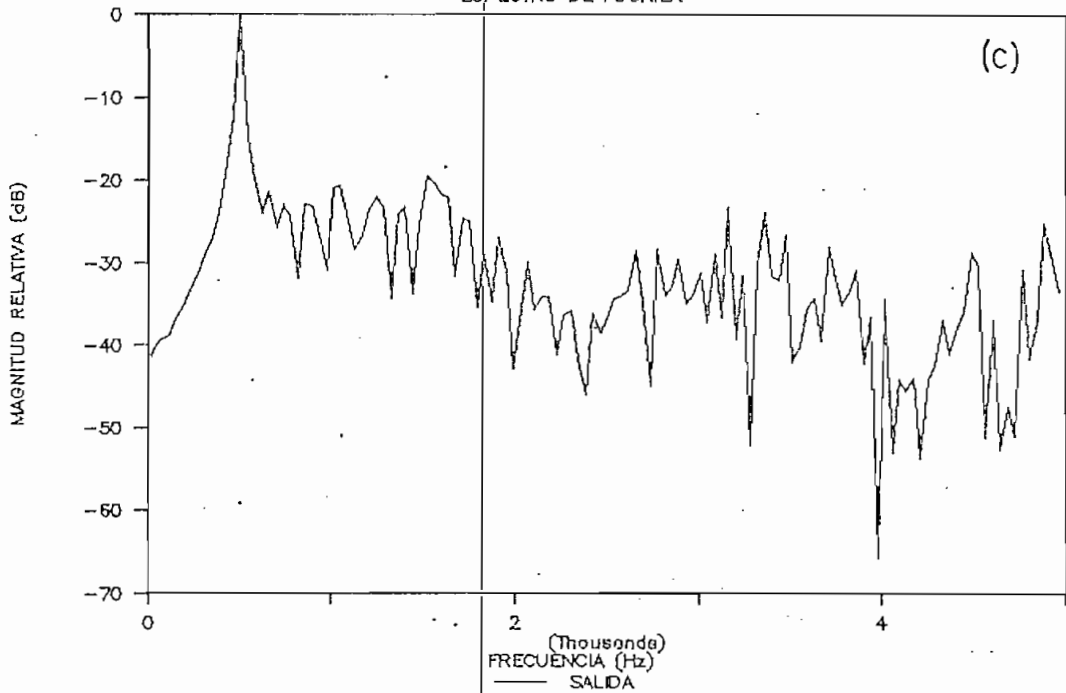


Ejemplo # 25

Procesamiento de una señal periódica contaminada con ruido coloreado. $N = 6$, $T = 3$.
(a) Comparación entrada/salida. (b) Análisis espectral de la señal de entrada.

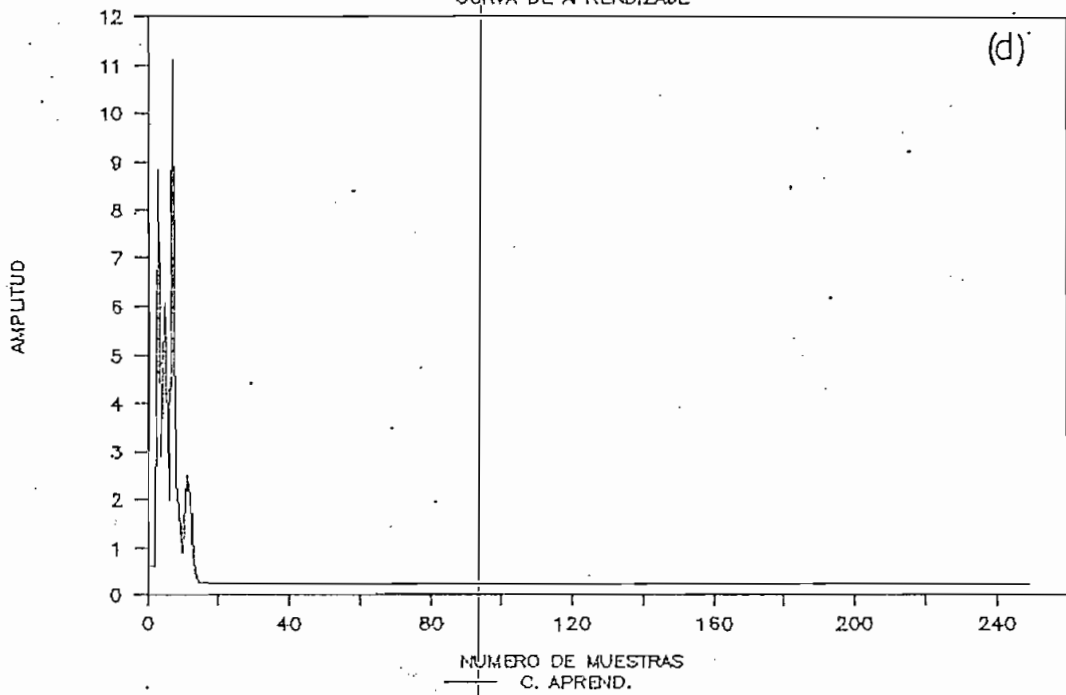
CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



CANCELACION ADAPTIVA DE RUIDO

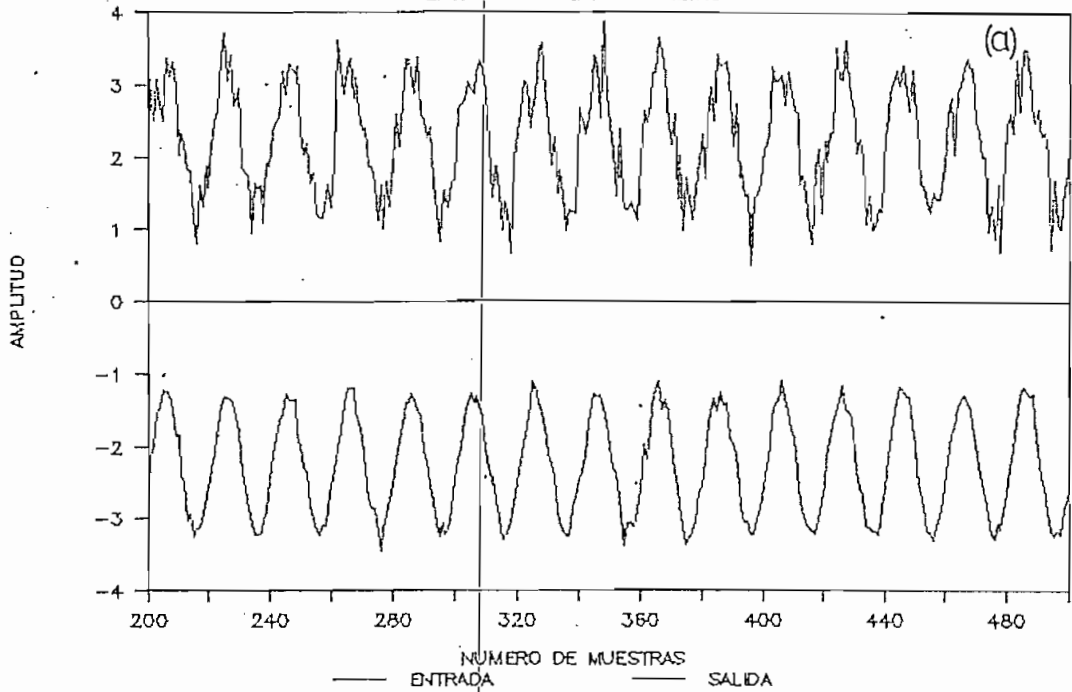
CURVA DE APRENDIZAJE



Ejemplo # 25 (c) Análisis espectral de la señal de salida. (d) Curva de aprendizaje.

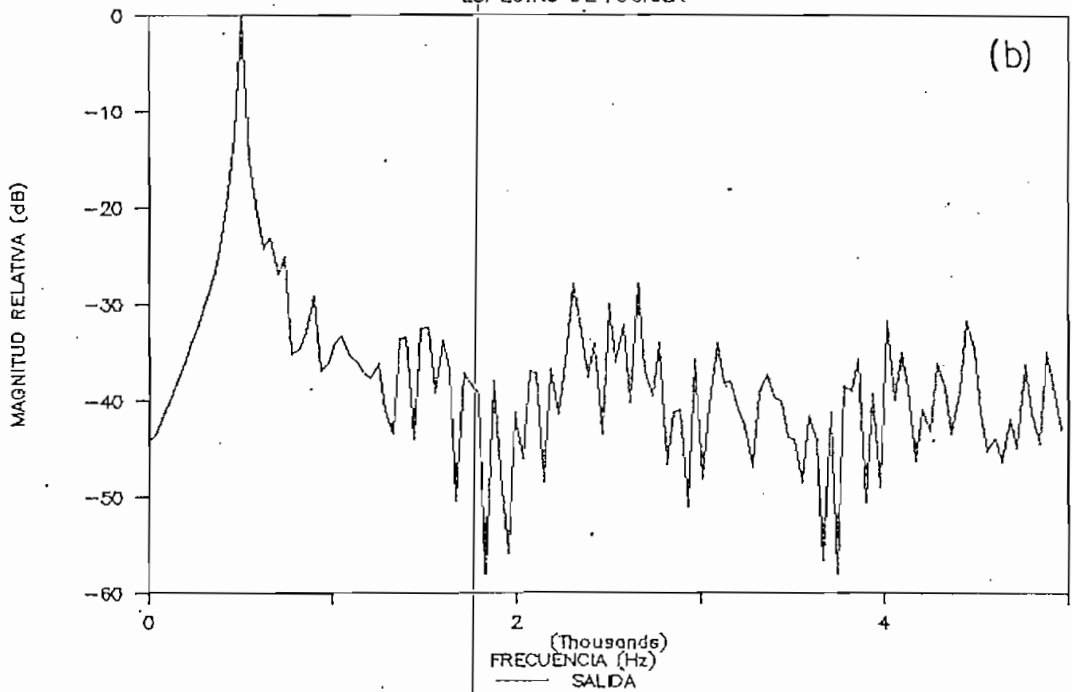
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

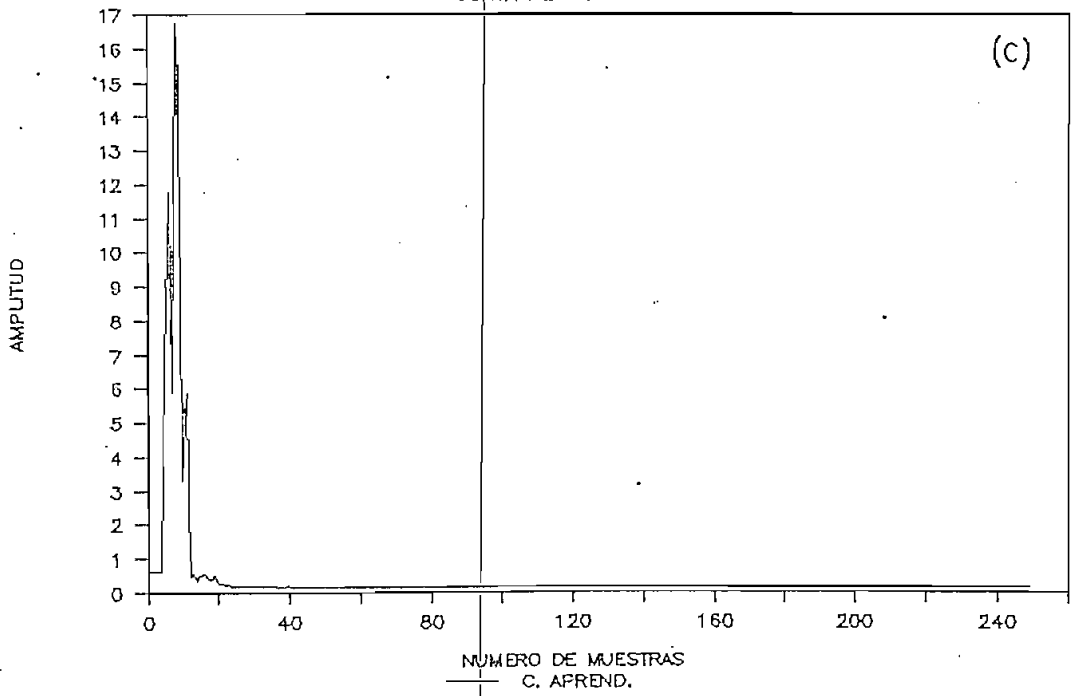


Ejemplo # 26

Procesamiento de una señal periódica contaminada con ruido coloreado. $N = 10$, $T = 5$.
(a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

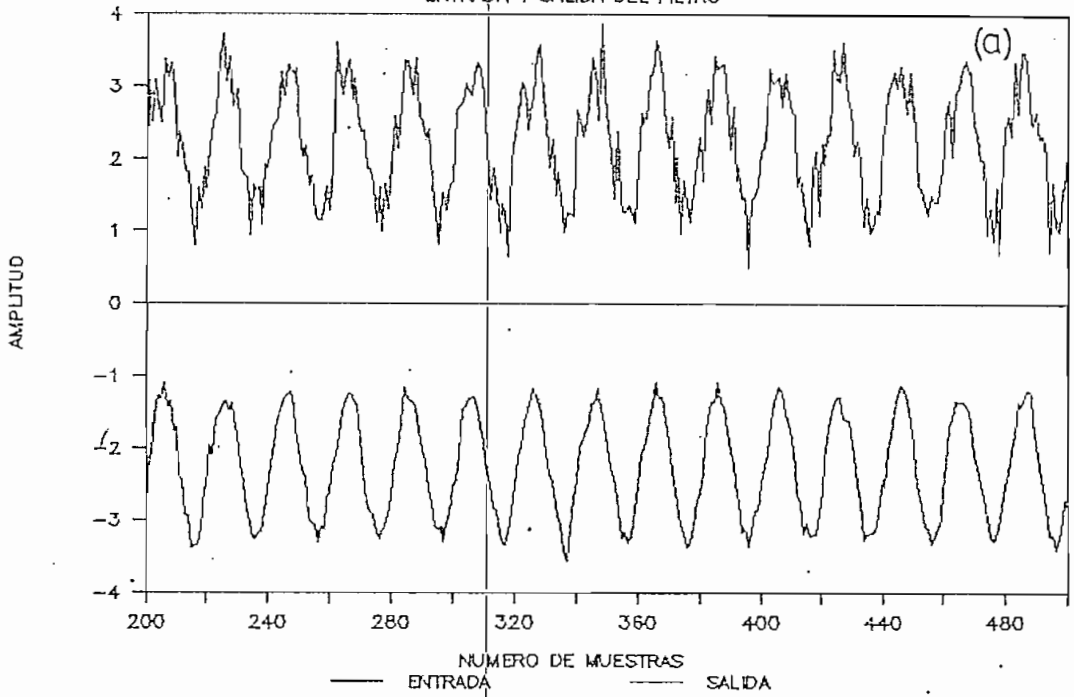
CURVA DE APRENDIZAJE



Ejemplo # 26 (c) Curva de aprendizaje.

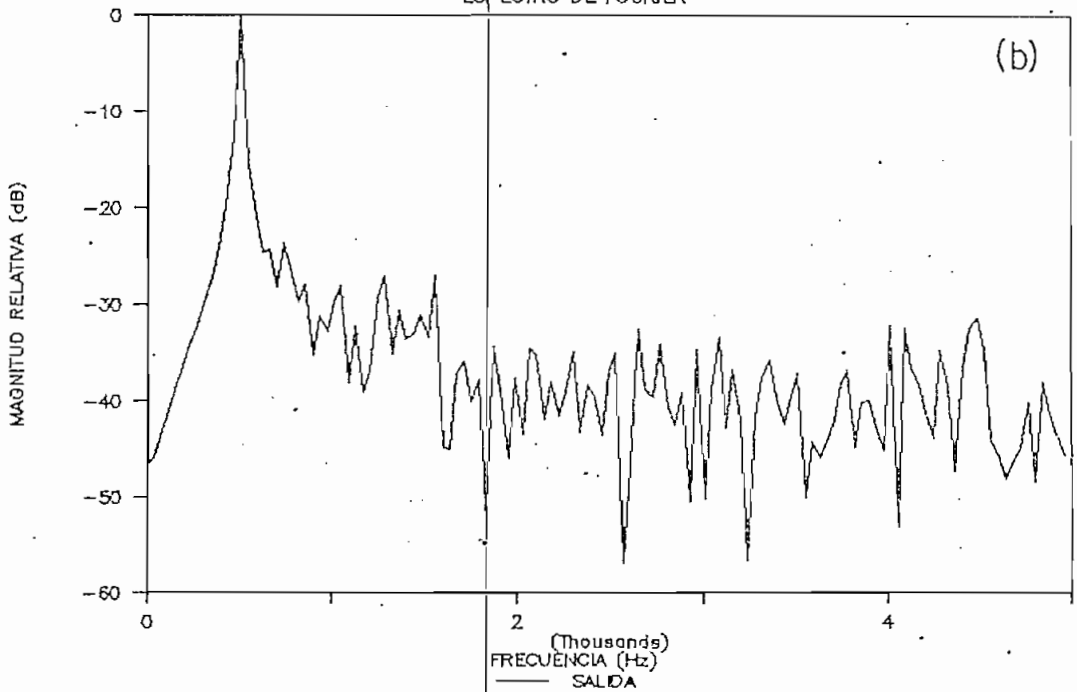
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER

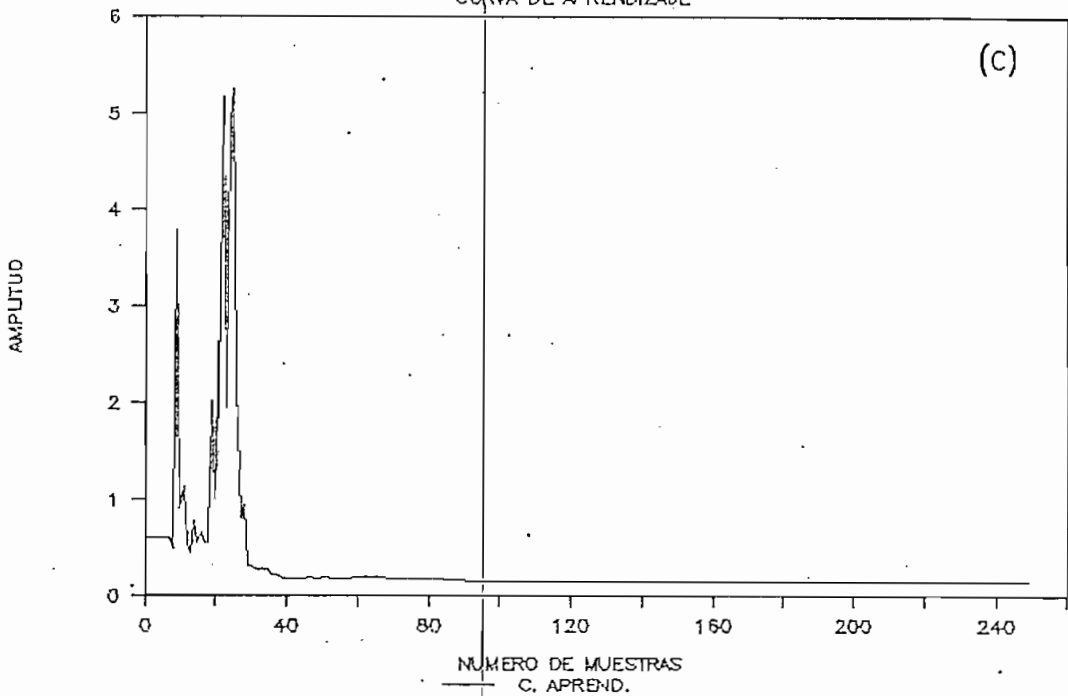


Ejemplo # 27

Procesamiento de una señal periódica contaminada con ruido coloreado. $N = 16$, $T = 8$.
(a) Comparación entrada/salida. (b) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE

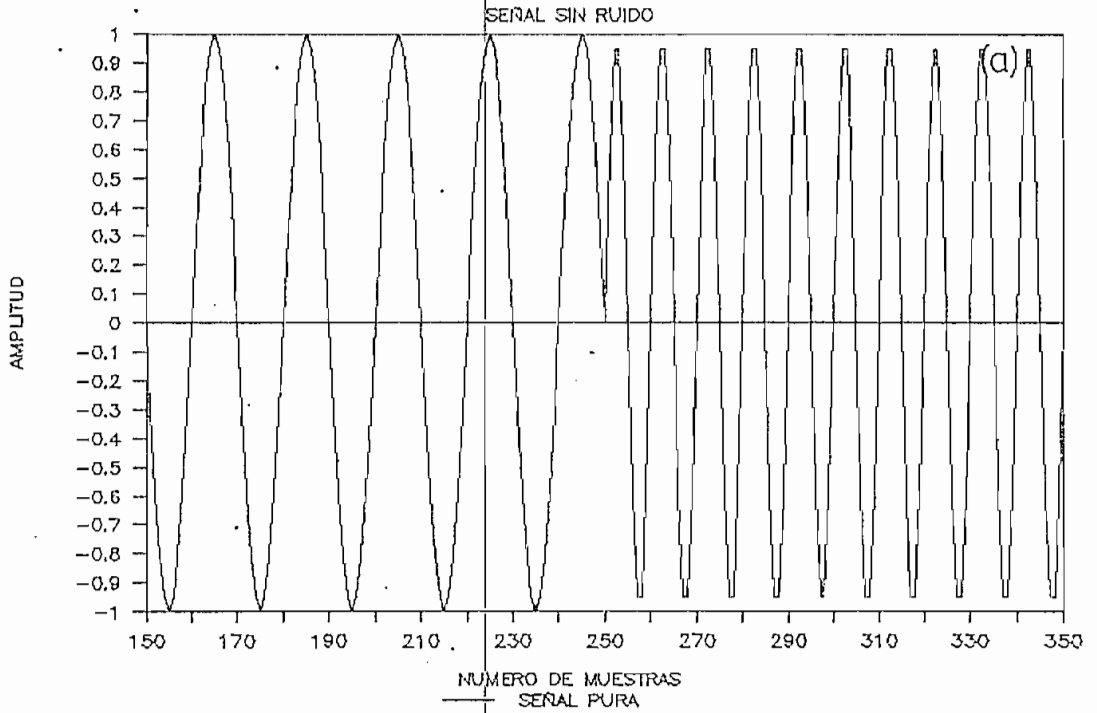


Ejemplo # 27 (c) Curva de aprendizaje.

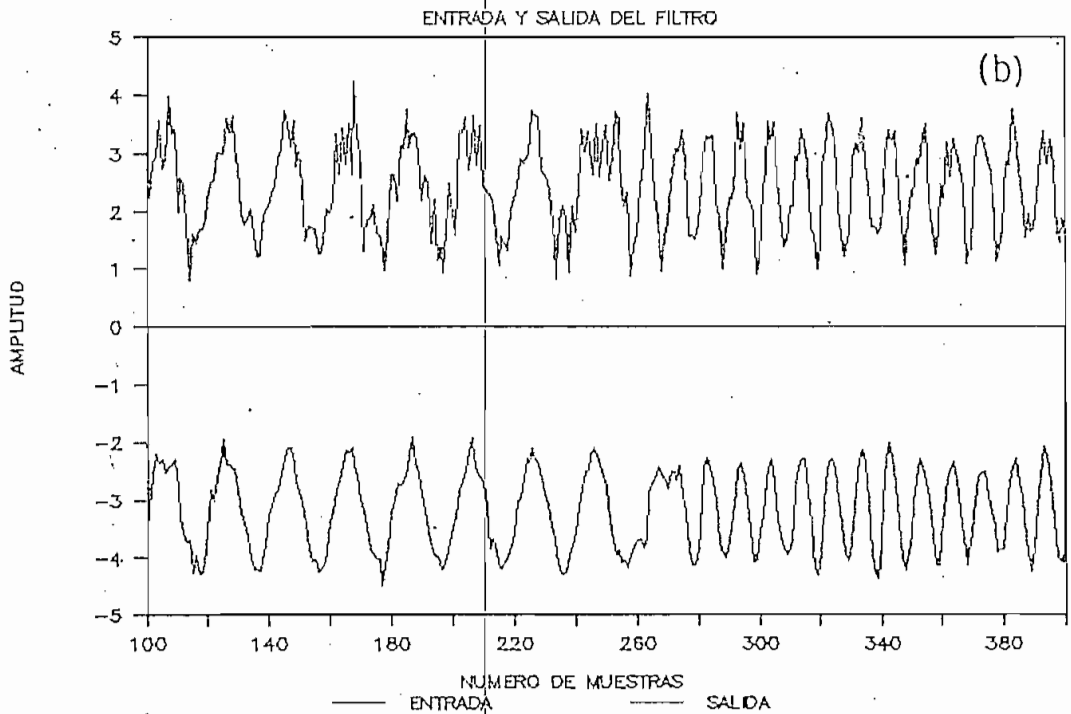
Ejemplo # 28

En este ejemplo durante el proceso se realizó cambios en la frecuencia de la señal periódica, tratándose de analizar si el sistema de cancelación se adapta a tales variaciones. Así pues, durante los 0.025 seg. de muestreo inicial se generó una senoide de frecuencia 500 Hz. y amplitud unitaria, luego en los 0.025 seg. restantes la frecuencia de la señal se varió a 1 KHz. y la amplitud se mantuvo en 1. El ruido coloreado contaminador de dicha señal fue el mismo de los ejemplos anteriores (25, 26 y 27). El filtro adaptivo utilizado fue de orden 16, $T = 8$, $\sigma = 10^2$ y $\phi = 0.998$.

CANCELACION ADAPTIVA DE RUIDO

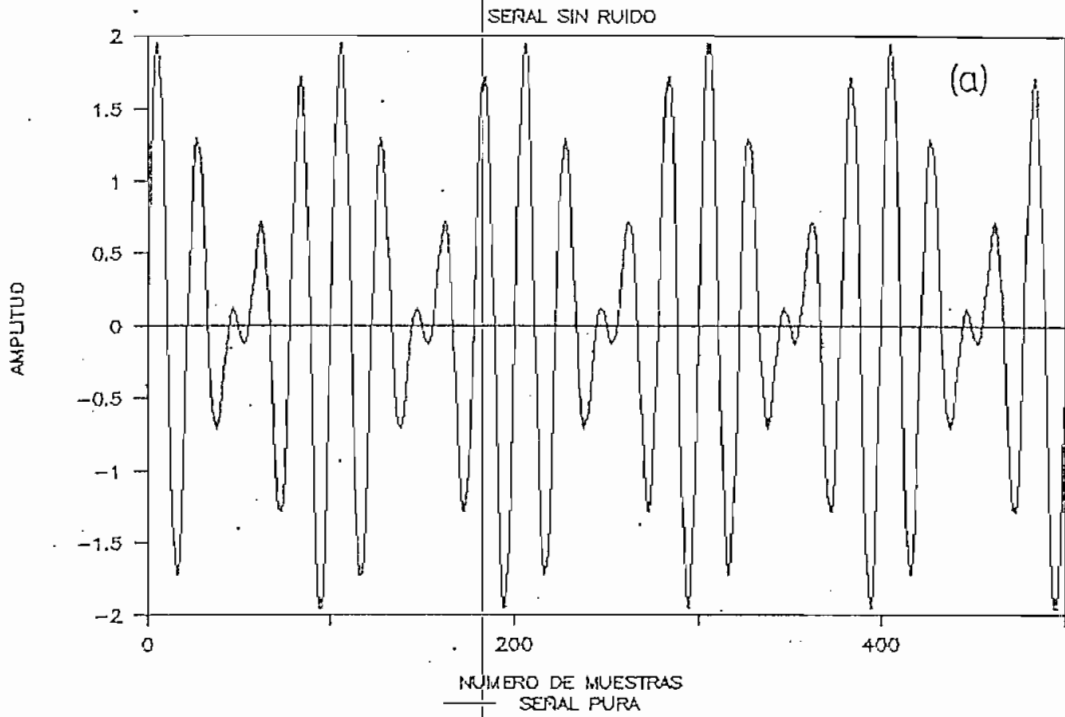


CANCELACION ADAPTIVA DE RUIDO

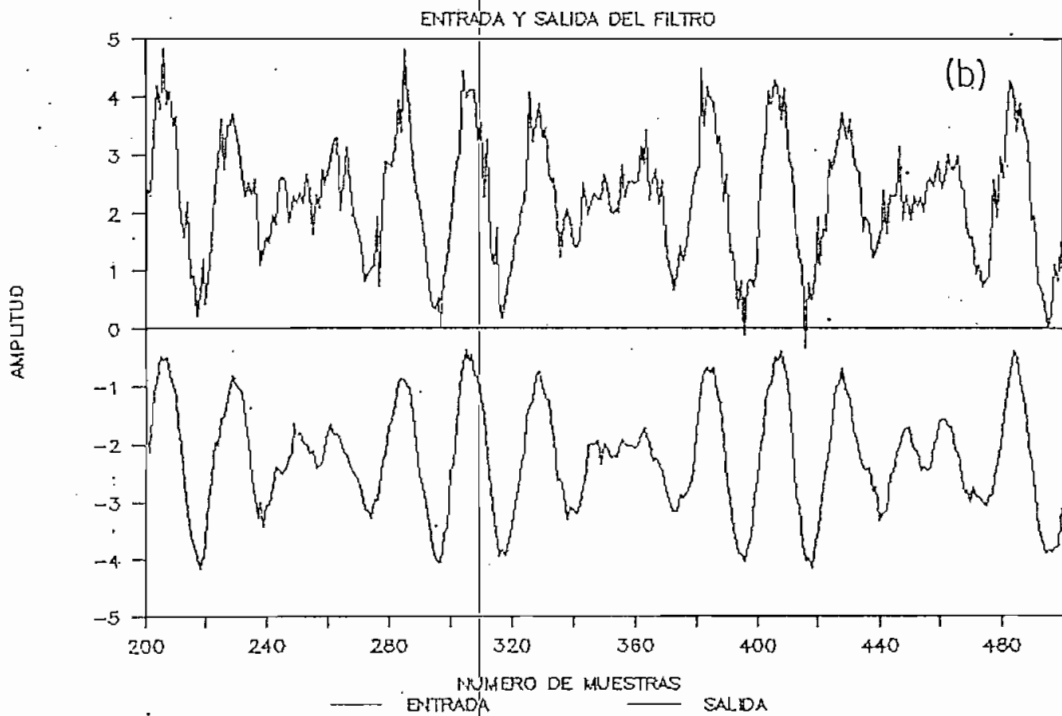


Ejemplo # 28 Procesamiento de una señal cuyas características de frecuencia varían. (a) Señal periódica libre de interferencia. (b) Comparación entrada/salida.

CANCELACION ADAPTIVA DE RUIDO



CANCELACION ADAPTIVA DE RUIDO

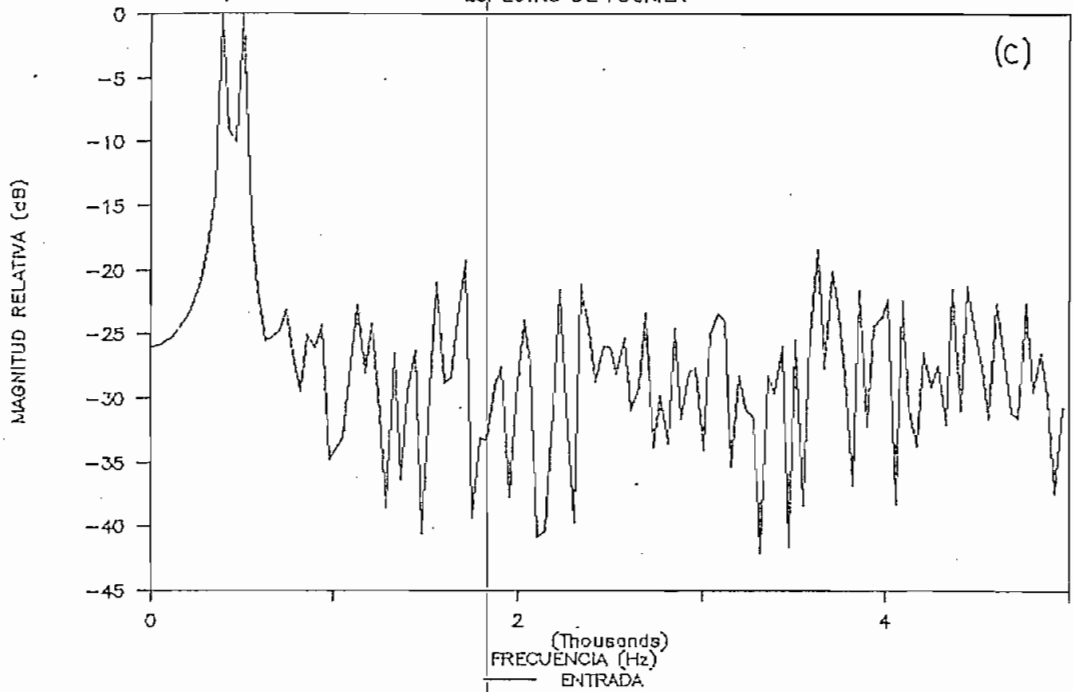


Ejemplo # 29

Procesamiento de una señal periódica constituida por la suma de dos sinusoides. (a) Señal periódica libre de ruido. (b) Comparación entrada/salida.

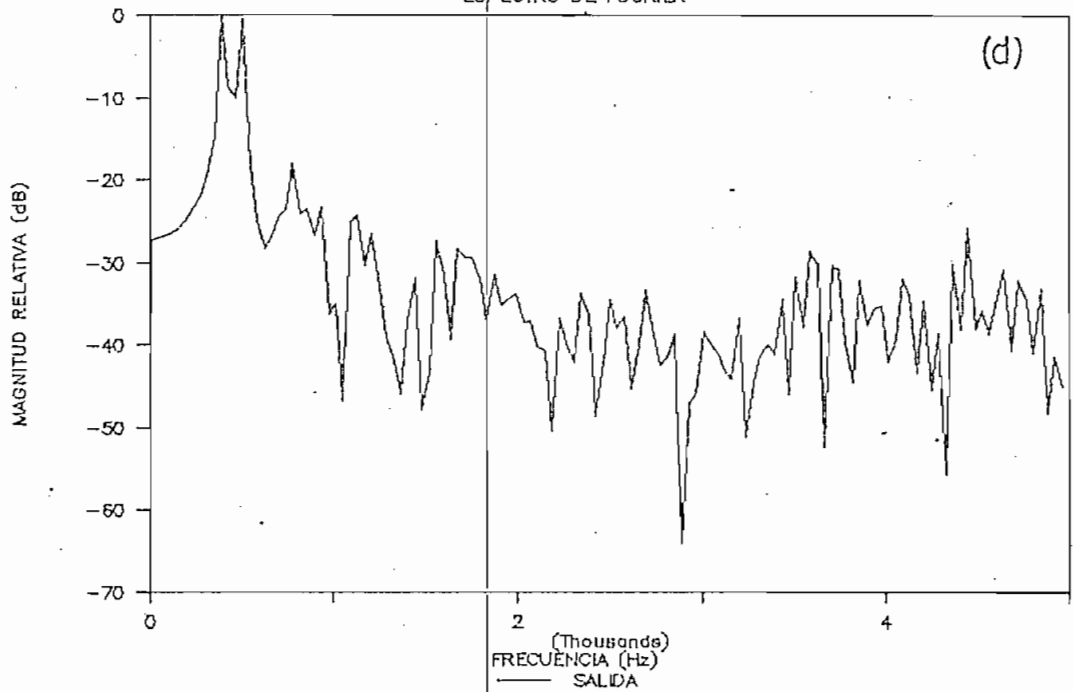
CANCELACION ADAPTIVA DE RUIDO

ESPECTRO DE FOURIER



CANCELACION ADAPTIVA DE RUIDO

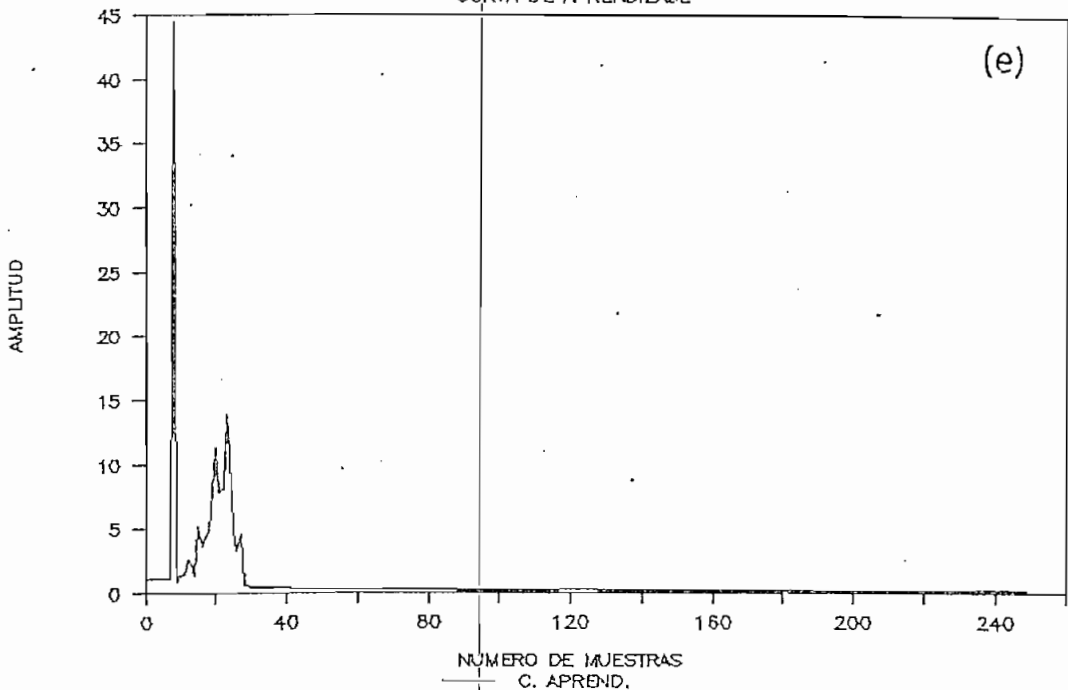
ESPECTRO DE FOURIER



Ejemplo # 29 (c) Análisis espectral de la señal de entrada. (d) Análisis espectral de la señal de salida.

CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE



Ejemplo # 29 (e) Curva de aprendizaje.

Ejemplo # 29

En este caso la señal periódica estuvo constituida por la suma de 2 sinusoides, la primera de frecuencia 400 Hz. y $A = 1$, la segunda de frecuencia 500 Hz. y amplitud igual a 1; la señal resultante se muestreó durante 0.05 seg. a una frecuencia de muestreo de 10 KHz., y se le añadió ruido coloreado de iguales características a las del ejemplo # 28. El filtro adaptivo utilizado fue el mismo del ejemplo anterior.

Los ejemplos desarrollados para el algoritmo RLS son prácticamente los mismos que para el algoritmo LMS, el comportamiento del filtro adaptivo RLS es análogo en cada caso al del LMS, de manera que las conclusiones se corresponden. Debe hacerse sin embargo una disgresión en cuanto a los ejemplos (22,23 y 24) que analizan la incidencia del

valor del factor de olvido, propio del algoritmo RLS, de ellos se desprende que su valor determina en última instancia la calidad de la señal filtrada, puesto que una elección inadecuada de su valor puede provocar pobres comportamientos del sistema, esto se debe a que el factor de olvido, ϕ , determina la memoria del algoritmo, de manera que para sistemas de la naturaleza de nuestra aplicación (variaciones lentas), es adecuado un valor de ϕ cercano a 1, donde prácticamente son tomados en consideración en un instante del proceso todos los datos anteriores en la determinación de los parámetros, al escoger un valor de ϕ pequeño (0.95 para el ejemplo # 24) la memoria del algoritmo se reduce a 20 muestras anteriores, con tal número de muestras lógicamente no es posible obtener buenos resultados como se observa en la figura (a) del ejemplo 24.

4.2 ANALISIS COMPARATIVO DE LOS RESULTADOS OBTENIDOS MEDIANTE LOS ALGORITMOS DESARROLLADOS

4.2.1 EFICIENCIA COMPUTACIONAL

La eficiencia computacional tiene que ver con la velocidad con la que un determinado algoritmo entrega resultados, al ser implementado en un computador. Constituye por tanto una característica de fundamental importancia en aplicaciones donde el tiempo de proceso es determinante; así, en un sistema en línea (on line) es de interés un tiempo de procesamiento mínimo, a fin de que lo más pronto posible se entreguen resultados confiables de la señal procesada, esto es de mayor interés aún en sistemas cuyas características varían rápidamente, donde es deseable que el sistema se reajuste casi inmediatamente a tales cambios.

Se trata de analizar el tiempo de ejecución que involucra cada algoritmo; para el efecto se corrieron varios ejem-

plos variando el número de muestras y el orden del filtro. Para el algoritmo LMS se probó con 200 , 400 y 600 muestras, en cada caso el orden del filtro se varió en el rango de 2 hasta 20. Para el algoritmo RLS así mismo se probó con 200 , 400 y 600 muestras, en este caso el orden del filtro se varió únicamente en el rango de 2 hasta 14, pues un orden mayor involucra un gran tiempo de procesamiento. Los resultados obtenidos se grafizaron y constan en la figura 4.1.

El análisis de la figura 4.1 permite concluir que el algoritmo LMS es más eficiente computacionalmente con respecto al RLS.

En el algoritmo RLS, contando el número de operaciones involucradas en un paso del mismo, provee una medida de su complejidad computacional, es posible implementar dicho algoritmo tal que requiera $2.5n^2 + 4n$ multiplicaciones y adiciones a cada paso de tiempo (siendo n el orden del filtro), para filtros de orden alto una complejidad computacional que se incrementa con el cuadrado del orden del filtro es a menudo inaceptable. Por el contrario en el algoritmo LMS son necesarias únicamente $2n + 1$ operaciones (multiplicaciones y sumas) a cada paso del tiempo, de manera que la complejidad computacional se incrementa con el orden del filtro. Estas relaciones se pueden observar en las figuras correspondientes a cada uno de los algoritmos, (figura 4.1) donde se muestra gráficamente que el tiempo de proceso requerido para el RLS es proporcional al cuadrado del filtro, en tanto que para el LMS la relación entre el orden del filtro y el tiempo de proceso es aproximadamente lineal. El número de operaciones involucradas en cada uno de los algoritmos (complejidad computacional) determinará su eficiencia.

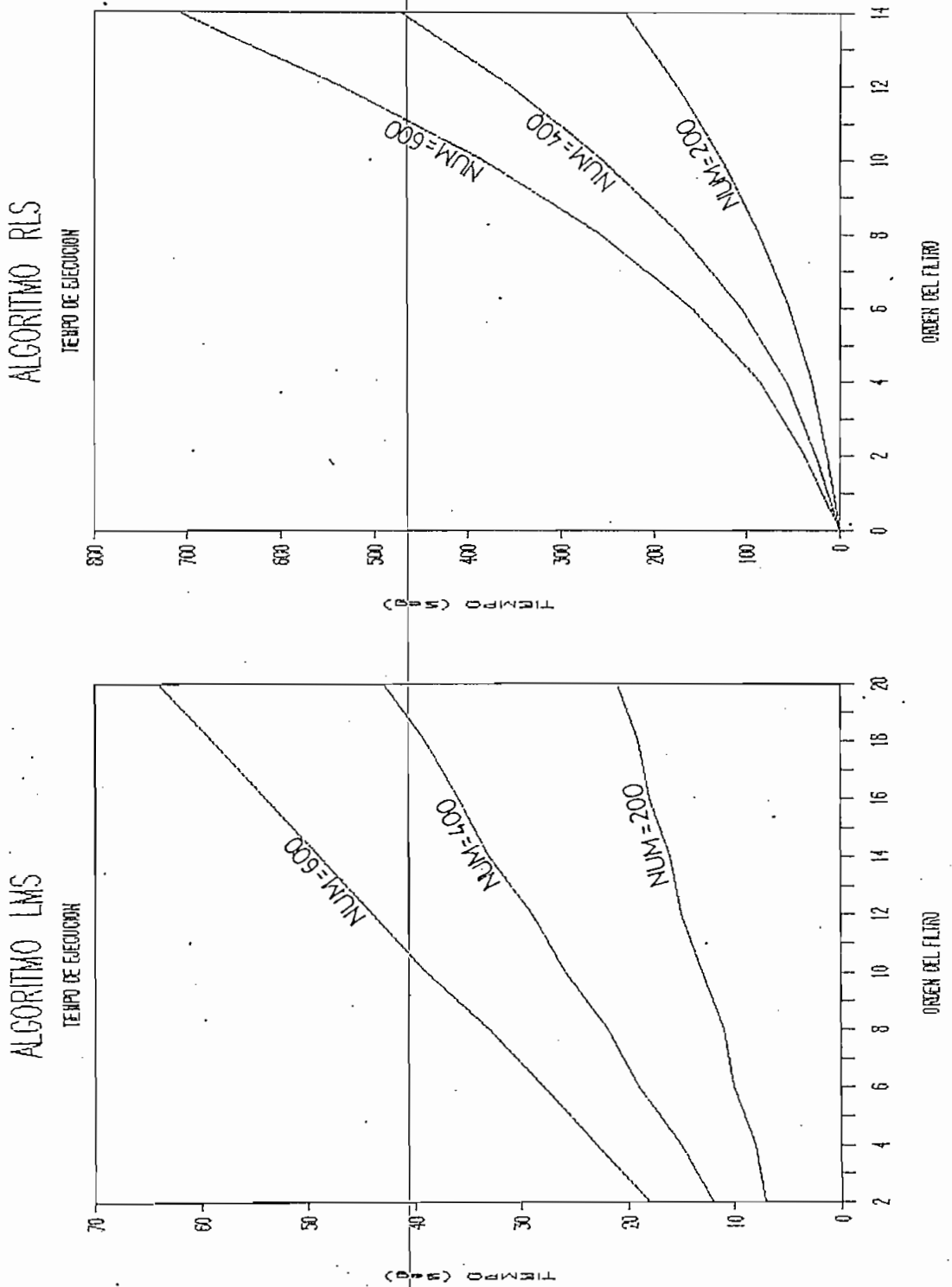


Fig. 4.1 Análisis del tiempo de ejecución empleado en el algoritmo LMS y RLS.

4.2.2 VELOCIDAD DE CONVERGENCIA

La curva de aprendizaje es una herramienta ideal que permite analizar la velocidad con la que se produce la convergencia en cada uno de los algoritmos implementados.

Como se observa en los ejemplos # 10, 11, 12 y 13, la velocidad de convergencia del algoritmo LMS está fuertemente afectada por la elección del parámetro μ , en tanto que el factor de olvido ρ tiene poco efecto en la velocidad de convergencia inicial del algoritmo RLS como lo muestran los ejemplos # 22, 23 y 24.

A continuación se introducirá un nuevo parámetro denominado factor de desajuste, en base al cual es posible realizar un análisis comparativo de la convergencia de los algoritmos implementados.

Factor de desajuste en los algoritmos adaptivos. M

Todo sistema adaptivo o sistema aprendiz capaz de adaptarse a velocidades en tiempo real, experimenta pérdidas en su comportamiento debido a que tal sistema se ajusta basándose en promedios estadísticos tomados sobre un número finito de muestras. Mientras más rápido se adapta el sistema, en general, más pobre será su comportamiento esperado luego de la adaptación.

Cuando se usa un algoritmo de adaptación, se genera un error cuadrático medio en exceso, pues de acuerdo con la ecuación (2.37) y (3.106) el error cuadrático medio que se genera, $E[\epsilon_j^2]$, es igual al error cuadrático medio mínimo más otro término. Una medida del desajuste del sistema adaptivo comparado con el sistema óptimo de Wiener (aquel cuyos coeficientes de ponderación para el filtro se calculan de acuerdo a la ecuación 2.13), se determina por la

razón del error cuadrático medio en exceso al error cuadrático medio mínimo, esta medida adimensional de la pérdida de comportamiento es conocida como el "factor de desajuste", M .

El factor de desajuste para el algoritmo RLS está dado por [4]:

$$M = \frac{n}{\Omega} \approx (1 - \phi)^n \quad (4.1)$$

n , es el orden del filtro
 Ω , es la longitud de memoria efectiva del algoritmo
 ϕ , es el factor de olvido y,
 M , es el factor de desajuste

En tanto que M , para el algoritmo LMS está dado por [4]:

$$M = \mu \text{ traza}[R_{xx}] = \mu n r_{xx}(0) = \mu n \text{ Var}\{x_j\} \quad (4.2)$$

n , es el orden del filtro
 μ , es el parámetro de convergencia
 x_j , son los datos de entrada.
 $\text{Var}\{x_j\}$, corresponde a la varianza de x_j .
 M , es el factor de desajuste.

Los dos algoritmos LMS y RLS pueden hacerse operar con el mismo factor de desajuste eligiendo los parámetros μ y ϕ , tal que:

$$1 - \phi = \mu \text{ Var}\{x_j\} \quad (4.3)$$

Si los datos de entrada se normalizan de manera que tengan varianza unitaria, la ecuación (4.3) quedará de la forma:

$$1 - \phi = \mu \quad (4.4)$$

Esta ecuación es útil para hacer una comparación del comportamiento de los dos algoritmos. A continuación se implementará un ejemplo al respecto.

Ejemplo # 30

Este ejemplo se desarrolla con el criterio anteriormente expuesto, a fin de poder hacer un análisis comparativo, sobre todo respecto a las propiedades de convergencia tanto del algoritmo LMS como del RLS.

Los datos se generan de modo que la varianza sea igual a la unidad. Sabemos que la potencia de ruido blanco gaussiano es igual a su varianza, σ_n^2 ; y, sea σ_s^2 la potencia de la senoide; de manera que para cumplir con la potencia entrada unitaria, esto es:

$$\sigma_s^2 + \sigma_n^2 = r_{xx}(0) = 1 \quad (4.5)$$

hacemos que la amplitud de la senoide sea igual a 1, en consecuencia la potencia correspondiente será 0.5 ; en igual forma el ruido contaminador lo generaremos con varianza igual a 0.5 ; de esta forma la potencia de entrada total será igual a 1.

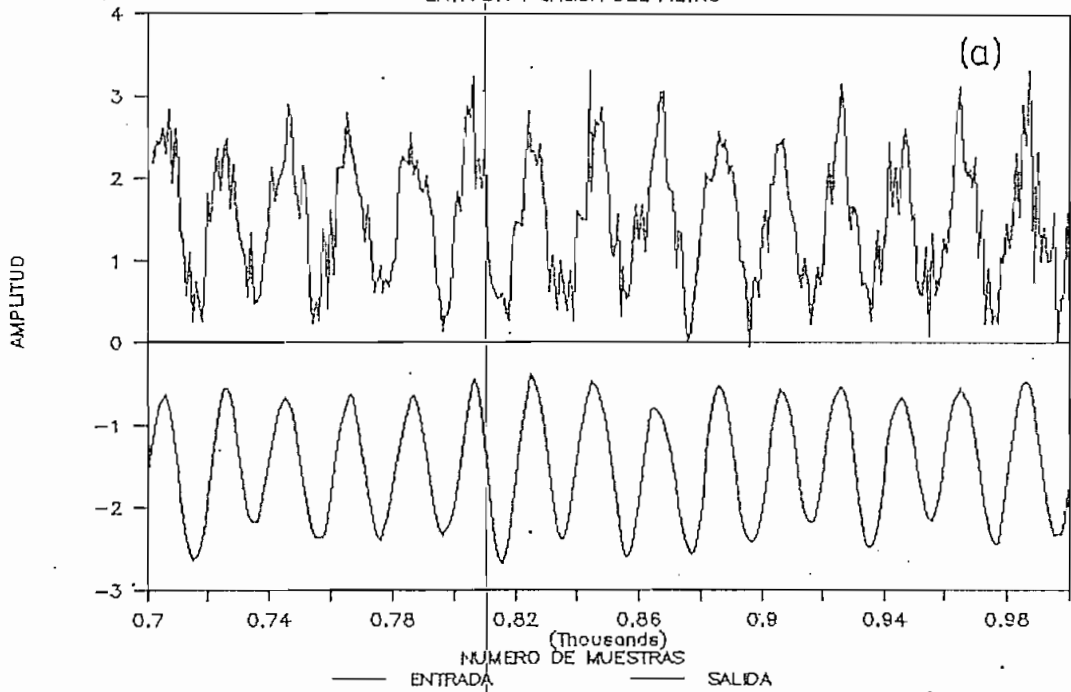
Se seleccionan los parámetro μ y ϕ tal que: $1-\phi = \mu$. Así, si $\phi = 0.998$, el valor para μ deberá ser $\mu = 0.002$

A fin de poder establecer comparaciones, se genera entonces una señal sinusoidal de frecuencia 100 Hz. , $A = 1$ (A es la amplitud), muestreada a 2 KHz. durante 0.5 seg. , lo que da un total de 1000 muestras de señal, a la cual se le suma ruido blanco aditivo de varianza 0.5. Ambos filtros se seleccionan de orden $= 10$, y longitud de retardo $= 5$. De los resultados obtenidos se muestran los gráficos correspondientes a la comparación entrada/salida y las curvas de aprendizaje respectivas.

Todos los ejemplos que se incluyen en el presente trabajo de tesis han sido desarrollados en un computador STAR XT, con una capacidad de 640 Kbytes de memoria, trabajando a 10 MHz.

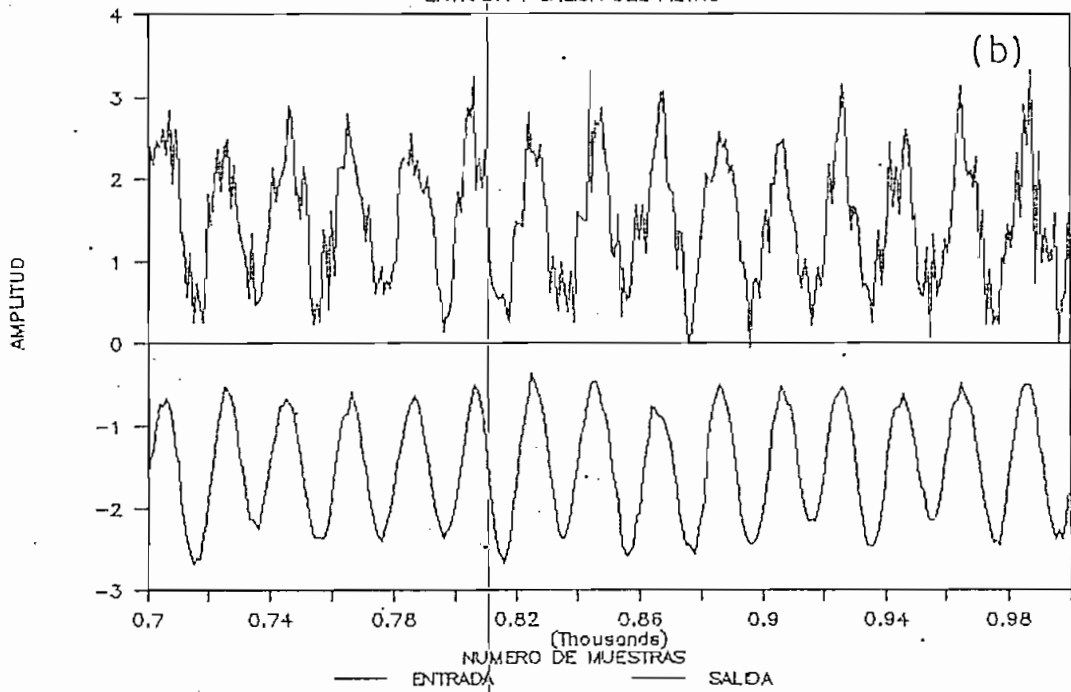
CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO



CANCELACION ADAPTIVA DE RUIDO

ENTRADA Y SALIDA DEL FILTRO

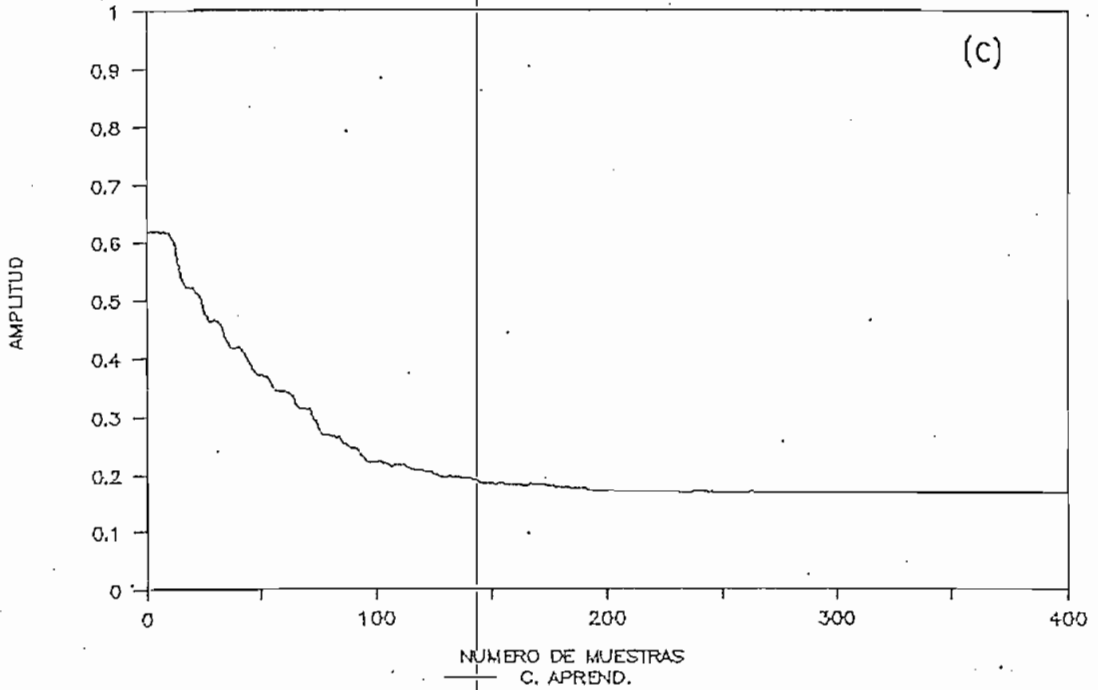


Ejemplo # 30

Procesamiento de una señal contaminada con ruido blanco, mediante el algoritmo RLS y LMS, con igual factor de desajuste. (a) resultados LMS. (b) resultados RLS.

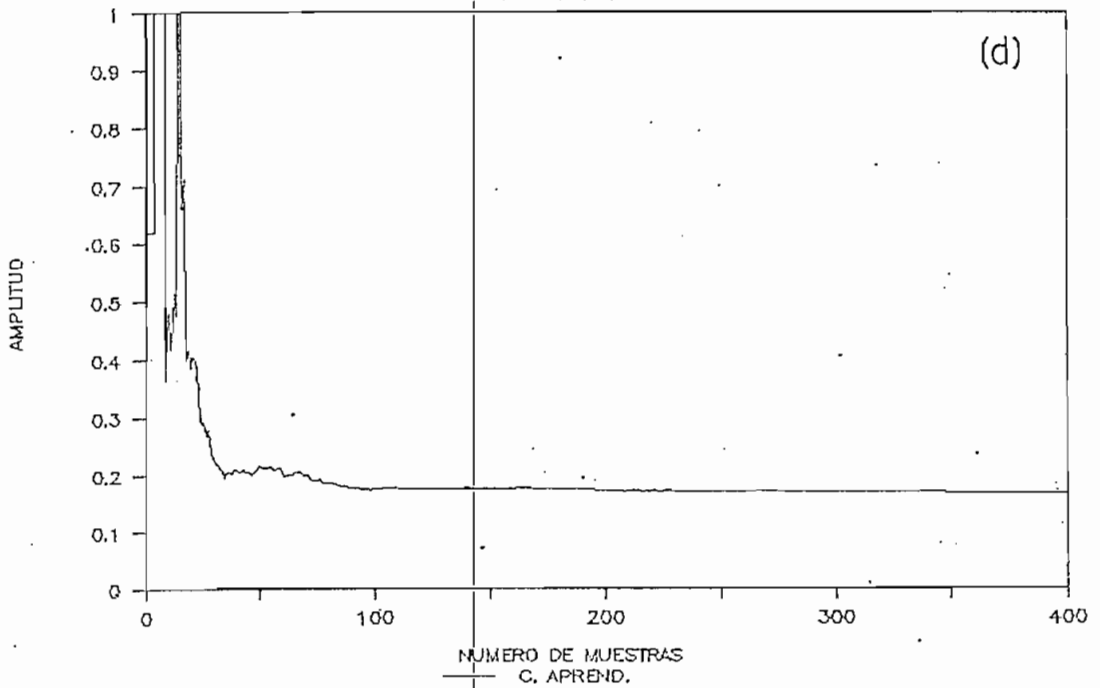
CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE



CANCELACION ADAPTIVA DE RUIDO

CURVA DE APRENDIZAJE



Ejemplo # 30 (c) Curva de aprendizaje para el filtro LMS. (d) curva de aprendizaje para el filtro RLS.

De los gráficos correspondientes a la curva de aprendizaje para cada uno de los algoritmos se desprende que La convergencia inicial del algoritmo RLS es considerablemente más rápida que la convergencia del LMS. Esto se deriva de la naturaleza de los algoritmos, pues el algoritmo LMS es una técnica de búsqueda del gradiente, en tanto que el algoritmo RLS es un procedimiento del tipo Newton-Raphson, y es bien conocido que éste último converge mucho más rápido.

La velocidad de convergencia es de fundamental interés durante la fase inicial, cuando el sistema adaptivo empieza a operar y cuando rastrea variaciones en la señal o en los parámetros del sistema.

4.2.3 CALIDAD DE LA SEÑAL FILTRADA

Finalmente y como parámetro muy importante en el análisis comparativo realizado respecto a los dos algoritmos implementados, se tiene la calidad de la señal filtrada, la cual muestra el nivel de reducción en el ruido conseguido luego del procesamiento adaptivo, y el nivel de distorsión que se produce en la señal filtrada. Para los dos métodos estudiados se observa de los resultados obtenidos en todos los ejemplos corridos, que efectivamente se produce una notable disminución en el nivel del ruido de la señal filtrada con respecto al ruido de la señal original, esto se refleja tanto en la calidad de la señal procesada como en los análisis espectrales respectivos.

4.3 CONCLUSIONES

A partir del estudio realizado en el presente trabajo de tesis se puede concluir lo siguiente:

- Las principales ventajas del método de cancelación adaptiva de ruido son: su capacidad adaptiva, bajo nivel de ruido de salida y baja distorsión de señal, siendo estas dos últimas características generalmente más bajas de lo que se puede conseguir con sistemas de filtrado convencionales.

- Con el presente trabajo de tesis se demuestra la eficiencia de un cancelador adaptivo de ruido, aplicado para eliminar una interferencia de banda ancha que contamina a una señal periódica, a pesar de que en esta aplicación no se dispone de una entrada de referencia externa. La correspondencia entre la señal pura y la salida del filtro adaptivo no es perfecta a causa de dos limitaciones fundamentales, así:
 - el filtro es de longitud finita
 - el filtro tiene una velocidad finita de adaptación.

- Mientras mayor sea el número de coeficientes de ponderación que se usen en el filtro, más cercana es su respuesta a la del filtro óptimo de Wiener. Sin embargo, incrementando el número de coeficientes el proceso adaptivo se torna mas lento y además se incrementa el costo de implementación; por tal razón se debe previamente analizar cuidadosamente los requerimientos en el comportamiento del filtro antes de que este pueda diseñarse para una aplicación particular.

- Del estudio comparativo realizado en la sección 4.2, se desprende que el algoritmo LMS es más eficiente computacionalmente que el algoritmo RLS; sin embargo la ventaja de este último radica en su velocidad de convergencia superior.
- La elección del parámetro μ en el algoritmo LMS determina la velocidad de convergencia de éste, valores de μ cercanos al límite superior producen adaptaciones rápidas pero generan un valor relativamente grande del error en estado estable; un valor de μ próximo a cero producirá un error en estado estable cercano al mínimo posible, pero la velocidad de convergencia se torna mucho mas lenta.
- El factor de olvido, ρ , determina la memoria efectiva del algoritmo RLS, por tanto la elección de este parámetro está íntimamente ligada con las características del sistema y de la señal, si estas varían lentamente es adecuado un valor de ρ cercano a la unidad, por el contrario si tales variaciones son rápidas es necesario un valor de ρ menor a la unidad.
- En general se puede decir que se ha logrado cumplir con el objetivo de la tesis, esto es proporcionar dos algoritmos que al ser incorporados en un sistema de cancelación de ruido, permitan cumplir una aplicación específica como es la de eliminar una interferencia de banda ancha (ruido blanco o coloreado) que contamina a una señal periódica. El principio de cancelación adaptiva de ruido es una técnica de variadas aplicaciones, las cuales incluyen la cancelación de varias clases de interferencias en electrocardiogramas, de ruido en señales habladas (speech), cancelación de interferencias en los lóbulos laterales de una antena, y de interferencias periódicas o de banda

ancha en las cuales no se dispone de una entrada de referencia externa.

4.4 RECOMENDACIONES

- Estudiar los filtros adaptivos de respuesta de impulso finita (FIR) con características de fase lineal, los cuales son de gran aplicabilidad en aquellos problemas donde la dispersión en frecuencia debida a la fase no lineal es dañina.
- Estudiar los filtros adaptivos IIR, y sus correspondientes algoritmos adaptivos.
- Sería conveniente disponer en la Facultad de Ingeniería Eléctrica de un sistema de adquisición de datos que se pueda adaptar al equipo de computación con que actualmente se cuenta. Tal configuración permitiría una implementación en línea de ésta y muchas otras aplicaciones que se pueden dar a los sistemas de cancelación adaptiva de ruido.

ANEXO A

MANUAL DE USO DEL PROGRAMA

En esta sección se dan algunas indicaciones útiles para el correcto uso y funcionamiento de los programas implementados en el presente trabajo de tesis.

A.1 OBJETIVO

El conjunto de programas que se detallan a continuación permite realizar la simulación completa de un cancelador adaptivo de interferencia de banda ancha en una señal periódica. Incluye rutinas que generarán una serie de datos (onda periódica, ruido blanco y ruido coloreado), cuya combinación constituirá la señal de entrada al sistema de cancelación; permitiendo seleccionar entre dos algoritmos adaptivos diferentes, el de mínimos cuadrados medios y el de mínimos cuadrados recursivos.

Adicionalmente se incluyen rutinas para análisis del comportamiento del filtro; estas son: una rutina para la determinación de la curva de aprendizaje y una rutina que permite realizar un análisis de los espectros de la señal de entrada y salida del cancelador.

A.2 REQUERIMIENTOS DE HARDWARE

Para la ejecución de los programas se debe disponer del siguiente equipo:

- Un computador del tipo IBM PC/XT/AT o compatible que reúna las características que se detallan a continuación:
 - a) mínimo 512 Kbytes de memoria
 - b) 2 unidades de diskette, o una unidad de disco duro y una de diskette.
 - c) monitor con capacidad de gráficos.
- Impresora.

A.3 REQUERIMIENTOS DE SOFTWARE

En dos diskettes, que se detallarán posteriormente, se encuentran grabados prácticamente todos los archivos necesarios para la ejecución del presente trabajo de tesis; requiriéndose adicionalmente lo siguiente:

- Un diskette del Sistema Operativo MS-DOS ver. 3.0 o más actualizada.
- El diskette PGRAPH del LOTUS 123 ver. 2.0 (para graficar los resultados en papel).
- Un diskette para datos.

No es necesario disponer del diskette del sistema LOTUS 123, pues los archivos necesarios para la rutina de gráficos ya se han incluido.

Contenido de los diskettes

Los programas y archivos necesarios se encuentran en dos diskettes, identificados como:

Disco #1 (Programas).- En éste se encuentran todas las rutinas para la simulación de los filtros adaptivos. El contenido se muestra a continuación. Los archivos con la extensión .BAS son los programas fuente, y a partir de ellos pueden recu-

perarse en caso necesario los archivos ejecutables.EXE.

DIRECTORIO DEL DISCO #1

Volume in drive A is PROGRAMAS
Directory of A:\

COMMAND	COM	23210	3-07-85	1:43p
FILTRO	BAT	42	2-04-88	10:09a
GRAFICO	BAT	1275	2-06-88	11:03p
KEY-FAKE	COM	536	9-17-85	12:07p
TESIS	BAS	11664	2-26-88	9:37a
DATGEN	BAS	19756	2-27-88	3:43p
ALGORITHM	BAS	12818	2-27-88	4:59p
CURVA	BAS	13265	2-26-88	9:53a
FOURIER	BAS	6625	2-26-88	10:09a
BASRUN20	EXE	63232	6-25-85	4:42p
TESIS	EXE	12087	2-27-88	2:54p
DATGEN	EXE	19687	2-27-88	5:03p
ALGORITHM	EXE	14039	2-27-88	5:04p
CURVA	EXE	14055	2-27-88	2:56p
FOURIER	EXE	8167	2-27-88	2:57p

15 File(s) 96256 bytes free

Disco #2 (Gráficos).- En el cual se encuentran todos los archivos necesarios para la grafización en pantalla de los resultados.

DIRECTORIO DEL DISCO #2

Volume in drive A is GRAFICOS
Directory of A:\

COMMAND	COM	23210	3-07-85	1:43p
KEY-FAKE	COM	536	9-17-85	12:07p
123	CMP	133848	9-25-85	1:23a
123	CNF	265	2-06-88	4:52p
123	EXE	9216	7-15-86	6:42p
123	SET	38447	10-31-87	8:47a
GRAFICO	WK1	8329	2-15-88	3:38p
123MONO	SET	38447	10-31-87	8:47a
GRAFICO1	WK1	8329	2-15-88	3:34p
123COLOR	SET	38466	5-06-87	4:56p
COLOR	BAT	99	2-22-88	6:54p
GRAFICO	BAT	1275	2-06-88	11:03p
MONO	BAT	104	2-22-88	6:56p

13 File(s) 54272 bytes free

A.4 DESCRIPCION DE LOS PROGRAMAS

FILTRO.BAT

Es el programa inicial de entrada, tiene la finalidad de reservar un espacio en memoria del computador para el utilitario KEY-FAKE.COM que será usado luego. Realizada esta operación se procede a transferir el control al programa maestro TESIS.EXE.

Programas de Procesamiento:

TESIS.EXE	Programa maestro, se encarga de coordinar y enlazar las diferentes rutinas.
DATGEN.EXE	Programa para generación de datos.
ALGORITHM.EXE	Programa de simulación de los filtros adaptivos.
CURVA.EXE	Programa para determinar la curva de aprendizaje.
FOURIER.EXE	Programa que realiza el análisis espectral de las señales de entrada o salida del filtro.

Estos programas han sido desarrollados usando el lenguaje BASIC, y luego compilados para hacerlos directamente ejecutables. Así pues se usaron los siguientes paquetes de software:

GWBASIC - 2.02. Microsoft 1984. Release 3.10
IBM BASIC COMPILER ver. 2.0

Para compilar los programas se usaron las opciones:
BASCOM/E/X/S/V.

GRAFICO.BAT

Cuando en el programa maestro se selecciona la opción de gráficos, se realiza una salida temporal del mismo y un enlace con el archivo GRAFICO.BAT. Este programa debe realizar el enlace con la hoja de cálculo LOTUS 123, que se usa para obtener resultados gráficos del proceso.

Es importante para facilitar el uso de los programas, que el operador ejecute el mínimo número de operaciones, de manera que estos se tornen accesibles a personas con muy poco conocimiento sobre programación al mismo tiempo que se reduce el número de errores por mal manejo de los paquetes de software. Con esta filosofía se ha tratado de obtener un funcionamiento en base a menús, donde el usuario ejecuta operaciones simples.

El programa GRAFICO.BAT pide que se cambie el disco de programas por el de gráficos, y luego arranca con la hoja de cálculo electrónica LOTUS 123.

Para evitar las operaciones iniciales de cargar la hoja de trabajo en donde se encuentran los macros y su ejecución, se ha incluido un utilitario que permite automatizar el proceso, pasando el programa directamente del menú principal al menú de gráficos.

El programa utilitario en mención es el KEY-FAKE.COM (PC Magazine Utilities, Vol I); el cual permite almacenar una secuencia de teclas inmediatamente antes de ingresar a una aplicación específica (en este caso el LOTUS), para luego ir entregando esa secuencia al programa conforme vaya éste pidiendo datos desde el teclado, hasta que se acaben las teclas almacenadas, el programa de aplicación las acepta como si fueran teclas presionadas ese instante.

La secuencia almacenada, permite cargar la hoja de trabajo GRAFICO.WK1 y luego ejecutar el macro inicial de arranque, hasta pasar al trabajo con menús.

Una vez que se han realizado los gráficos, esta rutina GRAFICO.BAT, permite volver al programa maestro, del que se había salido temporalmente.

Si no se ejecuta el programa KEY-FAKE.COM antes de ingresar al programa maestro (Generado en BASIC), cuando se sale temporalmente a la rutina de gráficos, y se intenta volver, ocurre un error, pues ese utilitario queda residente en memoria y va a superponerse con el espacio ocupado por el programa TESIS.EXE. Este problema se soluciona ejecutando el utilitario KEY-FAKE.COM antes de ingresar al programa maestro, con lo que queda reservada su zona de memoria. Esta operación la ejecuta el programa FILTRO:BAT.

GRAFICO.WK1

Hoja de trabajo generada en LOTUS 123, que contiene todos los macros (rutinas) necesarias para grafizar los resultados.

PROGRAMAS ADICIONALES INCLUIDOS

- archivos LOTUS necesarios para gráficos.
- hay dos archivos que se usan para configurar el programa dependiendo del tipo de monitor que se disponga, estos son:
 - MONO.BAT , el cual se debe ejecutar si se dispone de un monitor monocromático.
 - COLOR.BAT , se debe ejecutar si se dispone de un monitor a color.

Estos se encuentran en el disco #2 (GRAFICOS).

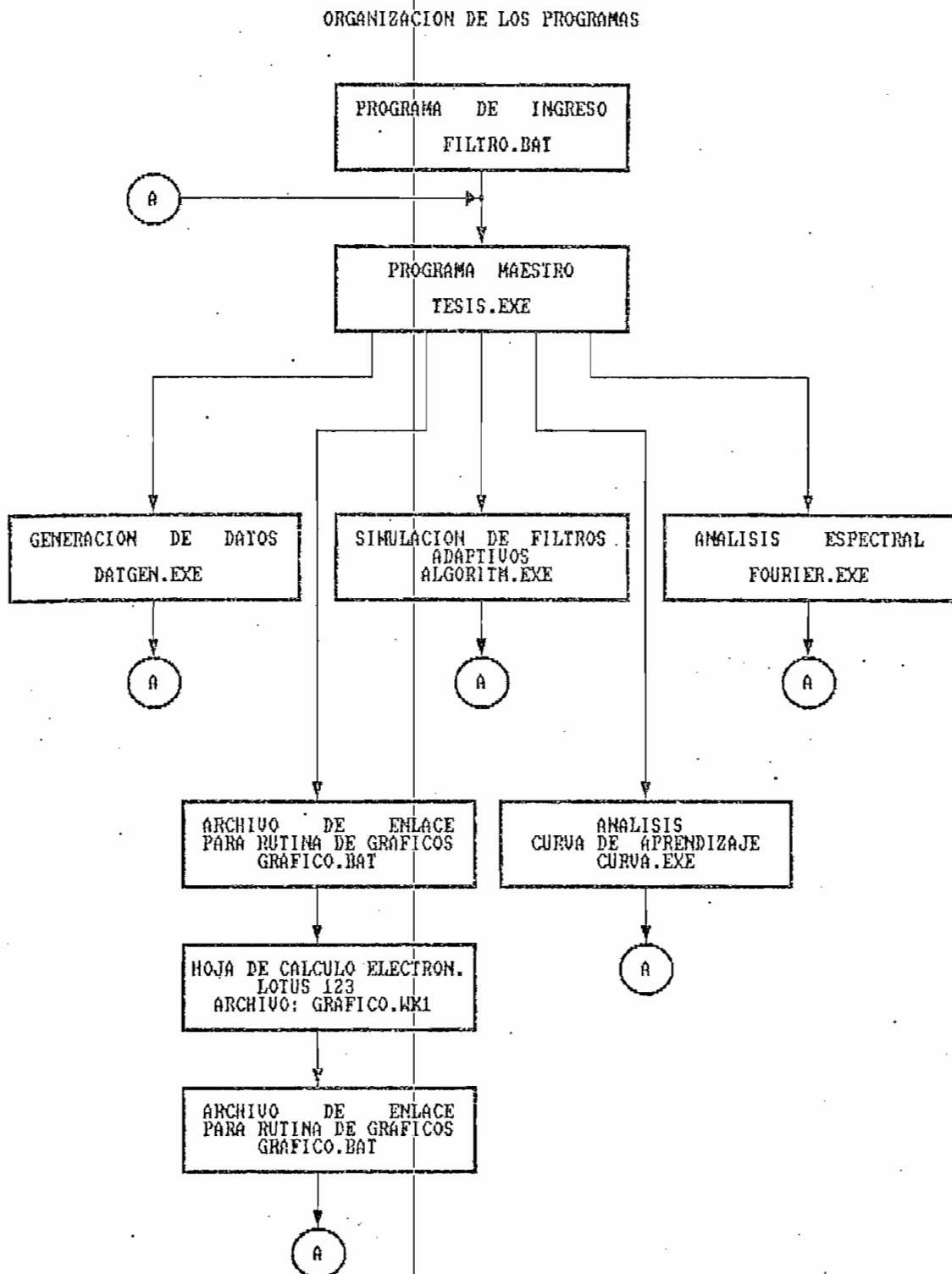


Fig. A.1 Organización de los programas.

A.5 EJECUCION DEL PROGRAMA

La organización de los programas se muestra en el diagrama de bloques de la figura A.1.

El procedimiento a seguir es el siguiente:

- Encender el computador, y la impresora si se dispone de una.
- Colocar el disco #1 "PROGRAMAS" en la unidad A y un disco formateado en la unidad B (disco de datos).
- Ejecutar el programa FILTRO.EXE, para lo cual se usa el comando¹:

FILTRO ←

- De esta forma se ingresa al programa principal TESIS-.EXE, se expone un rótulo de presentación e inmediatamente se pasa al menú principal, con las siguientes opciones:

- 1.- Unidad del disco de datos
- 2.- Generación de datos de entrada
- 3.- Simulación de filtros adaptivos
- 4.- Análisis del desempeño del filtro
- 5.- Grafización de resultados
- 6.- Terminar el programa

Lo expuesto anteriormente se esquematiza en forma general en el diagrama de flujo de la figura A.2

Opción 1 del menú principal:

Siempre debe ejecutarse esta rutina al ingresar al programa, pues sirve para definir en qué unidad de disco estarán localizados los archivos de datos. Se reconocen 3 unidades

¹ ← con esta simbología se representa el presionar la tecla ENTER.

válidas: A, B o C. Su operación se sintetiza en el diagrama de flujo de la figura A.3

Opción 2 del menú principal:

Sirve para generar los datos de entrada al filtro, el primer dato que pide es el nombre del archivo de salida, se debe ingresar un nombre válido de 8 caracteres máximo (sin extensión, pues el programa automáticamente le coloca la extensión .DIN).

El programa verifica si existe el archivo, de ser así pregunta si se desean recuperar los datos generados anteriormente o reemplazarlos; en caso de que no exista el archivo o se desee reemplazar los datos de uno ya existente, se ejecuta la rutina DATGEN.EXE. Todo esto se muestra gráficamente en el diagrama de flujo de la figura A.4(a).

Al ejecutarse la rutina DATGEN.EXE, como se muestra en el diagrama de la figura A.4(b), se solicitan los siguientes datos:

- Número de sinusoides, que pueden ser entre 0 y 3 (0 indica que no hay señal, sólo ruido).
- Las frecuencias y amplitudes de las sinusoides.
- La frecuencia de muestreo.
- Tiempo de muestreo.

y se procede a calcular la señal.

Puede añadirse a continuación una señal con otras características de amplitud y frecuencia (se simula un cambio en la señal para comprobar como se reajusta el filtro), en este caso se solicita nuevamente se ingresen los datos an-

teriormente especificados, a excepción de la frecuencia de muestreo.

A continuación se pasa a la generación de ruido blanco, y se pide la varianza del mismo (si la varianza es cero, no se generará ruido), con tal dato se procede a generar el ruido blanco.

A continuación se ofrece la alternativa de generar ruido coloreado, lo cual se realiza pasando ruido blanco a través de filtros digitales Butterworth, los mismos que pueden ser del tipo:

- 0.- Ruido sin filtrar (blanco)
- 1.- Filtro Pasabajos.
- 2.- Filtro Pasaaltos.
- 3.- Filtro Pasabanda.

Con la opción 0 no se filtra el ruido; con las otras opciones se solicita el ingreso de las frecuencias de corte respectivas y el orden del filtro deseado. Se ejecuta a continuación el filtrado.

Los resultados tanto de la generación de la señal como del ruido se almacenan en un archivo con extensión .DIN, en este punto se presenta la opción para sacar un reporte impreso con los datos de la señal y el ruido. Se retorna luego al menú del programa principal.

Opción 3 del menú principal:

Para ingresar a esta opción deben haberse ejecutado las opciones 1 y 2.

Primeramente deberá configurarse el filtro, para el efecto se solicitan los siguientes datos:

- Tipo de algoritmo: LMS o RLS.
- Orden del filtro.
- Número de retardos

Si se seleccionó el algoritmo RLS se solicita:

- Factor de olvido
- Constante de inicialización

Independientemente del tipo de algoritmo que se haya seleccionado deberán ingresarse los siguientes datos:

- Número de muestras a procesarse, esto debido a que pueden generarse un gran número de muestras de datos, pero no necesariamente procesarlas todas.
- Número de muestras a almacenar, es útil cuando se desea observar los resultados en estado estable (se almacenan las últimas muestras procesadas).

Este proceso se esquematiza en el diagrama de flujo de la figura A.5(a).

Luego se pasa a la rutina de ejecución de los algoritmos, la cual consta en el diagrama de flujo de la figura A.5(b) y en la que se pide los siguientes datos:

- Nombre del archivo de salida, automáticamente el programa coloca la extensión .DOT.
- Si se ha seleccionado el algoritmo LMS, se realiza un cálculo del parámetro de convergencia crítico y se da la opción para escoger el valor de dicho parámetro con el que se ejecutará el proceso.

Se ejecutan a continuación los algoritmos y se almacenan los resultados en el archivo correspondiente, nuevamente se presenta aquí la opción de obtener un reporte impreso de la conformación del filtro.

Una vez ejecutado el algoritmo correspondiente, se enlaza con el programa principal y se retorna al menú.

Opción 4 del menú principal:

En la cual es posible ejecutar dos rutinas para análisis del comportamiento del filtro, su operación se muestra en el diagrama de flujo de la figura A.6(a), y en ella se ofrece un submenú:

- 1.- Curva de aprendizaje.
- 2.- Análisis de Fourier de la señal de entrada.
- 3.- Análisis de Fourier de la señal de salida.
- 4.- Retornar al menú anterior.

Opción 1: Con los datos de la configuración del filtro (opción 3 del menú principal) se procede a calcular la curva de aprendizaje, para el efecto se ejecuta la rutina CURVA.EXE, los resultados se almacenan en un archivo con la extensión .CUR. La operación de esta rutina se sintetiza en el diagrama de flujo de la figura A.6(b)

Opción 2: Realiza un análisis de Fourier de la señal de entrada, esto lo hace mediante la rutina FOURIER.EXE, los resultados se almacenan en un archivo con extensión .FIN.

Opción 3: Realiza un análisis de Fourier para la salida del filtro, nuevamente por medio de la rutina FOURIER.EXE, almacenándose los resultados en un archivo con extensión .FOU.

El diagrama de flujo de la figura A.6(c) muestra gráficamente el funcionamiento de la rutina FOURIER.EXE.

Opción 4: Retorna al menú principal

Opción 5 del menú principal:

Permite la grafización de los resultados. Desde el programa TESIS.EXE se realiza una salida temporal (SHELL"GRAFICO.BAT") al sistema operativo y en éste al programa GRAFICO.BAT, al hacerlo se solicita un cambio al disco #2 de Gráficos.

De inmediato se entra al LOTUS 123 y a los menús de gráficos que se han estructurado de acuerdo a la Figura A.8.

Es entonces posible obtener los siguientes gráficos:

- | | |
|----------|--|
| entrada | - Ruido |
| | - Señal periódica |
| | - Señal + ruido |
| salida | - Entrada al filtro |
| | - Salida del filtro |
| | - Comparación (entrada y salida simultáneamente) |
| Análisis | - Curva de aprendizaje |
| | - Análisis de Fourier (entrada) |
| | - Análisis de Fourier (salida) |

Al ingresar a cualquiera de estas posibilidades, se debe seleccionar cuál archivo se va a visualizar, el programa muestra sólo los del tipo adecuado para la selección.

El resto del proceso se reduce simplemente a ubicarse en la opción deseada, usando las teclas de movimiento del cursor, y a presionar la tecla ENTER (↵); o a presionar directamente la primera letra de la opción deseada.

En cada caso se da la opción de almacenar los gráficos en archivos .PIC, para imprimirlos luego usando el programa PGRAPH.EXE que es parte del sistema LOTUS 123.

Finalizada la sesión de gráficos se retorna al menú principal.

Opción 6 del menú principal:

Termina la ejecución del programa y retorna al sistema operativo.

NOTA.-

En todas aquellas secciones donde es necesario ingresar datos, se ha incluido una indicación que señala el rango posible de los mismos, en caso de que el dato ingresado no se encuentre dentro de lo especificado, éste no será aceptado y se solicitará su reingreso.

En todos los programas se han incluido rutinas de intercepción de posible errores, algunos de ellos devuelven la línea de programa en que se produjo el error y el código del mismo; se recomienda en esos casos revisar los listados adjuntos a este trabajo y el manual "BASIC REFERENCE-IBM Corp. Version 3.0".

En algún caso de error grave del que no se pueda recuperar el programa (por ejemplo daño en algún programa del disco) se recomienda reinicializar el sistema.

ARCHIVOS DE DATOS

La extensión que se va dando a cada uno de los archivos durante la ejecución del programa, los identifica plenamente así:

Extensión	Contenido
.DIN	archivo que contiene los datos de entrada al filtro.
.DOT	archivo que contiene los datos de salida del filtro.
.CUR	archivo que contiene los puntos de la curva de aprendizaje.
.FIN	archivo que contiene el análisis espectral de una señal de entrada al filtro.
.FOU	archivo que contiene el análisis espectral de una señal de salida del filtro.
.PIC	archivos que contienen gráficos.

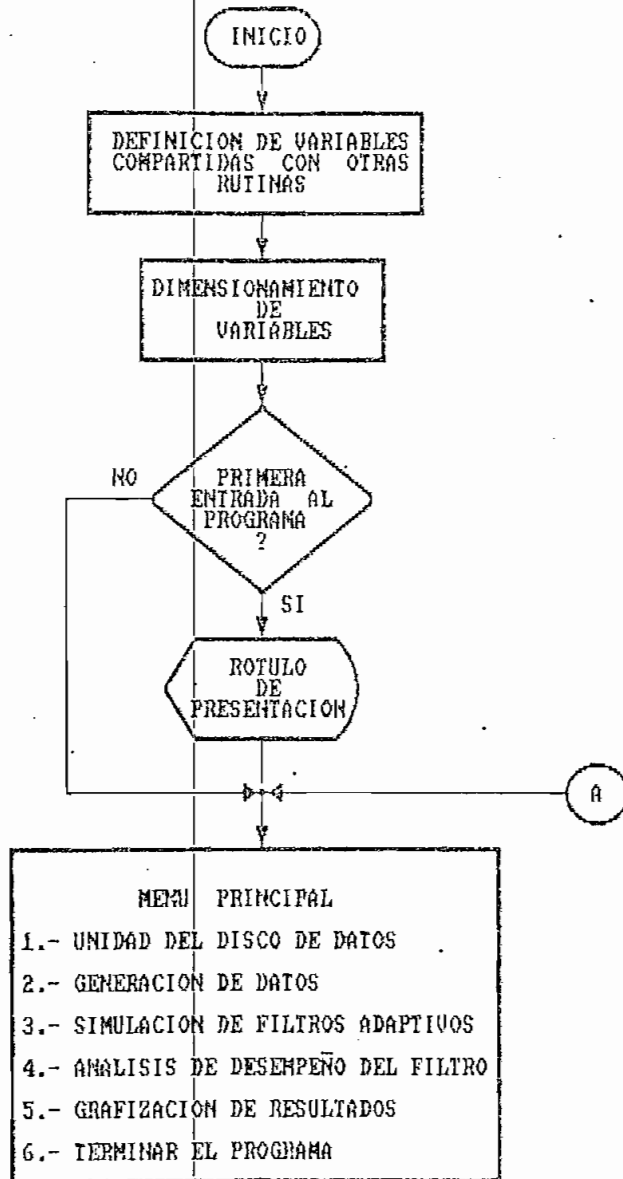


Fig. A.1 Diagrama de flujo del programa principal TESIS.EXE

OPCION 1 DEL MENU PRINCIPAL

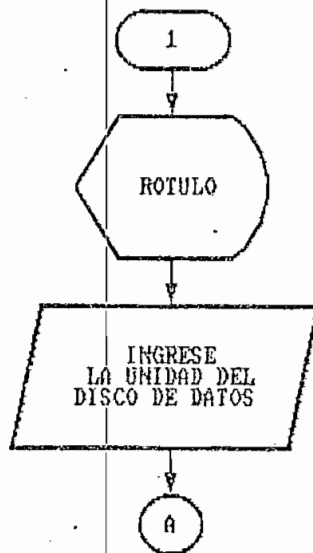
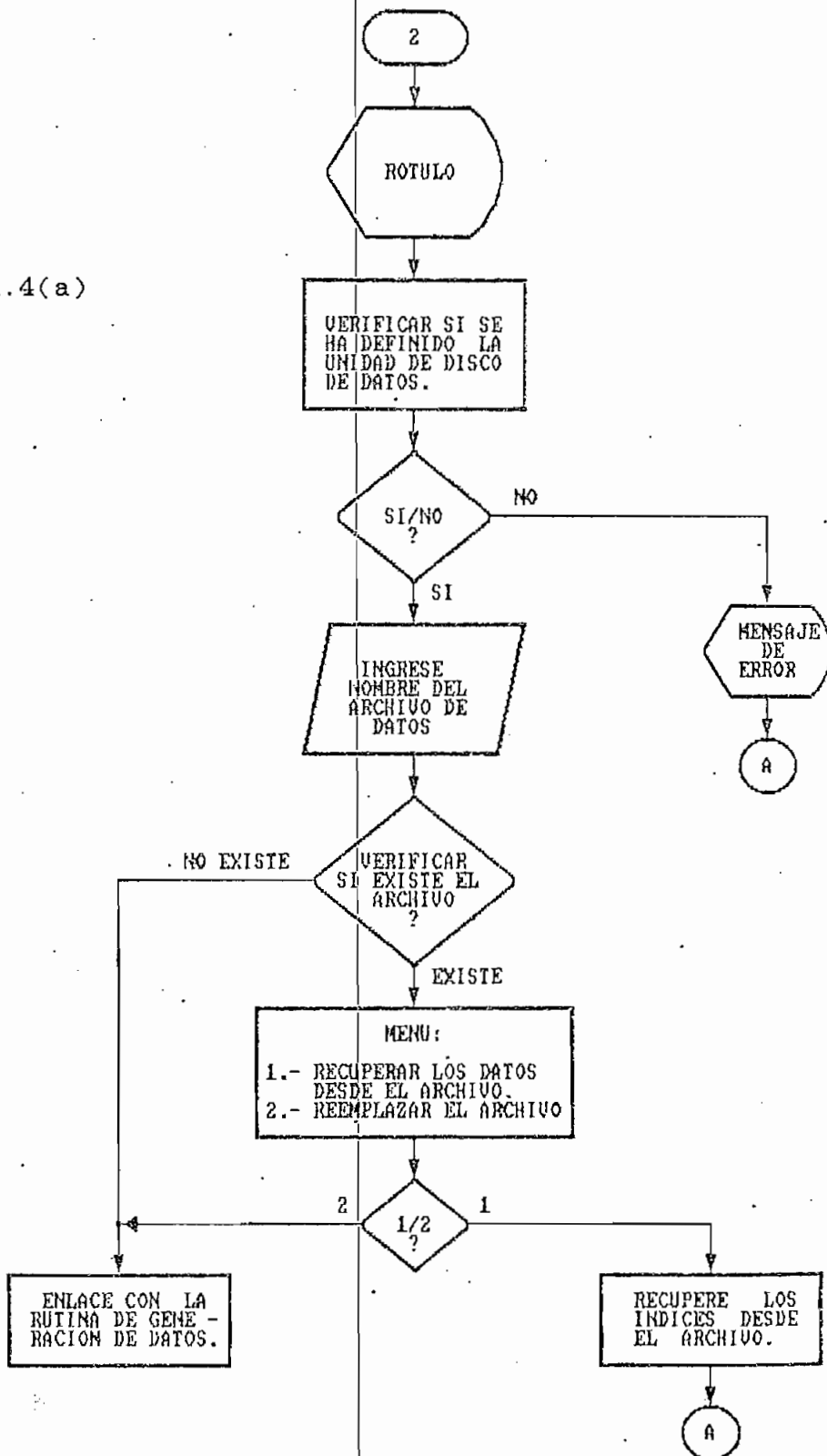


Fig. A.3 Diagrama de flujo de la opción 1 del menú principal.

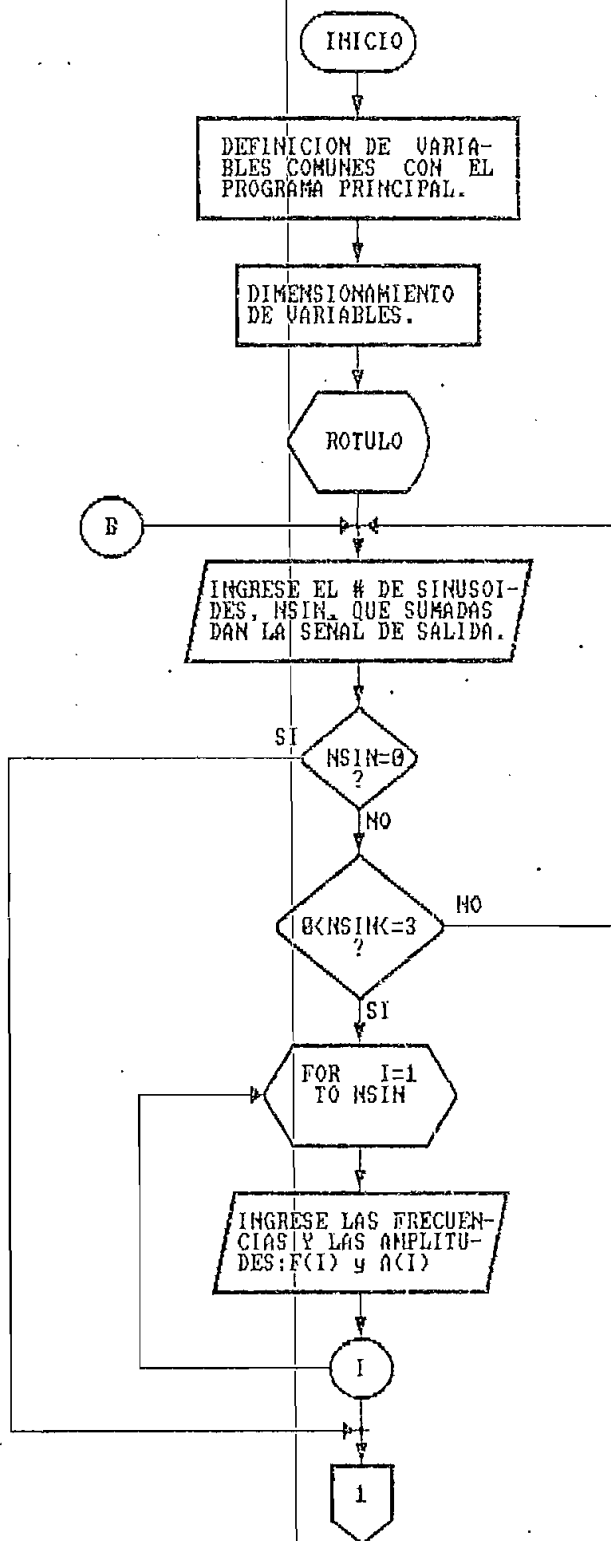
OPCION 2 DEL MENU PRINCIPAL

Fig. A.4(a)

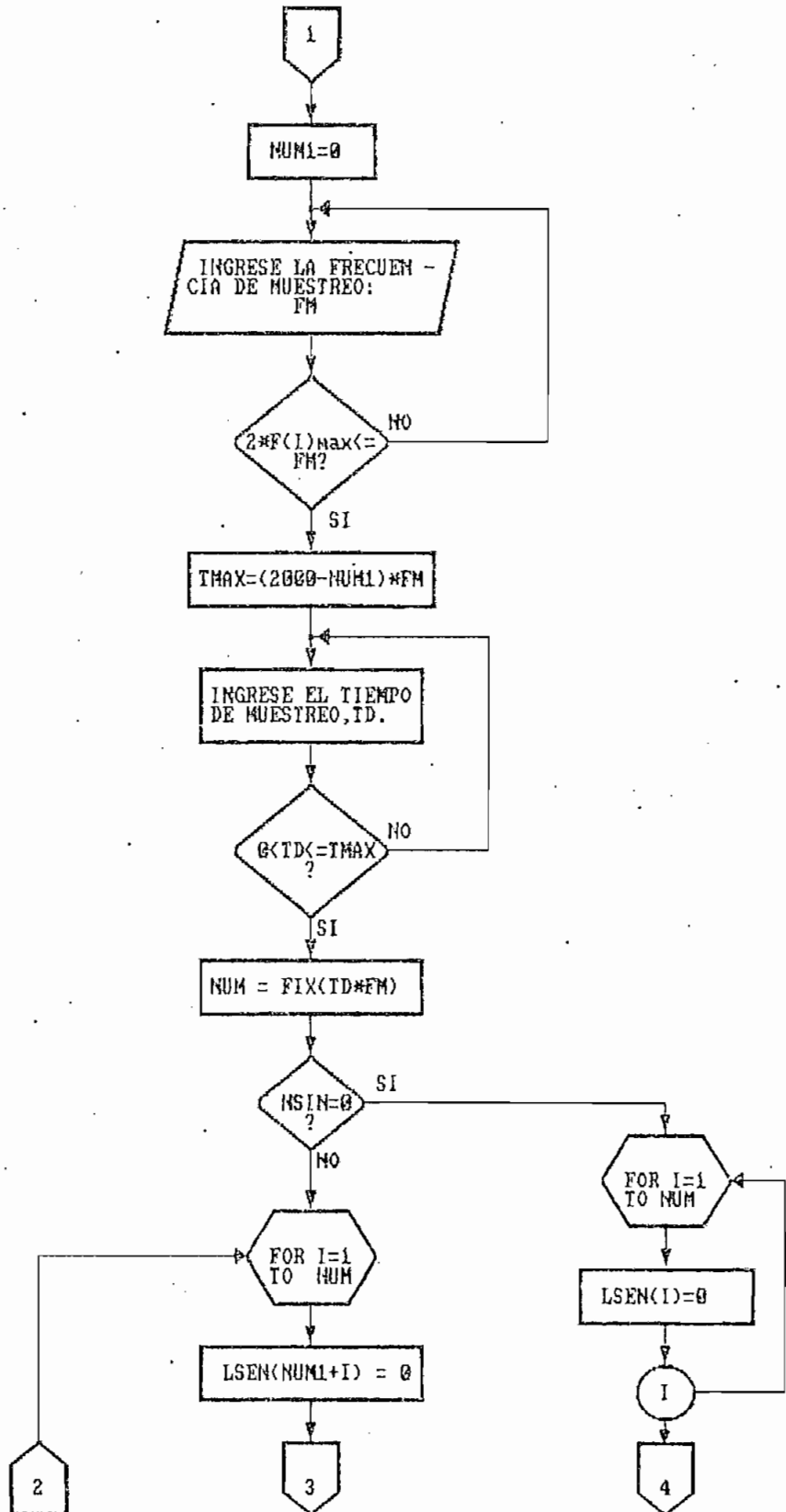


RUTINA PARA LA GENERACION DE DATOS DATGEN.EXE

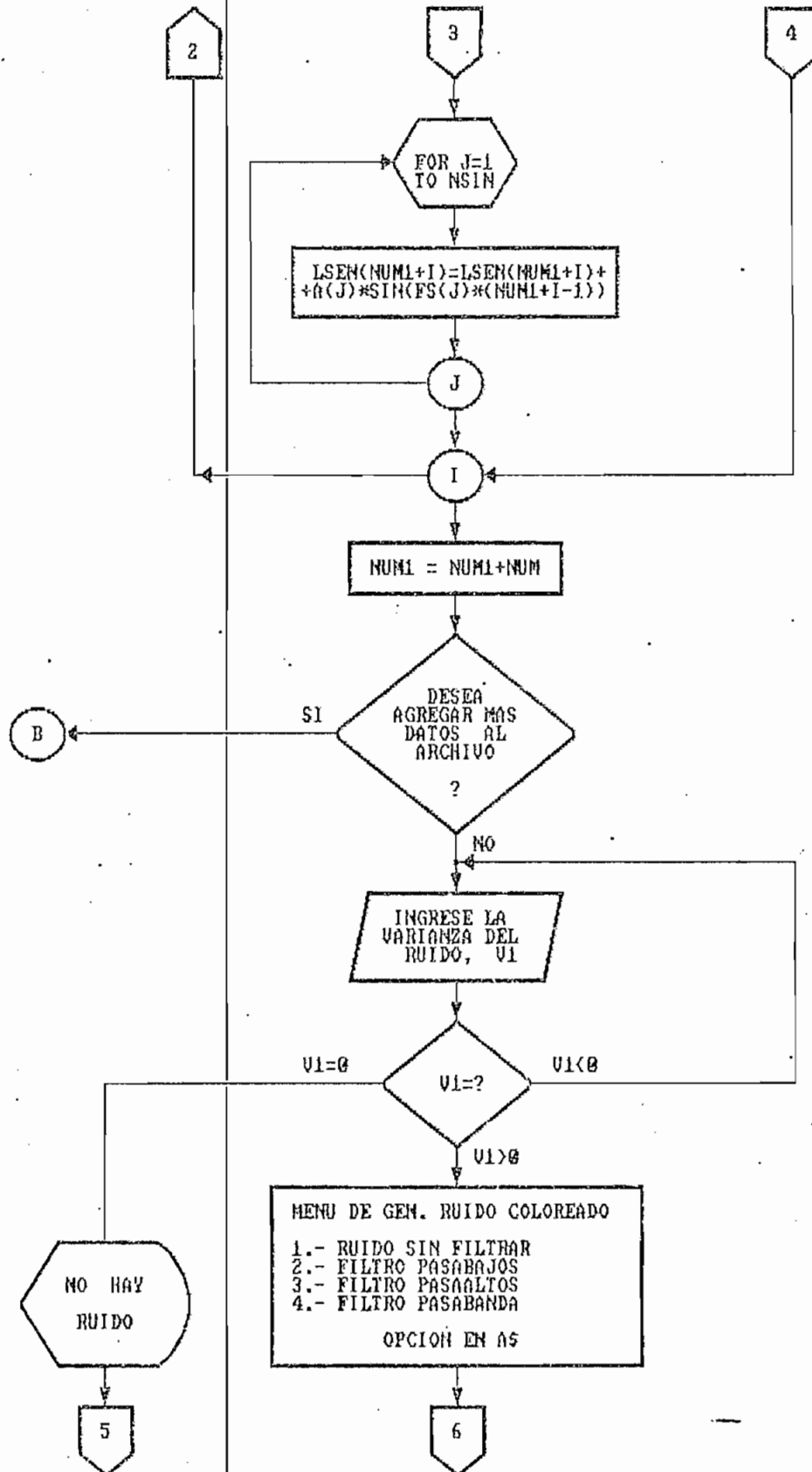
Fig. A.4(b)



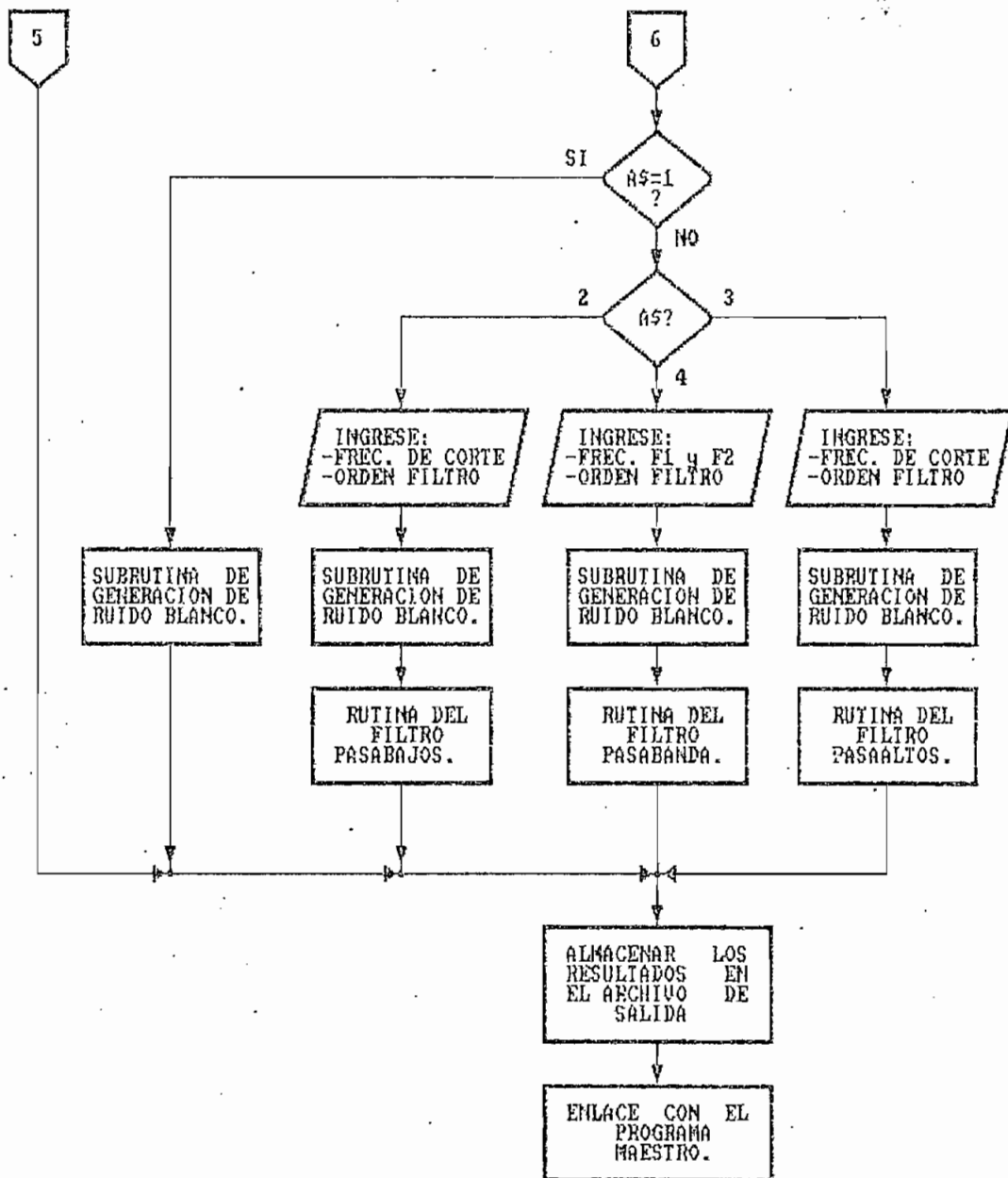
..CONTINUACION



...CONTINUACION

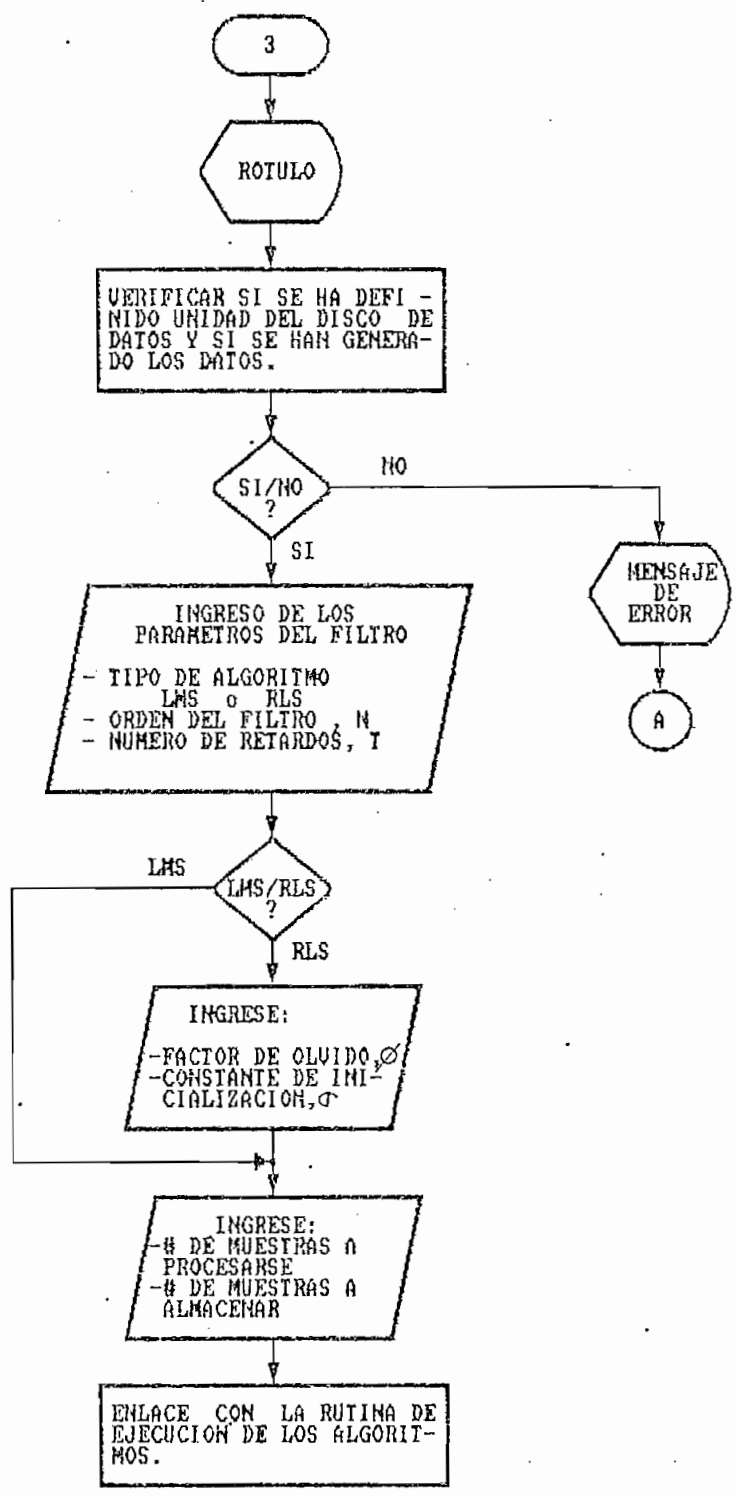


...CONTINUACION



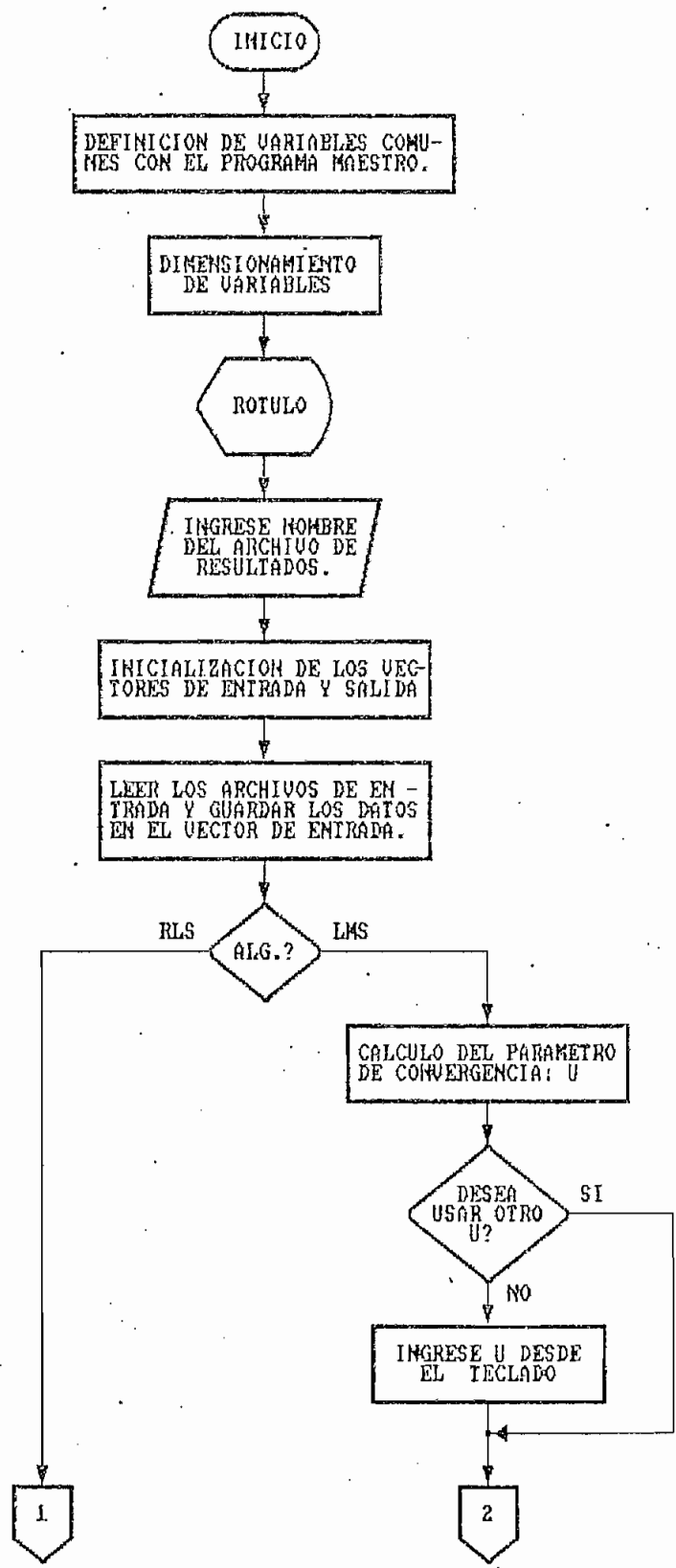
OPCION 3 DEL MENU PRINCIPAL

Fig. A.5(a)

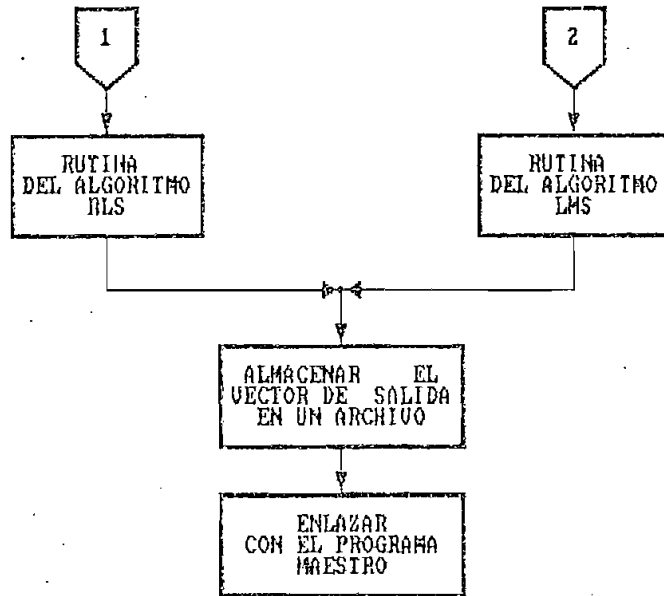


RUTINA DE EJECUCION DE LOS ALGORITMOS ALGORITHM.EXE

Fig. A.5(b)



...CONTINUACION



OPCION 4 DEL MENU PRINCIPAL

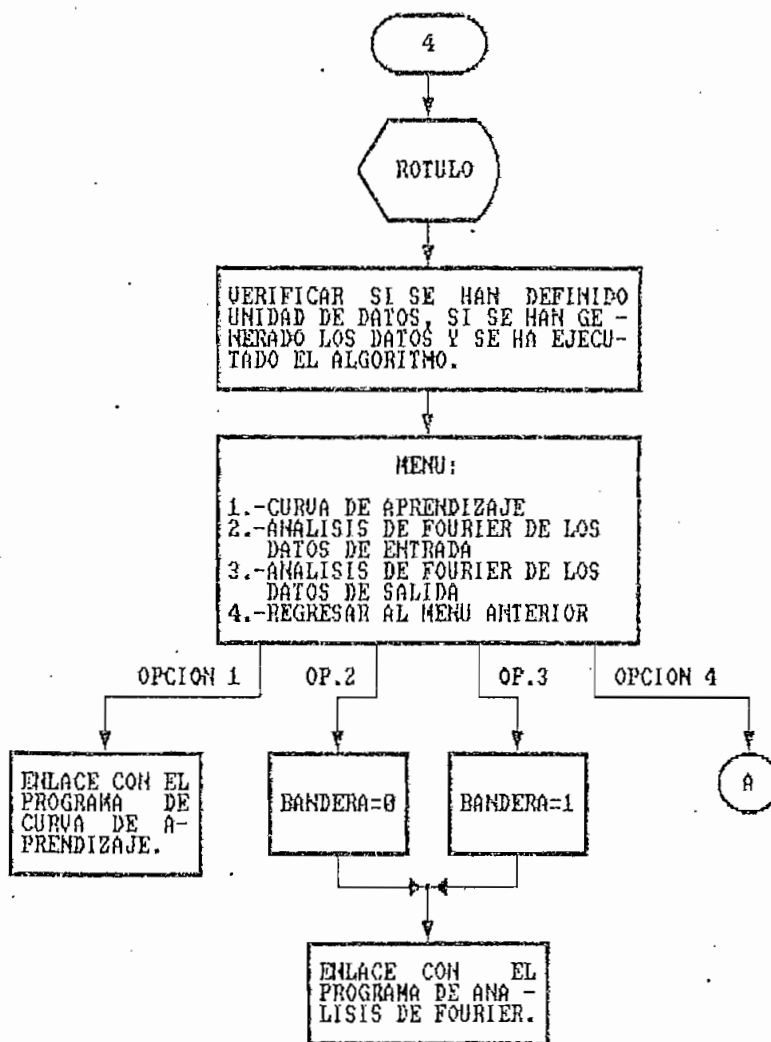
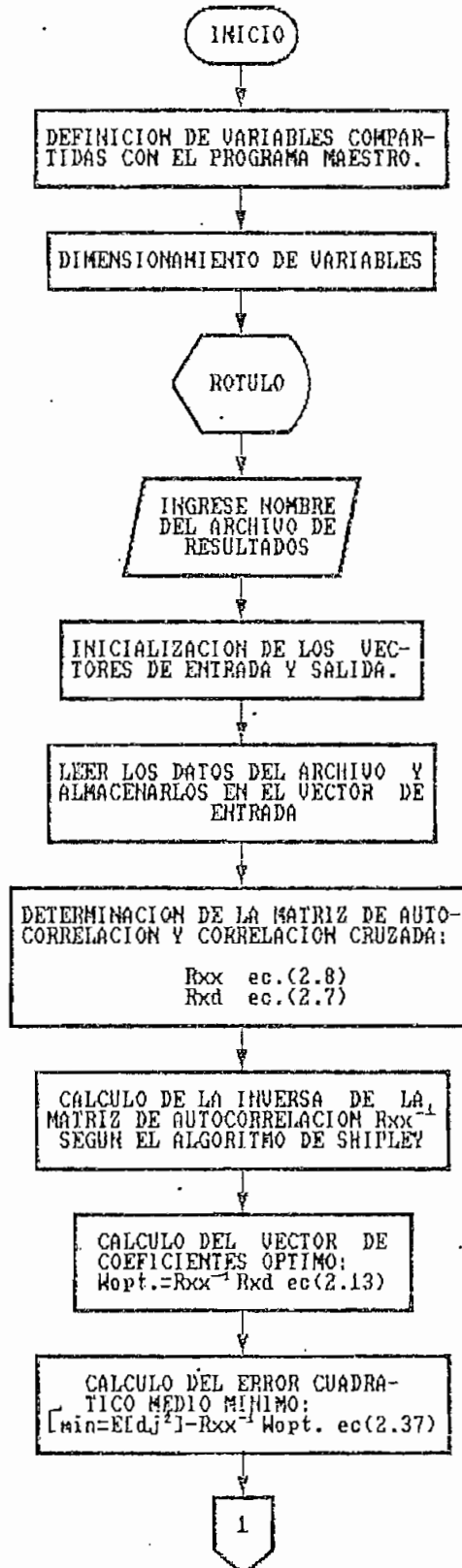


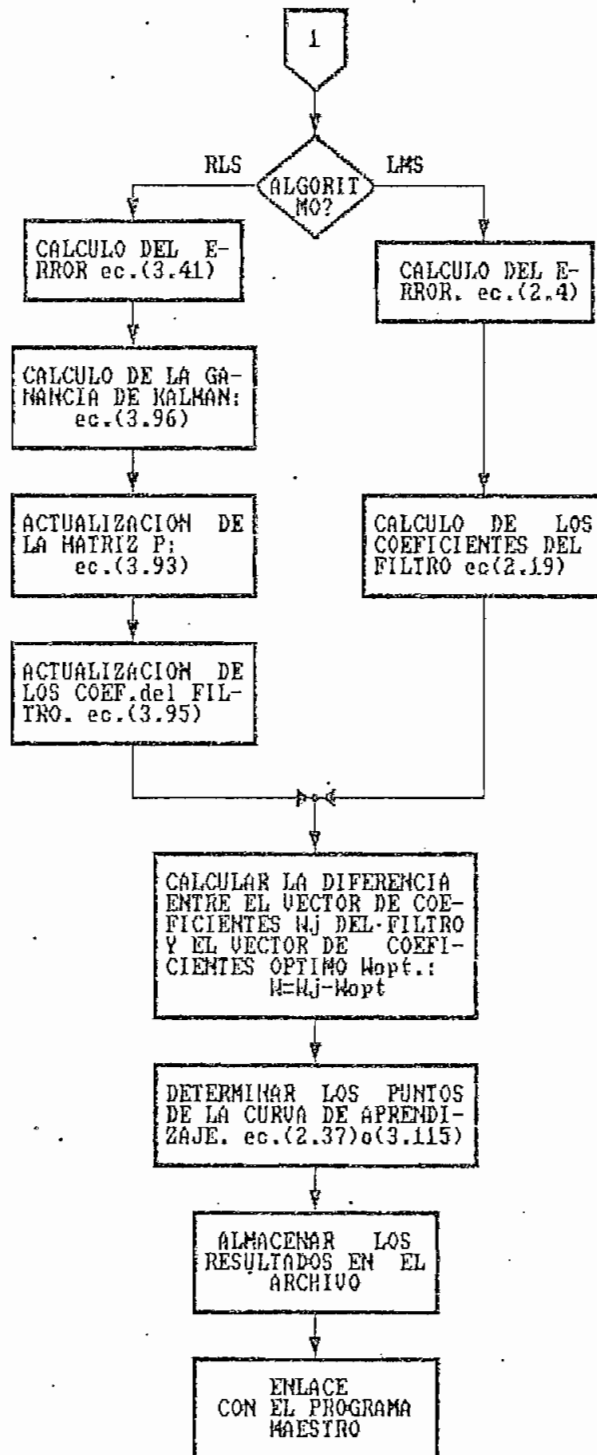
Fig. A.6(a) Diagrama de flujo de la opción 4 del menú principal.

RUTINA PARA DETERMINACION DE LA CURVA DE APRENDIZAJE
CURVA.EXE

Fig. A.6(b)

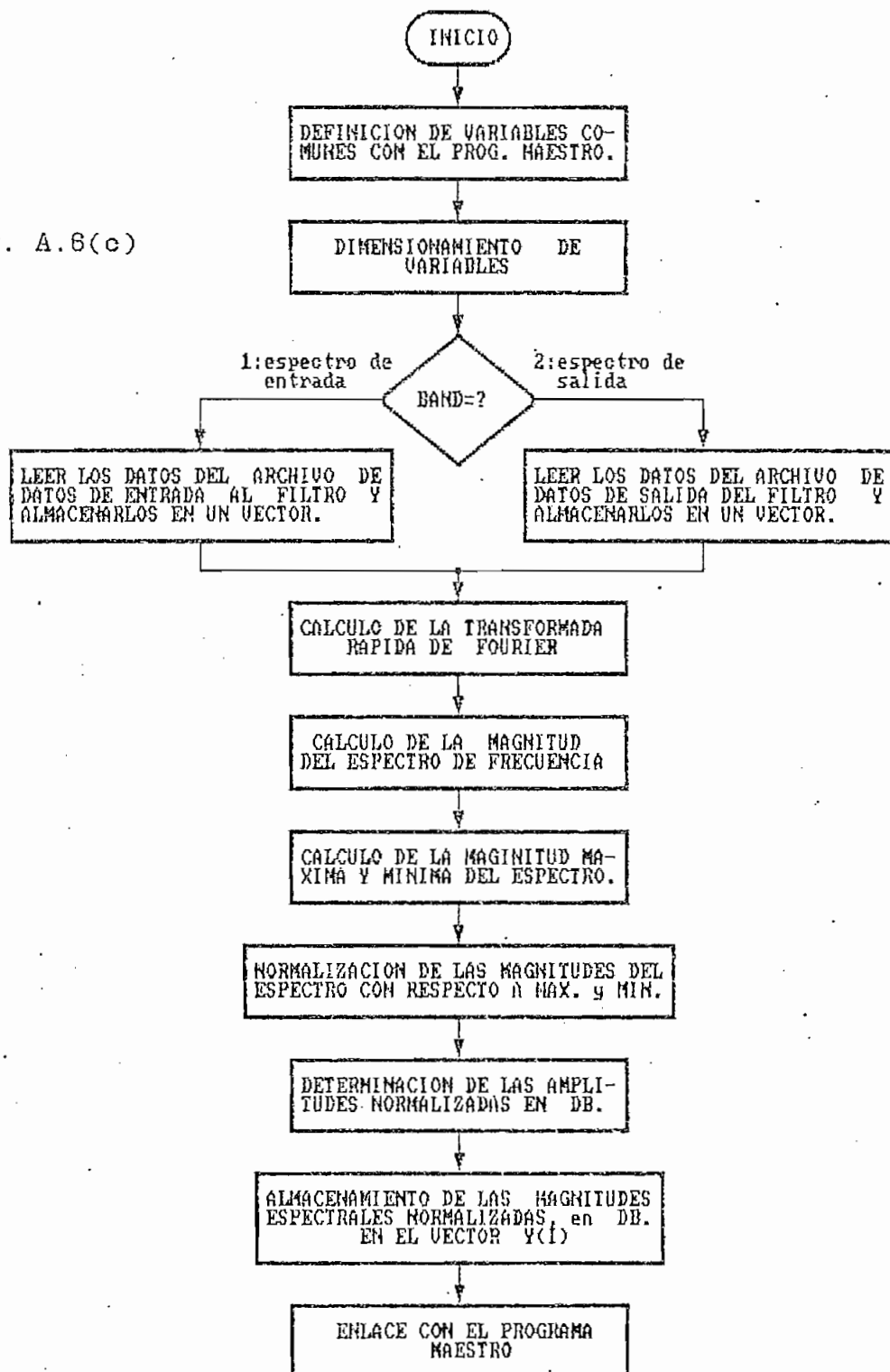


...continuacion

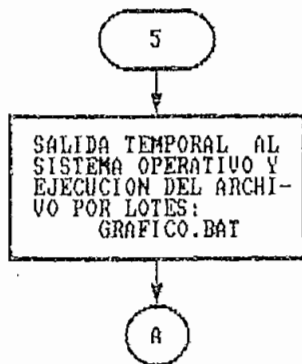


RUTINA PARA CALCULO DE LA TRANSFORMADA RÁPIDA DE FOURIER FOURIER.EXE

Fig. A.6(c)



OPCION 5 DEL MENU PRINCIPAL



OPCION 6 DEL MENU PRINCIPAL

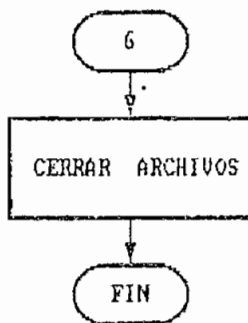


Fig. A.7 Diagramas de flujo de las opciones 5 y 6 del menú principal.

MENUS DE GRAFICOS

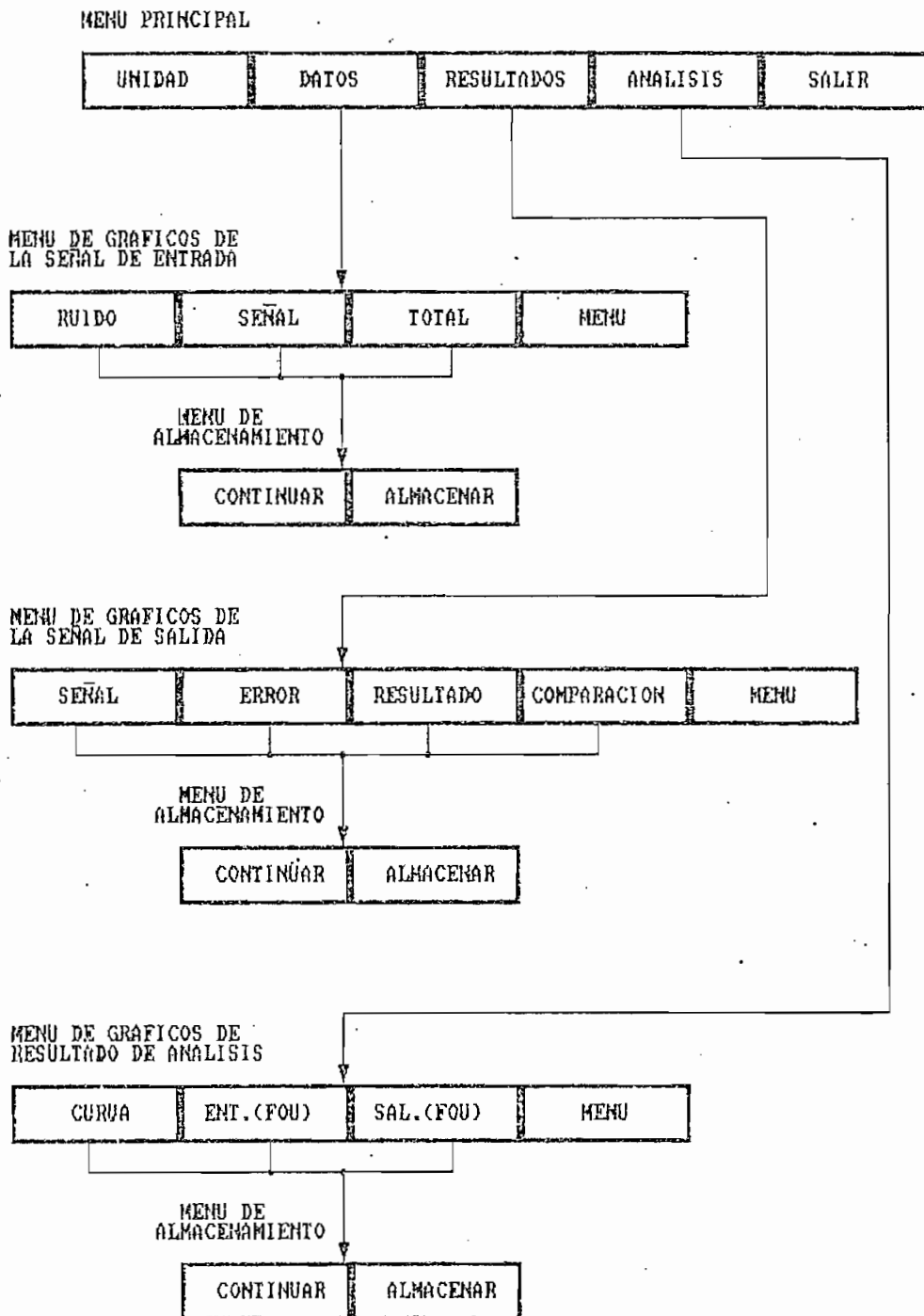


Fig. A.8 Menús de las rutinas para gráficos en el LOTUS 123.

ANEXO B

LISTADOS DE LOS PROGRAMAS

PROGRAMAS EN BASIC

B.1 PROGRAMA TESIS.BAS

```
10 '#####
20 '          ESCUELA POLITECNICA NACIONAL
30 '
40 '          FACULTAD DE INGENIERIA ELECTRICA
50 '
60 '          TESIS DE GRADO
70 '
80 '          "CANCELACION ADAPTIVA DE RUIDO"
90 '
100 '          AUTOR : MARIA SOLEDAD JIMENEZ JIMENEZ
110 '          DIRECTOR : Ing. GUALBERTO HIDALGO
120 '
130 '          MARZO -1988
140 '#####
150 '
160 '
170 '
180 'ZONA DE DEFINICION DE VARIABLES
190 '
200 DEFDBL L
210 COMMON UNI$,NOMS$,NOM$,NUM1,NUM,N,NM,X,LU,SIGMA,LAMB,ALB$,BAND1,BAND2
220 DIM A(2)
230 '
240 ' ZONA DEL ROTULO Y MENU
250 '
260 KEY OFF:IF BAND1=1 THEN 320 ELSE CLS:BAND1=1
270 LOCATE 5,13:PRINT "ESCUELA POLITECNICA NACIONAL"CHR$(13)CHR$(13)TAB(9)"FACULTAD DE INGENIERIA ELECTRI
C A"CHR$(13)CHR$(13)TAB(33)"TESIS DE GRADO"CHR$(13)CHR$(13)TAB(24)CHR$(34);
280 PRINT "CANCELACION ADAPTIVA DE RUIDO"CHR$(34)CHR$(13)CHR$(13)CHR$(13)TAB(21)
"AUTOR : MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(13)TAB(23)"DIRECTOR : Ing. G
UALBERTO HIDALGO";
290 PRINT CHR$(13)CHR$(13)TAB(34)"MARZO DE 1988"CHR$(13)CHR$(13)CHR$(13)CHR$(13)
CHR$(13)TAB(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
300 IF INKEY$="" THEN 300
310 '
320 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(
13)TAB(33)"MENU PRINCIPAL"
330 LOCATE 9,20:PRINT"UNIDAD DEL DISCO DE DATOS..... 1"CHR$(13)CHR$(13)
TAB(20)"GENERACION DE LOS DATOS DE ENTRADA..... 2"CHR$(13)CHR$(13)TAB(20)"SIMULA
CION DE FILTROS ADAPTIVOS..... 3"CHR$(13)CHR$(13)TAB(20);
340 PRINT "ANALISIS DEL DESEMPEÑO DEL FILTRO..... 4"CHR$(13)CHR$(13)TAB(20)"GRA
FIZACION DE RESULTADOS..... 5"CHR$(13)CHR$(13)TAB(20)"TERMINAR EL PROGR
AMA..... 6"CHR$(13)CHR$(13)CHR$(13)TAB(30)"ESCOJA UNA OPCION..."
350 A$=INKEY$:IF A$="" THEN 350
360 IF A$="1" THEN 420 ELSE IF A$="2" THEN 540 ELSE IF A$="3" THEN 780
```

```
370 IF A$="4" THEN 1350 ELSE IF A$="5" THEN 1760 ELSE IF A$="6" THEN END
380 BEEP:GOTO 350
390 '
400 '#####
410 '
420 '          RUTINA DE INGRESO DE LA UNIDAD DEL DISCO DE DATOS
430 '
440 '#####
450 '
460 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(
13)TAB(28)"UNIDAD DEL DISCO DE DATOS"
470 LOCATE 14,20:PRINT"INGRESE LA UNIDAD DEL DISCO DE DATOS..(A,B,C)"
480 LOCATE 17,35:INPUT"UNIDAD = ",UNI$:IF INSTR("ABCabc",UNI$)<>0 THEN 490 ELSE
BEEP:LOCATE 17,43:PRINT STRING$(30," "):GOTO 480
490 NOM$="":ALB$="":NUM=0:NUM1=0:N=0:X=0
500 GOTO 320
510 '
520 '#####
530 '
540 '          EJECUCION DE LA RUTINA DE GENERACION DE DATOS
550 '
560 '#####
570 '
580 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(
13)TAB(30)"GENERACION DE DATOS"
590 IF(UNI$<>"") THEN 620
600 LOCATE 12,17:PRINT"NO SE HA DEFINIDO LA UNIDAD DEL DISCO DE DATOS"CHR$(13)CH
R$(13)TAB(21)"PRESIONE CUALQUIER TECLA PARA CONTINUAR":BEEP
610 IF INKEY$="" THEN 610 ELSE 320
620 LOCATE 12,16:PRINT"INGRESE EL NOMBRE DEL ARCHIVO (max. 8 caracteres)"
630 LOCATE 14,32:INPUT"NOMBRE = ",NOM$:IF(LEN(NOM$)<9)AND(LEN(NOM$)>0) THEN 640
ELSE BEEP:LOCATE 14,43:PRINT STRING$(30," "):GOTO 630
640 NOMS$="":NOMX$="":ON ERROR GOTO 700
650 OPEN UNI$+":"+NOM$+".DIN" FOR INPUT AS#1:INPUT#1,NUM1,FM,V1:CLOSE #1
660 LOCATE 16,23:PRINT"EL ARCHIVO " UNI$ ":" NOM$ ".DIN YA EXISTE"CHR$(13)CHR$(
13)TAB(20)"1.-RECUPERAR LOS DATOS DESDE EL ARCHIVO"CHR$(13)CHR$(13)TAB(20)"2.-DE
STRUIR EL CONTENIDO (REEMPLAZARLO)":BEEP
670 LOCATE 23,28:PRINT"ESCOJA UNA OPCION...1/2"
680 A$=INKEY$:IF A$="" THEN 680 ELSE IF A$="1" THEN 690 ELSE IF A$="2" THEN 710
ELSE BEEP:GOTO 680
690 ALB$="":GOTO 320
700 RESUME 720
710 FOR I= 16 TO 25:LOCATE I,1:PRINT STRING$(79," "):NEXT I
720 CLOSE:LOCATE 23,24:PRINT"#### ESPERE POR FAVOR ####"
730 ON ERROR GOTO 1930
740 CHAIN "DATGEN"
750 '

```

```
760 '#####
770 '
780 '                RUTINA DE CONFIGURACION DEL FILTRO
790 '
800 '#####
810 '
820 KEY 1,"":KEY (1) ON:ON KEY (1) GOSUB 1900
830 KEY 2,"":KEY (2) ON:ON KEY (2) GOSUB 1910
840 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)TAB(
28)"CONFIGURACION DEL FILTRO"
850 IF(NUM1>0)THEN 880 ELSE LOCATE 12,18:PRINT"EL ARCHIVO DE DATOS NO SE HA GENE
RADO TODAVIA":BEEP
860 LOCATE 16,22:PRINT"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
870 IF INKEY$="" THEN 870 ELSE KEY(1)OFF:KEY(2)OFF:GOTO 320
880 LOCATE 25,5:PRINT"F1,(ENTER)- REINICIAR EL INGRESO DE DATOS   F2,(ENTER)- ME
NU PRINCIPAL";
890 LOCATE 7,18:PRINT"NDMBRE DEL ARCHIVO DE DATOS:   " UNI*+":"*+NOM$+" .DIN"CHR$
(13)CHR$(13)TAB(20)*SELECCIONAR EL TIPO DE ALGORITMO ADAPTIVO"CHR$(13)CHR$(13)TA
B(30)*1.-LMS"TAB(44)*2.-RLS"CHR$(13)CHR$(13)TAB(27)*SELECCIONE EL TIPO...(1/2)";
900 A$=INKEY$:IF A$="" THEN 900 ELSE IF A$="1" THEN ALG$="LMS" ELSE IF A$="2" TH
EN ALG$="RLS" ELSE BEEP:GOTO 900
910 FOR I=9 TO 13:LOCATE I,1:PRINT STRING$(80," "):NEXT I
920 LOCATE 6,35:PRINT"FILTRO "ALG$
930 LOCATE 9,25:INPUT"ORDEN DEL FILTRO (1<N<25) = ",N
940 IF (N<1)OR(N>25)OR(FIX(N)<>N) THEN BEEP:LOCATE 9,52:PRINT"          ":GOTO
930
950 LOCATE 10,7:PRINT"LA SEÑAL DE REFERENCIA DEBE SER UNA VERSION RETARDADA DE L
A SEÑAL"CHR$(13)TAB(13)"PRINCIPAL.(RETARDO ENTRE UNO Y LA LONGITUD DEL FILTRO)"
960 LOCATE 12,25:INPUT"INGRESE EL RETARDO (1<X<N) = ",X
970 IF (X<1)OR(X>N)OR(FIX(X)<>X) THEN BEEP:LOCATE 12,53:PRINT"          ":GOTO
960
980 FOR I=10 TO 12:LOCATE I,1:PRINT STRING$(80," "):NEXT I
990 LOCATE 10,28:PRINT"NUMERO DE RETARDOS = " X
1000 LOCATE 12,14:IF ALG$="LMS" THEN 1070
1010 PRINT"   VALOR DEL FACTOR DE OLVIDO LAMBDA (0<L<=1) =":INPUT" ",LAMB
1020 LOCATE 12,1:PRINT STRING$(79," "):IF(LAMB<0)OR(LAMB>1) THEN BEEP:GOTO 1000
1030 LOCATE 12,24:PRINT"FACTOR DE OLVIDO LAMBDA = " CSNG(LAMB)
1040 LOCATE 13,13:PRINT"VALOR DE LA CONSTANTE DE INICIALIZACION (s>100) =":INPU
T" ",SIGMA
1050 LOCATE 13,1:PRINT STRING$(79," "):IF SIGMA<100 THEN BEEP:GOTO 1040
1060 LOCATE 13,23:PRINT"CONSTANTE DE INICIALIZACION = " SIGMA
1070 NUM=NUM1:LOCATE 14,20:PRINT"EL ARCHIVO DE DATOS TIENE"; NUM1;" MUESTRAS"CH
R$(13)CHR$(13)TAB(20)" DESEA PROCESAR UN NUMERO MENOR ?...S/N";
1080 A$=INKEY$:IF A$="" THEN 1080 ELSE IF(A$="N")OR(A$="n")THEN 1110 ELSE IF (A$
<>"S")AND(A$<>"s") THEN 1080
1090 LOCATE 16,1:PRINT STRING$(80," "):LOCATE 16,28:INPUT"NUMERO DE MUESTRAS = "
,NUM
1100 IF (NUM>1)AND(NUM<NUM1)AND(FIX(NUM)=NUM) THEN 1110 ELSE BEEP:GOTO 1090
1110 FOR I=14 TO 16:LOCATE I,1:PRINT STRING$(80," "):NEXT I
1120 LOCATE 14,25:PRINT"NUMERO DE MUESTRAS NUM = " NUM
1130 LOCATE 16,9:PRINT"PARA ANALISIS DEL COMPORTAMIENTO DEL FILTRO EN ESTADO EST
ABLE"CHR$(13)TAB(16)"SE PUEDEN ALMACENAR SOLO LAS ULTIMAS M MUESTRAS"
```

```

1140 LOCATE 18,18:INPUT"NUMERO DE MUESTRAS A ALMACENAR (M<NUM) = ",NM:NM=FIX(NM)
1150 IF(NM<1)OR(NM>NUM)OR(FTX(NM)<>NM) THEN BEEP:LOCATE 18,58:PRINT"          ":G
OTO 1140
1160 FOR I=16 TO 18:LOCATE I,1:PRINT STRING$(79," "):NEXT I
1170 LOCATE 15,20:PRINT"SE GUARDARAN LAS ULTIMAS " NM " MUESTRAS"
1180 LOCATE 21,20:PRINT"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
1190 IF INKEY$="" THEN 1190 ELSE KEY(1)OFF:KEY(2)OFF
1200 '
1210 '#####
1220 '
1230 '          RUTINA DE EJECUCION DE LOS ALGORITMOS
1240 '
1250 '#####
1260 '
1270 FOR I=5 TO 25:LOCATE I,1:PRINT STRING$(78," "):NEXT I
1280 LOCATE 23,24:PRINT"#### ESPERE POR FAVOR ####"
1290 ON ERROR GOTO 1930
1300 CHAIN "ALGORITM"
1310 '
1320 '
1330 '#####
1340 '
1350 '          RUTINAS DE ANALISIS DE COMPORTAMIENTO
1360 '
1370 '#####
1380 '
1390 FOR I=5 TO 25:LOCATE I,1:PRINT STRING$(78," "):NEXT I
1400 IF(UNI$<>"")AND(NOM$<>"")THEN 1430
1410 LOCATE 12,15:PRINT"NO SE HAN DEFINIDO LA UNIDAD DE DATOS NI EL ARCHIVO"CHR$(
(13)CHR$(13)TAB(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR":BEEP
1420 IF INKEY$="" THEN 1420 ELSE 320
1430 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR
$(13)TAB(27)"ANALISIS DE COMPORTAMIENTO"
1440 LOCATE 10,20:PRINT"CURVA DE APRENDIZAJE ..... 1"CHR$(13)CHR$(1
3)TAB(20)"ANALISIS ESPECTRAL (DATOS DE ENTRADA).. 2"CHR$(13)CHR$(13)TAB(20)"ANAL
ISIS ESPECTRAL (DATOS DE SALIDA)... 3"
1450 LOCATE 16,20:PRINT "VOLVER AL MENU ANTERIOR..... 4"CHR$(13)CHR$(
13)CHR$(13)TAB(30)"ESCOJA UNA OPCION..."
1460 A$=INKEY$:IF A$="" THEN 1460
1470 IF A$="1" THEN 1630 ELSE IF A$="2" THEN 1490 ELSE IF A$="3" THEN 1500
1480 IF A$="4" THEN 320 ELSE BEEP:GOTO 1460
1490 BAND2=0:GOTO 1510
1500 BAND2=1
1510 '
1520 '#####
1530 '
1540 '          ENLACE CON EL PROGRAMA DE ANALISIS DE FOURIER
1550 '
1560 LOCATE 23,24:PRINT"#### ESPERE POR FAVOR ####"
1570 ON ERROR GOTO 1930
1580 CHAIN "FOURIER"
1590 '

```



```
1600 '  
1610 '#####  
1620 '  
1630 '          ENLACE CON EL PROGRAMA DE CURVA DE APRENDIZAJE  
1640 '  
1650 IF {NUM1<>0}AND{ALG<>""}AND{N<>0}AND{X<>0}AND{NUM<>0}AND{NM<>0} THEN 1690.  
1660 FOR I=5 TO 25:LOCATE I,1:PRINT STRING$(79," ");NEXT I  
1670 LOCATE 12,22:PRINT*NO SE HA CONFIGURADO EL FILTRO TODAVIA*CHR$(13)CHR$(13)T  
AB(20)*PRESIONE CUALQUIER TECLA PARA CONTINUAR*:BEEP  
1680 IF INKEY$="" THEN 1680 ELSE 320  
1690 LOCATE 23,24:PRINT*#### ESPERE POR FAVOR ####"  
1700 ON ERROR GOTO 1930  
1710 CHAIN "CURVA"  
1720 '  
1730 '  
1740 '#####  
1750 '  
1760 '          Rutina de Graficos  
1770 '  
1780 '#####  
1790 '  
1800 ON ERROR GOTO 1930  
1810 SHELL "GRAFICO.BAT"  
1820 GOTO 320  
1830 '  
1840 '#####  
1850 '  
1860 '          Rutinas de Deteccion de Errores  
1870 '  
1880 '#####  
1890 ' INTERCEPCION DE LAS TECLAS F1 Y F2  
1900 RETURN 040  
1910 KEY(1)OFF:KEY(2)OFF:RETURN 320  
1920 '  
1930 'INTERCEPCION DE ERRORES PRODUCIDOS EN EL ENLACE DE LOS PROGRAMAS  
1940 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");NEXT I:KEY(1)OFF:KEY(2)OFF  
1950 BEEP:LOCATE 20,10:PRINT*HA OCURRIDO ALGUN PROBLEMA DURANTE EL ENLACE CON LA  
S RUTINAS*CHR$(13)TAB(13)*VERIFIQUE QUE EL DISCO DE PROGRAMAS SEA EL CORRECTO O"  
CHR$(13)TAB(15)*REINICIALICE EL SISTEMA...PRESIONE CUALQUIER TECLA"  
1960 LOCATE 24,9:PRINT*LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;  
1970 IF INKEY$="" THEN 1970 ELSE RESUME
```

B.2 PROGRAMA DATGEN.BAS

```
10 '#####
20 '
30 '          RUTINA DE GENERACION DE DATOS
40 '
50 '          MARIA SOLEDAD JIMENEZ JIMENEZ
60 '
70 '#####
80 '
90 ' DIMENSIONAMIENTO DE VARIABLES
100 '
110 DEFDBL L
120 COMMON UNI$,NOMS$,NOM$,NUM1,HUM,H,NM,X,LU,SIGMA,LAMB,ALG$,BAND1,BAND2
130 DIM LRUI(2030),LSEN(2030),LYOUT(2030),FS(10),A(10)
140 DIM NUMM(2),FREC(2,3),AMP(2,3),NUMSIN(2)
150 LPI=3.1415926536#:NUM1=0:NSENAL=1 ':RANDOMIZE TIMER
160 '
170 ' INGRESO DE DATOS
180 '
190 KEY OFF
200 '#####
210 '
220 '          INGRESO DE DATOS
230 '
240 '#####
250 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(13)TAB(25)"RUTINA DE GENERACION DE DATOS"
260 IF(UNI$<>"")AND(NOM$<>"")THEN 280 ELSE BEEP:LOCATE 12,16:PRINT"NO SE HAN DE
FINIDO LA UNIDAD DEL DISCO DE DATOS"CHR$(13)TAB(28)"NI EL NOMBRE DEL ARCHIVO"
270 FOR I=1 TO 5000:NEXT I:GOTO 3550
280 LOCATE 9,14:PRINT"ESTA RUTINA GENERA LOS DATOS NECESARIOS PARA COMPROBAR EL"
CHR$(13)TAB(14)"FUNCIONAMIENTO DEL FILTRO. LA SEÑAL QUE ENTREGA PUEDE SER"
290 PRINT TAB(14)"LA COMBINACION DE UNA O MAS SINUSOIDES Y RUIDO BLANCO O"CHR$(13)TAB(14)"COLOREADO"
300 LOCATE 14,14:PRINT"PUEDA INCLUIRSE EN UN SOLO ARCHIVO UNA PRIMERA PARTE CO
N"CHR$(13)TAB(14)"CIERTO TIPO DE SEÑAL Y A CONTINUACION CAMBIAR A OTRA SEÑAL"CHR$(13)TAB(14)"PARA COMPROBAR COMO SE REAJUSTA EL FILTRO"
310 LOCATE 21,20:PRINT"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
320 IF INKEY$="" THEN 320
330 KEY 1,"":ON KEY(1) GOSUB 820
340 KEY 2,"":ON KEY(2) GOSUB 3540
350 ON ERROR GOTO 3290
360 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(13)TAB(25)"RUTINA DE GENERACION DE DATOS"
```

```
370 LOCATE 25,8:PRINT"F1,(ENTER)- REINICIAR EL INGRESO DE DATOS F2,(ENTER)- ME
NU PRINCIPAL";:KEY(1)ON:KEY(2)ON
380 LOCATE 7,19:PRINT"NOMBRE DEL ARCHIVO DE DATOS: "UNI*+":"+NOM*+ ".DIN"
390 LOCATE 9,12:PRINT"INGRESE EL NUMERO DE SINUSOIDES DE SU SEÑAL QUE PUEDE SER"
CHR$(13)TAB(14)"ENTRE 0 Y 3 (0 IMPLICA QUE SOLO HAY RUIDO SIN SEÑAL)"
400 LOCATE 12,24:INPUT"NUMERO DE SINUSOIDES (0<N<3) = ",NSIN
410 IF(NSIN>3)OR(NSIN<0)OR(FIX(NSIN)<>NSIN)THEN BEEP:LOCATE 12,54:PRINT"
":GOTO 400
420 FOR I=9 TO 12:LOCATE I,1:PRINT STRING$(79," "):NEXT I
430 NUMSIN(NSEÑAL)=NSIN:IF NSIN=0 THEN FMAX=0:GOTO 510
440 LOCATE 10,18:PRINT"LA SEÑAL ESTA COMPUESTA POR " NSIN " SINUSOIDE(S)"CHR$(13)
)CHR$(13)TAB(15)"INGRESE LAS RESPECTIVAS AMPLITUDES Y FRECUENCIAS.":FMAX=0
450 FOR I=1 TO NSIN
460 LOCATE 12+I,14:PRINT"FRECUENCIA # " I " (Hz) =";:INPUT" ",FS(I):FREC(NSEÑAL,I)
)=FS(I)
470 LOCATE 12+I,46:PRINT"AMPLITUD # " I " (V) =";:INPUT" ",A(I):AMP(NSEÑAL,I)=A(I)
)
480 IF(FS(I)<=0)OR(A(I)<=0)THEN BEEP:LOCATE 12+I,1:PRINT STRING$(79," "):GOTO 46
0
490 IF FS(I)>FMAX THEN FMAX=FS(I)
500 NEXT I:IF FMAX>0 THEN 530
510 LOCATE 16,17:INPUT"FRECUENCIA DE MUESTREO >2*F(I)max. (Hz) = ",FM
520 IF(FM<2*FMAX)THEN BEEP:LOCATE 16,58:PRINT" ":GOTO 510
530 TMAX=(2000-NUM1)/FM
540 LOCATE 17,18:PRINT"TIEMPO DE MUESTREO <" TMAX " Seg. (Seg) =";:INPUT" ",TD
550 IF(TD>TMAX)THEN BEEP:LOCATE 17,1:PRINT STRING$(79," "):GOTO 540
560 NUM=FIX(TD*FM):LOCATE 18,19:PRINT"NUMERO DE MUESTRAS EN ESTA SECCION = " NUM
570 IF NUM<1 THEN BEEP:LOCATE 17,1:PRINT STRING$(78," "):GOTO 540
580 KEY(1)OFF:KEY(2)OFF:LOCATE 25,1:PRINT STRING$(78," "):NUM*(NSEÑAL)=NUM
590 LOCATE 23,18:PRINT "#### ESPERE, ESTOY GENERANDO LA SEÑAL ####"
600 '
610 '#####
620 '
630 ' GENERACION DE SINUSOIDES
640 '
650 '#####
660 IF NSIN>0 THEN 690
670 FOR I=1 TO NUM:LSEN(NUM1+I)=0:NEXT I
680 GOTO 760
690 FOR J=1 TO NSIN:FS(J)=FS(J)*2*LPI/FM:NEXT J
700 FOR I=1 TO NUM
710 LSEN(NUM1+I)=0
720 FOR J=1 TO NSIN
730 LSEN(NUM1+I)=LSEN(NUM1+I)+A(J)*SIN(FS(J)*I*(NUM1+I-1))
740 NEXT J
750 NEXT I
760 FOR I=19 TO 23:LOCATE I,1:PRINT STRING$(79," "):NEXT I
770 NUM1=NUM1+NUM:NSEÑAL=NSEÑAL+1:IF(NUM1>1950)OR(NSEÑAL>2) THEN 860
780 LOCATE 20,10:PRINT"EL NUMERO TOTAL DE MUESTRAS GENERADAS NO DEBE EXEDER DE 2
000"CHR$(13)CHR$(13)TAB(19)"DESEA AGREGAR MAS DATOS A SU ARCHIVO...S/N"
790 A$=INKEY$:IF A$="" THEN 790 ELSE IF(A$="S")OR(A$="s") THEN 360 ELSE IF (A$="
N")OR(A$="n") THEN 860 ELSE BEEP:GOTO 790
800 '

```

```
810 'ROUTINA DE DETECCION DE TECLAS DE ESCAPE F1 Y F2
820 FM=0: NSENAL=1: NUM=0: NUM1=0: FMAX=0: NUMSIN(1)=0: NUMSIN(2)=0: RETURN 360
830 '
840 '#####
850 '
860 '                RUTINA DE GENERACION DE RUIDO BLANCO
870 '
880 '#####
890 '
900 DN KEY(1)GOSUB 3260
910 ON ERROR GOTO 3350
920 CLS:LOCATE 3,12:PRINT CHR$(34)*C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O*CHR$(34)CHR$(13)TAB(25)*MARIA SOLEDAD JIMENEZ JIMENEZ*CHR$(13)CHR$(
(13)TAB(21)*ROUTINA DE GENERACION DE RUIDO BLANCO"
930 LOCATE 25,8:PRINT"F1,(ENTER)- REINICIAR EL INGRESO DE DATOS   F2,(ENTER)- KE
NU PRINCIPAL";:KEY(1)ON:KEY(2)ON
940 LOCATE 8,14:INPUT "INGRESE LA VARIANZA DEL RUIDO (0=NO HAY RUIDO) = ",V1
950 IF V1<0 THEN BEEP:LOCATE 8,62:PRINT"                ":GOTO 940
960 IF V1>0 THEN 990
970 FOR I=1 TO NUM1:LRUI(I)=0:NEXT I
980 FIL$="":GOTO 2930
990 '
1000 LOCATE 12,18:PRINT"TIPOS DE FILTRO PARA GENERAR RUIDO COLOREADO"
'1010 LOCATE 14,30:PRINT"0.-RUIDO SIN FILTRAR"
1020 LOCATE 15,30:PRINT"1.-FILTRO PASABAJOS"
1030 LOCATE 16,30:PRINT"2.-FILTRO PASAALTOS"
1040 LOCATE 17,30:PRINT"3.-FILTRO PASABANDA"
1050 LOCATE 19,25:PRINT"SELECCIONE EL TIPO DE FILTRO...";
1060 A$=INKEY$:IF A$="" THEN 1060 ELSE IF INSTR("0123",A$)=0 THEN BEEP:GOTO 1060
1070 FOR I=1 TO 23:LOCATE I,1:PRINT STRING$(79," "):NEXT I
1080 LOCATE 10,11
1090 IF A$="0" THEN 1250 ELSE IF A$="2" THEN 1140 ELSE IF A$="3" THEN 1180
1100 PRINT"ROUTINA DE SIMULACION DE UN FILTRO DE BUTTERWORTH PASABAJOS"
1110 LOCATE 14,15:INPUT"INGRESE LA FRECUENCIA DE CORTE (FC>0) (Hz) = ",FC
1120 IF FC<=0 THEN BEEP:LOCATE 14,60:PRINT"                ":GOTO 1110
1130 FIL$="PASABAJOS":GOTO 1230
1140 PRINT"ROUTINA DE SIMULACION DE UN FILTRO DE BUTTERWORTH PASAALTOS"
1150 LOCATE 14,15:INPUT"INGRESE LA FRECUENCIA DE CORTE (FC>0) (Hz) = ",FC
1160 IF FC<=0 THEN BEEP:LOCATE 14,60:PRINT"                ":GOTO 1150
1170 FIL$="PASAALTOS":GOTO 1230
1180 PRINT"ROUTINA DE SIMULACION DE UN FILTRO DE BUTTERWORTH PASABANDA":FIL$="PAS
ABANDA"
1190 LOCATE 14,15:INPUT"INGRESE LA FRECUENCIA INFERIOR (F1>0) (Hz) = ",F1
1200 IF F1<=0 THEN BEEP:LOCATE 14,60:PRINT"                ":GOTO 1190
1210 LOCATE 16,15:INPUT"INGRESE LA FRECUENCIA SUPERIOR (F2>F1) (Hz) = ",F2
1220 IF F2<=F1 THEN BEEP:LOCATE 16,60:PRINT"                ":GOTO 1210
1230 LOCATE 18,14:INPUT"INGRESE EL ORDEN DEL FILTRO (PAR ENTRE 2 Y 10)= ",NS
1240 IF (FIX(NS/2)<>(NS/2))OR(NS<2)OR(NS>10) THEN BEEP:LOCATE 18,61:PRINT"
":GOTO 1230
1250 NS=NS/2:LOCATE 23,18:PRINT "### ESPERE, ESTOY GENERANDO EL RUIDO ###"
1260 LOCATE 25,1:PRINT STRING$(78," "):KEY(1)OFF:KEY(2)OFF:ORDEN=NS*2
1270 D1=SOR(V1)
1280 FOR I=1 TO NUM1
```

```
1290 Z=0
1300 FOR J=1 TO 12
1310 Z=Z+RND:NEXT J
1320 LRUI(I)=(Z-6)*D1/2.
1330 NEXT I
1340 IF A$="0" THEN FIL$="":GOTO 2930
1350 IF A$="2" THEN 1870 ELSE IF A$="3" THEN 2360
1360 '
1370 'RUTINAS PARA GENERAR RUIDO COLOREADO MEDIANTE FILTROS DE BUTTERWORTH
1380 '
1390 ' RUTINA PARA DISEÑAR UN FILTRO DIGITAL DE BUTTERWORTH PASABAJOS
1400 '
1410 ' SE DAN COMO DATOS DE ENTRADA LA FRECUENCIA DE CORTE FC EN HZ.,
1420 ' LA FRECUENCIA DE MUESTREO FM EN HZ. Y EL ORDEN DEL FILTRO.
1430 '
1440 ' EL NUMERO DE SECCIONES A REALIZARSE SERA 1/2 DEL ORDEN DEL FILTRO.
1450 '
1460 ' LA RUTINA ENTREGA COMO RESULTADOS LA SEÑAL FILTRADA EN EL VECTOR
1470 ' LRUI(I).
1480 '
1490 ' EL FILTRO DIGITAL TIENE NS SECCIONES EN CASCADA.
1500 ' LA K-ESIMA SECCION TIENE UNA FUNCION DE TRANSFERENCIA DADA POR:
1510 '
1520 '
1530 '
1540 '
1550 '
1560 ' ENTONCES SI LRUI(I) Y LYOUT(I) SON LAS ENTRADAS Y SALIDAS
1570 ' RESPECTIVAMENTE DE LA K-ESIMA SECCION, TENDREMOS:
1580 '
1590 ' LYOUT(I) = A(K) * [LRUI(I) + 2*LRUI(I-1) + LRUI(I-2)]
1600 '           - B(K)*LYOUT(I-1) - C(K)*LYOUT(I-2)
1610 '
1620 '
1630 '
1640 ' PREPARACION DEL VECTOR DE ENTRADA DE DATOS
1650 '
1660 FOR J=1 TO NUM1:LRUI(NUM1-J+3)=LRUI(NUM1-J+1):NEXT J
1670 LRUI(1)=0:LRUI(2)=0:LYOUT(1)=0:LYOUT(2)=0
1680 LWCP=TAN(FC*LPI/FM)
1690 FOR K=1 TO NS
1700 LCS=COS((2*(K+NS)-1)*LPI/(4*NS))
1710 LX=1/(1+LWCP^2-(2*LWCP*LCS))
1720 LA=LWCP^2*LX:LB=2*(LWCP^2-1)*LX:LC=(1+LWCP^2+(2*LWCP*LCS))*LX
1730 '
1740 ' CALCULO DEL VECTOR DE SALIDA DE LA K-ESIMA SECCION
1750 '
1760 FOR J=3 TO NUM1+2
1770 LYOUT(J)=LA*(LRUI(J)+2*LRUI(J-1)+LRUI(J-2))-LB*LYOUT(J-1)-LC*LYOUT(J-2)
1780 NEXT J
1790 '
1800 ' ACTUALIZACION DEL VECTOR DE ENTRADA DE DATOS PARA LA NUEVA SECCION
1810 '

```

```

1820 FOR J=1 TO NUM1+2:LRUI(J)=LYOUT(J):NEXT J
1830 NEXT K
1840 FOR J=3 TO NUM1+2:LRUI(J-2)=LYOUT(J):NEXT J
1850 GOTO 2900
1860 '
1870 ' RUTINA PARA DISEÑAR UN FILTRO DIGITAL DE BUTTERWORTH PASAALTOS
1880 '
1890 ' SE DAN COMO DATOS DE ENTRADA LA FRECUENCIA DE CORTE FC EN HZ.,
1900 ' LA FRECUENCIA DE MUESTREO FM EN HZ. Y EL ORDEN DEL FILTRO.
1910 '
1920 ' EL NUMERO DE SECCIONES A REALIZARSE SERA 1/2 DEL ORDEN DEL FILTRO.
1930 '
1940 ' LA RUTINA ENTREGA COMO RESULTADOS LA SEÑAL FILTRADA EN EL VECTOR
1950 ' LRUI(I).
1960 '
1970 ' EL FILTRO DIGITAL TIENE NS SECCIONES EN CASCADA.
1980 ' LA K-ESIMA SECCION TIENE UNA FUNCION DE TRANSFERENCIA DADA POR:
1990 '
2000 '
2010 '
2020 '
2030 '
2040 ' ENTONCES SI LRUI(I) Y LYOUT(I) SON LAS ENTRADAS Y SALIDAS
2050 ' RESPECTIVAMENTE DE LA K-ESIMA SECCION, TENDREMOS:
2060 '
2070 '
2080 '
2090 '
2100 '
2110 '
2120 ' PREPARACION DEL VECTOR DE ENTRADA DE DATOS
2130 '
2140 FOR J=1 TO NUM1:LRUI(NUM1-J+3)=LRUI(NUM1-J+1):NEXT J
2150 LRUI(1)=0:LRUI(2)=0:LYOUT(1)=0:LYOUT(2)=0
2160 LWCP=TAN(FC*LPI/FM)
2170 FOR K=1 TO NS
2180 LCS=COS((2*(K+NS)-1)*LPI/(4*NS))
2190 LA=1/(1+LWCP^2-(2*LWCP*LCS))
2200 LB=2*(LWCP^2-1)*LA:LC=(1+LWCP^2+(2*LWCP*LCS))*LA
2210 '
2220 ' CALCULO DEL VECTOR DE SALIDA DE LA K-ESIMA SECCION
2230 '
2240 FOR J=3 TO NUM1+2
2250 LYOUT(J)=LA*(LRUI(J)-2*LRUI(J-1)+LRUI(J-2))-LB*LYOUT(J-1)-LC*LYOUT(J-2)
2260 NEXT J
2270 '
2280 ' ACTUALIZACION DEL VECTOR DE ENTRADA DE DATOS PARA LA NUEVA SECCION
2290 '
2300 FOR J=1 TO NUM1+2:LRUI(J)=LYOUT(J):NEXT J
2310 NEXT K
2320 FOR J=3 TO NUM1+2:LRUI(J-2)=LYOUT(J):NEXT J
2330 GOTO 2900
2340 '

```

$$H(Z) = \frac{A(K) * (Z^2 - 2*Z + 1)}{Z^2 + B(K)*Z + C(K)}$$

$$LYOUT(I) = A(K) * [LRUI(I) - 2*LRUI(I-1) + LRUI(I-2)] - B(K)*LYOUT(I-1) - C(K)*LYOUT(I-2)$$

```

2350 '
2360 ' RUTINA PARA DISEÑAR UN FILTRO DIGITAL DE BUTTERWORTH PASABANDA
2370 '
2380 ' SE DAN COMO DATOS LAS FRECUENCIAS DE LA BANDA DE PASO F1 y F2
2390 ' EN HZ, LA FRECUENCIA DE MUESTRO FM EN HZ. Y EL ORDEN DEL FILTRO
2400 '
2410 ' EL NUMERO DE SECCIONES A REALIZARSE SERA 1/2 DEL ORDEN DEL FILTRO.
2420 '
2430 ' LA RUTINA ENTREGA COMO RESULTADOS LA SEÑAL FILTRADA EN EL VECTOR
2440 ' LRUI(1).
2450 '
2460 ' EL FILTRO DIGITAL TIENE NS SECCIONES EN CASCADA.
2470 ' LA K-ESIMA SECCION TIENE UNA FUNCION DE TRANSFERENCIA DADA POR:
2480 '
2490 '
2500 '
2510 '
2520 '
2530 ' ENTONCES SI LRUI(1) Y LYOUT(1) SON LAS ENTRADAS Y SALIDAS
2540 ' RESPECTIVAMENTE DE LA K-ESIMA SECCION, TENDREMOS:
2550 '
2560 '
2570 '
2580 '
2590 '
2600 '
2610 '
2620 ' PREPARACION DEL VECTOR DE ENTRADA DE DATOS
2630 '
2640 FOR J=1 TO NUM1:LRUI(NUM1-J+5)=LRUI(NUM1-J+1):NEXT J
2650 LRUI(1)=0:LRUI(2)=0:LRUI(3)=0:LRUI(4)=0
2660 LYOUT(1)=0:LYOUT(2)=0:LYOUT(3)=0:LYOUT(4)=0
2670 LW1=TAN(F1*LPI/FM):LW2=TAN(F2*LPI/FM)
2680 LWC=LW2-LW1
2690 LQ=LWC^2+(2*LW1*LW2)
2700 LS=LW1^2*LW2^2
2710 FOR K=1 TO NS
2720 LCS=COS((2*(K+NS)-1)*LPI/(4*NS))
2730 LP=-2*LWC*LCS
2740 LR=LP*LW1*LW2
2750 LX=1+LP+LQ+LR+LS
2760 LA=LWC^2/LX
2770 LB=(-4-2*LP+2*LR+4*LS)/LX
2780 LC=(6-2*LQ+6*LS)/LX
2790 LD=(-4+2*LP-2*LR+4*LS)/LX
2800 LE=(1-LP+LQ-LR+LS)/LX
2810 FOR J=5 TO NUM1+4
2820 LYOUT(J)=LA*(LRUI(J)-2*LRUI(J-2)+LRUI(J-4))-LB*LYOUT(J-1)-LC*LYOUT(J-2)-LD*
LYOUT(J-3)-LE*LYOUT(J-4)
2830 NEXT J
2840 '
2850 'ACTUALIZACION DEL VECTOR DE ENTRADA DE DATOS PARA LA NUEVA SECCION
2860 '

```

```
2870 FOR J=1 TO NUM1+4:LRUI(J)=LYOUT(J):NEXT J
2880 NEXT K
2890 FOR J=5 TO NUM1+4:LRUI(J-4)=LYOUT(J):NEXT J
2900 '
2910 '#####
2920 '
2930 '          Rutina de Almacenamiento de Datos en el Archivo de Salida
2940 '
2950 '#####
2960 '
2970 ON ERROR GOTO 3410
2980 LOCATE 25,1:PRINT STRING$(79," ");
2990 LOCATE 23,15:PRINT"#### ESPERE, ESTOY ALMACENANDO EN EL ARCHIVO ####"
3000 CLOSE:KEY(1)OFF:KEY(2)OFF:LOCATE 25,1:PRINT STRING$(79," ");
3010 OPEN UNI$+" "+NOM$+" ".DIN" FOR OUTPUT AS #1
3020 PRINT#1,NUM1,FM,V1
3030 FOR I=1 TO NUM1:PRINT #1,CSNG(LRUI(I)),CSNG(LSEN(I)):NEXT I
3040 CLOSE
3050 '
3060 '#####
3070 '
3080 '          Rutina de Presentacion de Resultados
3090 '
3100 '#####
3110 '
3120 CLS:LOCATE 1,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E R U I D O"CHR$(34)CHR$(13)TAB(26)"ARCHIVO DE DATOS "UNI$+" "+NOM$+" ".DIN"CHR
$(13)CHR$(13)TAB(15)"#### CARACTERISTICAS DE LA SEÑAL PERIODICA ####"
3130 CLOSE:OPEN "SCRN:" FOR OUTPUT AS#3
3140 GOSUB 3790
3150 LOCATE 25,25:PRINT"DESEA UN REPORTE IMPRESO...S/N";:DEEP
3160 A$=INKEY$:IF A$="" THEN 3160 ELSE IF(A$="n")OR(A$="N")THEN 3180 ELSE IF (A$
<)"s")AND(A$<)"S")THEN BEEP:GOTO 3160
3170 GOSUB 3600
3180 PRINT#3,CHR$(12);CLOSE:GOTO 3550
3190 '
3200 '#####
3210 '
3220 '          Rutinas de Intercepcion de Errores
3230 '
3240 '#####
3250 '
3260 'Rutina de Intercepcion de Tecla F1 (Ruido Blanco)
3270 RETURN 920
3280 '
3290 'Errores de la Rutina de Generacion de la Señal
3300 FOR I=22 TO 25:LOCATE 1,1:PRINT STRING$(79," ");:NEXT I
3310 BEEP:LOCATE 22,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE LA GENERACION DE LA
SEÑAL"CHR$(13)TAB(11)"REVISE SUS DATOS...PRESIONE CUALQUIER TECLA PARA CONTINUA
R";
3320 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
3330 IF INKEY$="" THEN 3330 ELSE RESUME 360
3340 '

```


B.3 PROGRAMA ALGORITHM.BAS

```
10 '#####
20 '
30 '          Rutina de Ejecucion de los Algoritmos Adaptivos
40 '
50 '          MARIA SOLEDAD JIMENEZ JIMENEZ
60 '
70 '#####
80 '
90 '
100 'Rutina de Ejecucion de los Algoritmos
110 '
120 'Dimensionamiento de las Variables
130 '
140 DEFDBL L
150 COMMON UNI$,NOMS$,NOM$,NUM1,NUM,N,NM,X,LU,SIGMA,LAMB,ALG$,BAND1,BAND2
160 DIM LPO(25,25),LHEST(25),LK(25),LAUX1(25)
170 DIM LSENRUI(2030),LYOUT(2030),LER(2030)
180 '
190 KEY OFF:IF(UNI$<>"")AND(NOM$<>"")THEN 230
200 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(
13)TAB(23)"Rutina de Ejecucion del Algoritmo"
210 LOCATE 12,16:PRINT"NO SE HAN DEFINIDO LA UNIDAD DEL DISCO DE DATOS"CHR$(13)T
AB(28)"NI EL NOMBRE DEL ARCHIVO"CHR$(13)CHR$(13)TAB(26)"#### ESPERE POR FAVOR &
####"
220 FOR I=1 TO 5000:NEXT I:GOTO 1980
230 KEY 1,"":ON KEY(1) GOSUB 2140
240 KEY 2,"":ON KEY(2) GOSUB 2160
250 ON ERROR GOTO 2190
260 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(
13)TAB(23)"Rutina de Ejecucion del Algoritmo"
270 LOCATE 25,8:PRINT"F1,(ENTER)- REINICIAR EL INGRESO DE DATOS  F2,(ENTER)- HE
RU PRINCIPAL*";KEY(1)ON:KEY(2)ON
280 LOCATE 7,19:PRINT"Nombre del Archivo de Datos: "UNI$+";"+NOM$+".DIN"
290 '
300 'Definicion del Archivo de Salida de los Datos
310 '
320 LOCATE 12,17:PRINT"EL ARCHIVO DE SALIDA SE LLAMARA : "UNI$+";"+NOM$+".DOT"CH
R$(13)CHR$(13)TAB(29)"DESEA CAMBIARLO...S/N"
330 A$=INKEY$:IF A$="" THEN 330 ELSE IF(A$="n")OR(A$="N") THEN 380 ELSE IF (A$<>
"s")AND(A$<>"S") THEN BEEP:GOTO 330
340 LOCATE 14,1:PRINT STRING$(79," ")
350 LOCATE 14,11:PRINT"NUEVO NOMBRE DEL ARCHIVO (max. 8 caracteres sin extensión
)"
```

```

360 LOCATE 16,31:INPUT "NOMBRE : ",NOMS$:IF LEN(NOMS$)>8 THEN BEEP:LOCATE 16,40:
PRINT " " " ":GOTO 360
370 GOTO 390
380 NOMS$=NOMS$
390 FOR I=12 TO 25:LOCATE I,1:PRINT STRING$(79," "):NEXT I
400 LOCATE 12,17:PRINT"EL ARCHIVO DE SALIDA SE LLAMA : "UNI$+" "+NOMS$+".DOT"
410 LOCATE 23,18:PRINT"##### ESPERE, ESTOY LEYENDO LOS DATOS #####"
420 '
430 'INICIALIZACION
440 '
450 ON ERROR GOTO 2250
460 FOR I=1 TO N:LHEST(I)=0:LSENROI(I)=0
470 FOR J=1 TO N:IF I=J THEN LPO(I,J)=SIGMA ELSE LPO(I,J)=0
480 NEXT J:NEXT I
490 FOR I=N+1 TO N+X-1:LSENROI(I)=0:NEXT I
500 '
510 'INGRESO DE LOS DATOS DESDE EL ARCHIVO
520 CLOSE
530 OPEN UNI$+" "+NOMS$+".DIN" FOR INPUT AS#1
540 INPUT#1,NUM1,FM,V1
550 FIN=NUM+N+X-1
560 FOR I=N+X TO FIN
570 INPUT #1,LROI,LESEN:LSENROI(I)=LROI+LESEN
580 NEXT I
590 CLOSE#1
600 '
610 'DETERMINACION DEL ALGORITMO ADAPTIVO
620 '
630 ON ERROR GOTO 2310
640 IF ALG$="LMS" THEN 680 ELSE IF ALG$="RLS" THEN 1150
650 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," "):NEXT I
660 BEEP:LOCATE 20,14:PRINT"ERROR .. NO SE HA DETERMINADO EL ALGORITMO A USARSE"
CHR$(13)CHR$(13)TAB(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
670 IF INKEY$="" THEN 670 ELSE GOTO 1980
680 '
690 '#####
700 '
710 '          A L G O R I T M O   A D A P T I V O   L M S
720 '
730 '#####
740 '
750 ' INICIO DEL ALGORITHM DE CALCULO
760 '
770 LAUX1=0
780 FOR I=N+X TO FIN
790 LAUX1=LAUX1+LSENROI(I)^2:NEXT I
800 LAUX1=LAUX1/NUM:LTRAZA=N*LAUX1
810 LU=1/(3*LTRAZA) 'LU ES EL PARAMETRO DE CONVERGENCIA
820 LOCATE 23,1:PRINT STRING$(79," "):
830 LOCATE 15,16:PRINT "PARAMETRO DE CONVERGENCIA CRITICO Uc. = ";CSNG(LU);" " C
HR$(13)CHR$(13)TAB(26)"DESEA USAR OTRO VALOR ... S/N ";:UMAX=CSNG(LU)
840 A$=INKEY$:IF A$="" THEN 840 ELSE IF(A$="S")OR(A$="s") THEN 850 ELSE IF (A$="
n")OR(A$="N")THEN 870 ELSE BEEP: GOTO 840

```

```
850 FOR I=16 TO 18:LOCATE I,1:PRINT STRING$(79, " ");:NEXT I
860 LOCATE 16,32:INPUT"VALOR DE U = ",LU:IF LU<0 THEN BEEP:LOCATE 16,44:PRINT"
      ":GOTO 860
870 FOR I=16 TO 18:LOCATE I,1:PRINT STRING$(79, " ");:NEXT I
880 LOCATE 17,22:PRINT "PARAMETRO DE CONVERGENCIA U =";CSNG(LU)
890 LOCATE 23,18:PRINT"#### ESTOY EJECUTANDO EL ALGORITMO LMS ####"
900 '
910 TINICIO=TIMER
920 FOR IX=1 TO NUM
930 '
940 ' CALCULO DE LA SALIDA DEL FILTRO
950 '
960 LYOUT(IX)=0
970 FOR I=1 TO N
980 LYOUT(IX)=LYOUT(IX)+LSENRUI(IX+I-1)*LHEST(N-I+1)
990 NEXT I
1000 '
1010 ' CALCULO DEL ERROR
1020 '
1030 LER(IX)=LSENRUI(IX+N+X-1)-LYOUT(IX)
1040 '
1050 ' ACTUALIZACION DE LOS COEFICIENTES DEL FILTRO
1060 '
1070 FOR I=1 TO N
1080 LHEST(I)=LHEST(I)+2*LU*LER(IX)*LSENRUI(IX+N-I)
1090 NEXT I
1100 NEXT IX
1110 '
1120 'FIN DEL ALGORITMO LMS
1130 '
1140 GOTO 1750
1150 '
1160 '#####
1170 '
1180 '          A L G O R I T M O   A D A P T I V O   R L S
1190 '
1200 '#####
1210 '
1220 ' INICIO DEL ALGORITMO DE CALCULO
1230 '
1240 LOCATE 23,18:PRINT"#### ESTOY EJECUTANDO EL ALGORITMO RLS ####"
1250 TINICIO=TIMER
1260 FOR IX=1 TO NUM
1270 '
1280 ' CALCULO DE LA SALIDA DEL FILTRO
1290 '
1300 LYOUT(IX)=0
1310 FOR I=1 TO N
1320 LYOUT(IX)=LYOUT(IX)+LSENRUI(IX+I-1)*LHEST(N-I+1)
1330 NEXT I
1340 '
1350 ' CALCULO DEL ERROR
1360 '
```

```
1370 LER(IX)=LSENRUI(IX+N+X-1)-LYOUT(IX).
1380 '
1390 ' CALCULO DE LA GANANCIA DE KALMAN
1400 '
1410 FOR I=1 TO N:LAUX1(I)=0
1420 FOR J=1 TO N
1430 LAUX1(I)=LAUX1(I)+LPO(I,J)*LSENRUI(IX+N-J)
1440 NEXT J:NEXT I
1450 LAUX2=0
1460 FOR I=1 TO N
1470 LAUX2=LAUX2+LSENRUI(IX+N-I)*LAUX1(I)
1480 NEXT I
1490 LAUX2=1/(LAUX2+LAMB)
1500 FOR I=1 TO N
1510 LK(I)=LAUX1(I)*LAUX2
1520 NEXT I
1530 '
1540 ' CALCULO DE LA MATRIZ DE AUTOCORRELACION
1550 '
1560 FOR I=1 TO N:LAUX1(I)=0
1570 FOR J=1 TO N
1580 LAUX1(I)=LAUX1(I)+LSENRUI(IX+N-J)*LPO(J,I)
1590 NEXT J:NEXT I
1600 FOR I=1 TO N
1610 FOR J=1 TO N
1620 LPO(I,J)=(LPO(I,J)-LK(I)*LAUX1(J))/LAMB
1630 NEXT J
1640 NEXT I
1650 '
1660 ' CALCULO DE LOS COEFICIENTES DEL FILTRO
1670 '
1680 FOR I=1 TO N
1690 LHST(I)=LHST(I)+LK(I)*LER(IX) 'VECTOR DE COEF. ACTUALIZADOS
1700 NEXT I
1710 NEXT IX
1720 '
1730 'FIN DEL ALGORITMO RLS
1740 '
1750 '
1760 '#####
1770 '
1780 '          RUTINA DE ALMACENAMIENTO DE RESULTADOS
1790 '
1800 ON ERROR GOTO 2370
1810 LOCATE 23,18:PRINT"#### ESTOY ALMACENANDO EN EL ARCHIVO ####"
1820 CLOSE
1830 OPEN UNI#+" "+NOMS#+".DOT" FOR OUTPUT AS #2
1840 PRINT#2,NH,FM,V1
1850 FOR I=NUM-NK TO NUM
1860 PRINT #2, CSNG(LSENRUI(I+N+X-1)),CSNG(LER(I)),CSNG(LYOUT(I))
1870 NEXT I
1880 CLOSE:TFINAL=TIMER
1890 '

```

```
1900 '#####
1910 '
1920 '          RUTINA DE PRESENTACION DE RESULTADOS
1930 '
1940 CLS:LOCATE 1,12:PRINT CHR$(34)"C A N C E L A C I O N  A D A P T I V A  D
E  R U I D O"CHR$(34)CHR$(13)CHR$(13)TAB(20)"RUTINA DE SIMULACION DE FILTROS AD
APTIVOS"CHR$(13)CHR$(13)
1950 OPEN "SCRN:" FOR OUTPUT AS #3
1960 GOSUB 2660
1970 LOCATE 25,25:PRINT"DESEA UN REPORTE IMPRESO...S/N";:BEEP
1980 A$=INKEY$:IF A$="" THEN 1980 ELSE IF(A$="n")OR(A$="N")THEN 2040 ELSE IF(A$<
>"s")AND(A$<>"S")THEN BEEP:GOTO 1980
1990 GOSUB 2480
2000 PRINT#3,CHR$(12):CLOSE
2010 '
2020 '#####
2030 '
2040 '          RUTINA DE ENLACE CON EL PROGRAMA MAESTRO
2050 '
2060 ON ERROR GOTO 2430
2070 CLS:LOCATE 23,25:PRINT"#### ESPERE POR FAVOR ####"
2080 CHAIN "TESIS"
2090 '
2100 '#####
2110 '
2120 '          RUTINA DE INTERCEPCION DE ERRORES
2130 '
2140 'INTERCEPCION DE LA TECLA F1
2150 RETURN 260
2160 'INTERCEPCION DE LA TECLA F2
2170 RETURN 1980
2180 '
2190 'INTERCEPCION DE ERRORES EN EL INGRESO DE DATOS
2200 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
2210 BEEP:LOCATE 20,13:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL INGRESO DE DATOS
,"CHR$(13)TAB(11)"REVISE SUS DATOS...PRESIONE CUALQUIER TECLA PARA CONTINUAR";
2220 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
2230 IF INKEY$="" THEN 2230 ELSE ON ERROR GOTO 2190:RESUME 230
2240 '
2250 'INTERCEPCION DE ERRORES EN EL ACCESO AL ARCHIVO DE DATOS
2260 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
2270 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ACCESO AL DISCO,
REVISE"CHR$(13)TAB(17)"LA COLOCACION DEL MISHO D QUE SEA EL CORRECTO"CHR$(13)TAB
(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
2280 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
2290 IF INKEY$="" THEN 2290 ELSE ON ERROR GOTO 2190:RESUME 230
2300 '
2310 'INTERCEPCION DE ERRORES DURANTE EL PROCESAMIENTO ADAPTIVO
2320 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
2330 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL PROCESAMIENTO,"C
HR$(13)TAB(11)"REVISE SUS DATOS...PRESIONE CUALQUIER TECLA PARA CONTINUAR";
2340 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
2350 IF INKEY$="" THEN 2350 ELSE ON ERROR GOTO 2190:RESUME 230
```

```
2360 '
2370 'INTERCEPCION DE ERRORES EN EL ALMACENAMIENTO DE RESULTADOS
2380 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
2390 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ACCESO AL DISCO,
REVISE"CHR$(13)TAB(17)"LA COLOCACION DEL MISMO O QUE SEA EL CORRECTO"CHR$(13)TAB
(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
2400 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
2410 IF INKEY$="" THEN 2410 ELSE RESUME 1780
2420 '
2430 'INTERCEPCION DE ERRORES DURANTE EL ENLACE CON EL PROGRAMA MAESTRO
2440 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
2450 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ENLACE CON EL PR
OGRAMA"CHR$(13)TAB(14)"MAESTRO, VERIFIQUE QUE EL DISCO DE PROGRAMAS SEA EL"CHR$(
13)TAB(9)"CORRECTO, O REINICIE EL SISTEMA...PRESIONE CUALQUIER TECLA"
2460 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
2470 IF INKEY$="" THEN 2470 ELSE RESUME 2040
2480 '#####
2490 '
2500 '                                RUTINA DE IMPRESION DE RESULTADOS
2510 '
2520 '#####
2530 ON ERROR GOTO 2540:GOTO 2570
2540 FOR I=24 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
2550 BEEP:LOCATE 24,23:PRINT"LA IMPRESORA NO SE ENCUENTRA LISTA":LOCATE 25,21:PR
INT"DESEA CONTINUAR CON LA IMPRESION...S/N";
2560 A$=INKEY$:IF A$="" THEN 2560 ELSE IF(A$="S")OR(A$="s") THEN RESUME ELSE IF(
A$="n")OR(A$="N")THEN STOP ELSE BEEP:GOTO 2560
2570 FOR I=24 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
2580 LOCATE 25,10:PRINT"ALISTE LA IMPRESORA...PRESIONE CUALQUIER TECLA PARA CONT
INUAR";
2590 IF INKEY$="" THEN 2590
2600 CLOSE:OPEN "LPT1:" FOR OUTPUT AS #3
2610 '
2620 'ROTULO
2630 '
2640 PRINT#3,TAB(13)"E S C U E L A   P O L I T E C N I C A   N A C I O N A L"CHR
$(13)CHR$(13)TAB(9)"F A C U L T A D   D E   I N G E N I E R I A   E L E C T R I
C A"CHR$(13)CHR$(13)TAB(16)"TESIS DE GRADO : "CHR$(34);
2650 PRINT#3,"CANCELACION ADAPTIVA DE RUIDO"CHR$(34)CHR$(13)CHR$(13)TAB(26)"MARI
A SOLEDAD JIMENEZ JIMENEZ"CHR$(13)TAB(34)"MARZO DE 1988"CHR$(13)CHR$(13)CHR$(13)
CHR$(13)TAB(22)"RUTINA DE EJECUCION DE LOS ALGORITMOS"CHR$(13)CHR$(13)
2660 PRINT#3,TAB(20)"ARCHIVO DE DATOS DE ENTRADA : "UNI$+":"+NOM$+"DIN"CHR$(13)
TAB(22)"ARCHIVO DE RESULTADOS : "UNI$+":"+NOM$+"DOT"CHR$(13)CHR$(13)CHR$(13)TA
B(23)"### CONFIGURACION DEL FILTRO ###"CHR$(13)
2670 PRINT#3,TAB(28)"ALGORITMO ADAPTIVO : "ALG$ CHR$(13)CHR$(13)TAB(30)"ORDEN DE
L FILTRO : "N CHR$(13)TAB(14)"NUMERO DE RETARDOS EN LA ENTRADA DE REFERENCIA : "
X CHR$(13)
2680 IF ALG$="RLS" THEN 2710
2690 PRINT#3,TAB(17)"PARAMETRO DE CONVERGENCIA CRITICO Uc. = " UMAX CHR$(13)TAB(
20)"PARAMETRO DE CONVERGENCIA USADO U = "CSNG(LU)
2700 GOTO 2720
2710 PRINT#3,TAB(18)"VALOR DEL FACTOR DE OLVIDO Lambda = " CSNG(LAMB) CHR$(13)TA
B(14)"VALOR DE LA CONSTANTE DE INICIALIZACION Sigaa = "SIGMA
```



```
2720 PRINT#3,CHR$(13)TAB(25)*SE HAN PROCESADO "NUM " MUESTRAS*CHR$(13)TAB(12)*EN
    EL ARCHIVO SE HAN ALMACENADO LAS ULTIMAS "NM " MUESTRAS"
2730 TPROC=TFINAL-TINICIO:MINUTOS=FIX(TPROC/60):SEGUNDOS=FIX(TPROC-MINUTOS*60)
2740 PRINT#3,CHR$(13)TAB(17)*"## TIEMPO DE PROCESAMIENTO : "MINUTOS" min. "SEGUND
    OS" seg. ##"
2750 RETURN
```

B.4 PROGRAMA CURVA.BAS

```

10 '#####
20 '
30 '          Rutina para DETERMINACION DE LA CURVA DE APRENDIZAJE
40 '
50 '          MARIA SOLEDAD JIMENEZ JIMENEZ
60 '
70 '#####
80 '
90 ' DIMENSIONAMIENTO DE VARIABLES
100 '
110 DEFDBL L
120 COMMON UNI$,NOMS$,NOM$,NUM1,NUM,N,NM,X,LU,SIGMA,LAMB,ALG$,BAND1,BAND2
130 DIM LSENUI(2030),LECM(2030),LRXX(25,25),LPX(25),LA(25,25),R(25)
140 DIM LHOPT(25),LHEST(25),LAUX1(25),LAUX2(25),LK(25)
150 E=.000001
160 '
170 KEY OFF:IF(UNI$<>"")AND(NOM$<>"")THEN 210
180 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(
13)TAB(14)"RUTINA DE DETERMINACION DE LA CURVA DE APRENDIZAJE"
190 BEEP:LOCATE 12,16:PRINT"NO SE HAN DEFINIDO LA UNIDAD DEL DISCO DE DATOS"CHR$(
13)TAB(28)"NI EL NOMBRE DEL ARCHIVO"CHR$(13)CHR$(13)TAB(26)"#### ESPERE POR FA
VOR ####"
200 GOTO 240
210 IF((ALG$="LMS")OR(ALG$="RLS"))AND(NUM<>0)AND(NUM1<>0)AND(N<>0)AND(X<>0) THEN
250
220 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
230 BEEP:LOCATE 20,14:PRINT"ERROR .. NO SE HA CONFIGURADO EL FILTRO A USARSE"
CHR$(13)CHR$(13)TAB(26)"#### ESPERE POR FAVOR ####"
240 FOR I=1 TO 5000:NEXT I:GOTO 3130
250 KEY 1,"":ON KEY(1) GOSUB 3220
260 KEY 2,"":ON KEY(2) GOSUB 3240
270 ON ERROR GOTO 3270
280 CLS:LOCATE 3,12:PRINT CHR$(34)"C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O"CHR$(34)CHR$(13)TAB(25)"MARIA SOLEDAD JIMENEZ JIMENEZ"CHR$(13)CHR$(
13)TAB(14)"RUTINA DE DETERMINACION DE LA CURVA DE APRENDIZAJE"
290 LOCATE 25,8:PRINT"F1,(ENTER)- REINICIAR EL INGRESO DE DATOS  F2,(ENTER)- ME
NU PRINCIPAL";:KEY(1)OR:KEY(2)ON
300 LOCATE 7,19:PRINT"NOMBRE DEL ARCHIVO DE DATOS: "UNI$+";"NOMS$+"DIN"
310 '
320 'DEFINICION DEL ARCHIVO DE SALIDA
330 '
340 LOCATE 12,16:PRINT"EL ARCHIVO DE SALIDA SE LLAMARA : "UNI$+";"NOMS$+"CUR"CH
R$(13)CHR$(13)TAB(29)"DESEA CAMBIARLO...S/N"
350 A$=INKEY$:IF A$="" THEN 350 ELSE IF(A$="n")OR(A$="N")THEN 400 ELSE IF(A$<>"s
")AND(A$<>"S") THEN BEEP:GOTO 350

```

```

360 LOCATE 14,1:PRINT STRING$(79," ");
370 LOCATE 14,11:PRINT"NUEVO NOMBRE DEL ARCHIVO (max. 8 caracteres sin extension
)"
380 LOCATE 16,31:INPUT"NOMBRE : ",NOMX$:IF(LEN(NOMX$)>8)OR(LEN(NOMX$)<1)THEN BEE
P:LOCATE 16,40:PRINT"                               ":GOTO 380
390 GOTO 410
400 NOMX$=NOM$
410 FOR I=12 TO 25:LOCATE I,1:PRINT STRING$(79," ");NEXT I
420 LOCATE 12,17:PRINT"EL ARCHIVO DE SALIDA SE LLAMA : "UNI$+"."+NOMX$+".CUR"
430 LOCATE 23,18:PRINT"##### ESPERE, ESTOY LEYENDO LOS DATOS #####"
440 '
450 'INICIALIZACION
460 '
470 ON ERROR GOTO 3330
480 FOR I=1 TO N:LHST(I)=0:LSENROI(I)=0:NEXT I
490 FOR I=N+1 TO N+X-1:LSENROI(I)=0:NEXT I
500 '
510 'INGRESO DE LOS DATOS DESDE EL ARCHIVO
520 '
530 CLOSE
540 OPEN UNI$+"."+NOMX$+".DIH" FOR INPUT AS#1
550 INPUT#1,NUM1,FH,V1
560 FIN=NUM+N+X-1
570 FOR I=N+X TO FIN
580 INPUT #1,LRUI,LSEN:LSENROI(I)=LRUI+LSEN
590 NEXT I
600 CLOSE#1
610 '
620 ON ERROR GOTO 3390
630 '
640 ' DETERMINACION DE LA MATRIZ DE AUTOCORRELACION Y CORR. CRUZADA
650 ' CALCULO DE UNA FILA DE LA MATRIZ DE AUTOCORRELACION.
660 ' CALCULO DEL VECTOR DE CORRELACION CRUZADA.
670 '
680 LOCATE 23,18:PRINT"##### CALC. MATRIZ DE AUTOCORR. Y CORR. #####"
690 FOR IT=1 TO N
700 LRXX(1,IT)=0:LPX(IT)=0
710 '
720 ' CALCULO DE UN ELEMENTO DE LA MATRIZ DE AUTOCORRELACION
730 ' CALCULO DE UN ELEMENTO DEL VECTOR DE CORRELACION CRUZADA.
740 '
750 FOR I=N+X TO FIN-IT+1
760 LRXX(1,IT)=LRXX(1,IT)+LSENROI(I)*LSENROI(I+IT-1)
770 LPX(IT)=LPX(IT)+LSENROI(I)*LSENROI(I-X-IT+1)
780 NEXT I
790 LRXX(1,IT)=LRXX(1,IT)/(NUM-IT+1)
800 LPX(IT)=LPX(IT)/(NUM-IT+1)
810 NEXT IT
820 '
830 ' CONFORMACION DE LA MATRIZ DE AUTOCORRELACION
840 '
850 FOR I=2 TO N:FOR J=1 TO N
860 LRXX(I,J)=LRXX(1,ABS(I-J)+1)

```

```
870 NEXT J:NEXT I
880 '
890 ' CALCULO DE LA INVERSA DE LA MATRIZ DE AUTOCORRELACION
900 ' USANDO EL ALGORITMO DE SHIPLEY
910 ' LA MATRIZ INVERSA QUEDA EN LA MISMA LOCALIDAD
920 '
930 ' CONSERVACION DE LA MATRIZ DE AUTOCORRELACION EN RX(X(I,J)
940 '
950 FOR I=1 TO N
960 FOR J=1 TO N
970 LA(I,J)=LRXX(I,J)
980 NEXT J:NEXT I
990 '
1000 ' ALGORITMO DE INVERSION
1010 '
1020 FOR I=1 TO N
1030 R(I)=0:NEXT I
1040 FOR K=1 TO N
1050 IF ABS(LA(K,K))<=E THEN 1090
1060 GOSUB 1260
1070 NEXT K
1080 GOTO 1110
1090 R(K)=K
1100 GOTO 1070
1110 FOR I=1 TO N
1120 IF R(I)=0 THEN 1160:K=1
1130 IF ABS(LA(I,I))<=E THEN 1160:R(I)=0
1140 GOSUB 1260
1150 GOTO 1110
1160 NEXT I
1170 FOR J=1 TO N
1180 IF R(J)=0 THEN 1200
1190 GOTO 1220
1200 NEXT J
1210 GOTO 1250
1220 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");NEXT I:KEY(1)OFF:KEY(2)OFF
1230 BEEP:LOCATE 20,12:PRINT"ERROR...LA MATRIZ DE AUTOCORRELACION NO TIENE "CHR$(
(13)TAB(25)"INVERSA .....REVISE SUS DATOS"CHR$(13)TAB(20)"PRESIONE CUALQUIER TE
CLA PARA CONTINUAR";
1240 IF INKEY$="" THEN 1240 ELSE 3130
1250 GOTO 1410
1260 LA(K,K)=1/LA(K,K)
1270 FOR J=1 TO N:IF J=K THEN 1290
1280 LA(K,J)=-LA(K,K)*LA(K,J)
1290 NEXT J
1300 FOR I=1 TO N:IF I=K THEN 1340
1310 FOR J=1 TO N:IF J=K THEN 1330
1320 LA(I,J)=LA(I,J)+LA(K,J)*LA(I,K)
1330 NEXT J
1340 NEXT I
1350 FOR I=1 TO N
1360 IF I=K THEN 1380
1370 LA(I,K)=LA(K,K)*LA(I,K)
```

```
1380 NEXT I
1390 RETURN
1400 '
1410 ' CALCULO DEL VECTOR DE COEFICIENTES DEL FILTRO OPTIMO
1420 '
1430 ' EL VECTOR DE COEFICIENTES OPTIMO SERA :HOPT=INV(LRXX(I,J)*LPX]
1440 '
1450 LOCATE 23,18:PRINT"#### CALC. VECTOR DE COEFIC. OPTIMOS ####"
1460 FOR I=1 TO N
1470 LHOPT(I)=0
1480 FOR J=1 TO N
1490 LHOPT(I)=LHOPT(I)+LA(I,J)*LPX(J)
1500 NEXT J:NEXT I
1510 '
1520 ' DETERMINACION DE LA ECUACION PARA LA CURVA DE APRENDIZAJE
1530 '      E = EMIN + (HEST'-HOPT')*R*(HEST-HOPT)
1540 ' CALCULO DEL ERROR CUADRATICO MEDIO MINIMO :EMIN=DPRI-PX(I)*HOPT(I)
1550 '
1560 ' CALCULO DE DPRI
1570 '
1580 LDPRI=0
1590 FOR I=N+X TO FIN:LDPRI=LDPRI+LSENROI(I)*LSENROI(I):NEXT I
1600 LDPRI=LDPRI/NUM
1610 '
1620 ' CALCULO DE PX*HOPT
1630 '
1640 LAUXX=0
1650 FOR I=1 TO N:LAUXX=LAUXX+LPX(I)*LHOPT(I):NEXT I
1660 '
1670 ' CALCULO DE EMIN
1680 '
1690 LEMIN=LDPRI-LAUXX
1700 '
1710 ' Rutina de calculo de los coeficientes del filtro
1720 '
1730 ' INICIO DEL ALGORITMO DE CALCULO
1740 '
1750 LOCATE 23,18:PRINT"#### CALCULANDO LOS PUNTOS DE LA CURVA ####"
1760 '
1770 ' DETERMINACION DEL TIPO DE ALGORITMO A USARSE
1780 '
1790 ON ERROR GOTO 3450
1800 IF ALG$="RLS" THEN 2160
1810 '
1820 '#####
1830 '
1840 '                ALGORITMO LHS
1850 '
1860 '#####
1870 '
1880 FOR IX=1 TO NUM
1890 '
1900 ' CALCULO DE LA SALIDA DEL FILTRO
```

```
1910 '  
1920 LYOUT=0  
1930 FOR I=1 TO N  
1940 LYOUT=LYOUT+LSENUI(IX+I-1)*LHEST(N-I+1)  
1950 NEXT I  
1960 '  
1970 ' CALCULO DEL ERROR  
1980 '  
1990 LER=LSENUI(IX+N+X-1)-LYOUT  
2000 '  
2010 ' ACTUALIZACION DE LOS COEFICIENTES DEL FILTRO  
2020 '  
2030 FOR I=1 TO N  
2040 LHEST(I)=LHEST(I)+2*LU*LER*LSENUI(IX+N-I)  
2050 NEXT I  
2060 '  
2070 'FIN DEL CALCULO DE LOS COEFICIENTES  
2080 '  
2090 'CALCULO DEL PUNTO DE LA CURVA DE APRENDIZAJE  
2100 '  
2110 GOSUB 2760  
2120 NEXT IX  
2130 GOTO 2990  
2140 '#####  
2150 '  
2160 '                                ALGORITMO RLS  
2170 '  
2180 '#####  
2190 '  
2200 FOR I=1 TO N:FOR J=1 TO N  
2210 IF I=J THEN LA(I,J)=SIGMA ELSE LA(I,J)=0  
2220 NEXT J:NEXT I  
2230 FOR IX=1 TO NUM  
2240 '  
2250 ' CALCULO DE LA SALIDA DEL FILTRO  
2260 '  
2270 LYOUT=0  
2280 FOR I=1 TO N  
2290 LYOUT=LYOUT+LSENUI(IX+I-1)*LHEST(N-I+1)  
2300 NEXT I  
2310 '  
2320 ' CALCULO DEL ERROR  
2330 '  
2340 LER=LSENUI(IX+N+X-1)-LYOUT  
2350 '  
2360 ' CALCULO DE LA GANANCIA DE KALMAN  
2370 '  
2380 FOR I=1 TO N:LAUX1(I)=0  
2390 FOR J=1 TO N  
2400 LAUX1(I)=LAUX1(I)+LA(I,J)*LSENUI(IX+N-J)  
2410 NEXT J:NEXT I  
2420 LAUX4=0  
2430 FOR I=1 TO N
```

```
2440 LAUX4=LAUX4+LSENROI(IX+N-1)*LAUX1(I)
2450 NEXT I
2460 LAUX4=1/(LAUX4+LAMB)
2470 FOR I=1 TO N
2480 LK(I)=LAUX1(I)*LAUX4
2490 NEXT I
2500 '
2510 ' CALCULO DE LA MATRIZ DE AUTOCORRELACION
2520 '
2530 FOR I=1 TO N:LAUX1(I)=0
2540 FOR J=1 TO N
2550 LAUX1(I)=LAUX1(I)+LSENROI(IX+N-J)*LA(J,I)
2560 NEXT J:NEXT I
2570 FOR I=1 TO N
2580 FOR J=1 TO N
2590 LA(I,J)=(LA(I,J)-LK(I)*LAUX1(J))/LAMB
2600 NEXT J
2610 NEXT I
2620 '
2630 ' CALCULO DE LOS COEFICIENTES DEL FILTRO
2640 '
2650 FOR I=1 TO N
2660 LHEST(I)=LHEST(I)+LK(I)*LER 'VECTOR DE COEF. ACTUALIZADOS
2670 NEXT I
2680 '
2690 'FIN DEL CALCULO DE LOS COEFICIENTES
2700 '
2710 'CALCULO DEL PUNTO DE LA CURVA DE APRENDIZAJE
2720 '
2730 GOSUB 2760
2740 NEXT IX
2750 GOTO 2990
2760 '#####
2770 '
2780 '          SUBROUTINA DE CALCULO DEL PUNTO EN LA CURVA DE APRENDIZAJE
2790 '
2800 '#####
2810 '
2820 FOR I= 1 TO N
2830 LAUX1(I)=LHEST(I)-LHOPT(I)
2840 NEXT I
2850 FOR J=1 TO N
2860 LAUX2(J)=0
2870 FOR I=1 TO N
2880 LAUX2(J)=LAUX2(J)+LRXX(I,J)*LAUX1(I)
2890 NEXT I:NEXT J
2900 LAUX3=0
2910 FOR I=1 TO N
2920 LAUX3=LAUX3+LAUX1(I)*LAUX2(I)
2930 NEXT I
2940 LECM(IX)=LENIN+LAUX3
2950 RETURN
2960 '

```

```
2970 '#####
2980 '
2990 '          ALMACENAMIENTO DE LOS RESULTADOS EN UN ARCHIVO
3000 '
3010 '#####
3020 '
3030 CLOSE:ON ERROR GOTO 3510
3040 LOCATE 23,18:PRINT"#### ALMACENANDO LOS RESULTADOS ####"
3050 OPEN UNI$+" "+NOMX$+".CUR" FOR OUTPUT AS # 2
3060 FOR I=1 TO NUM
3070 PRINT #2,CSNG(LECM(I))
3080 NEXT I
3090 CLOSE
3100 '
3110 '#####
3120 '
3130 '          RUTINA DE ENLACE CON EL PROGRAMA NUESTRO
3140 '
3150 ON ERROR GOTO 3570
3160 CHAIN "TESIS"
3170 '
3180 '#####
3190 '
3200 '          RUTINA DE INTERCEPCION DE ERRORES
3210 '
3220 'INTERCEPCION DE LA TECLA F1
3230 RETURN 280
3240 'INTERCEPCION DE LA TECLA F2
3250 RETURN 3130
3260 '
3270 'INTERCEPCION DE ERRORES EN EL INGRESO DE DATOS
3280 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
3290 BEEP:LOCATE 20,13:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL INGRESO DE DATOS
,CHR$(13)TAB(11)"REVISE SUS DATOS...PRESIONE CUALQUIER TECLA PARA CONTINUAR";
3300 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
3310 IF INKEY$="" THEN 3310 ELSE RESUME 250
3320 '
3330 'INTERCEPCION DE ERRORES EN EL ACCESO AL ARCHIVO DE DATOS
3340 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
3350 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ACCESO AL DISCO,
REVISE"CHR$(13)TAB(17)"LA COLOCACION DEL MISMO O QUE SEA EL CORRECTO"CHR$(13)TAB
(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
3360 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
3370 IF INKEY$="" THEN 3370 ELSE RESUME 470
3380 '
3390 'INTERCEPCION DE ERRORES DURANTE EL CALCULO DE LAS CORRELACIONES
3400 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
3410 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL CALCULO DE LAS C
HR$(13)TAB(25)"CORRELACIONES,REVISE SUS DATOS"CHR$(13)TAB(20)"PRESIONE CUALQUIER
TECLA PARA CONTINUAR";
3420 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
3430 IF INKEY$="" THEN 3430 ELSE RESUME 3130
3440 '

```



```
3450 'INTERCEPCION DE ERRORES DURANTE EL PROCESAMIENTO ADAPTIVO
3460 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
3470 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL PROCESAMIENTO,"C
HR$(13)TAB(11)"REVISE SUS DATOS...PRESIONE CUALQUIER TECLA PARA CONTINUAR";
3480 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
3490 IF INKEY$="" THEN 3490 ELSE RESUME 3130
3500 '
3510 'INTERCEPCION DE ERRORES EN EL ALMACENAMIENTO DE RESULTADOS
3520 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
3530 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ACCESO AL DISCO,
REVISE"CHR$(13)TAB(17)"LA COLOCACION DEL MISMO O QUE SEA EL CORRECTO"CHR$(13)TAB
(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
3540 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
3550 IF INKEY$="" THEN 3550 ELSE RESUME 2990
3560 '
3570 'INTERCEPCION DE ERRORES DURANTE EL ENLACE CON EL PROGRAMA MAESTRO
3580 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I:KEY(1)OFF:KEY(2)OFF
3590 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ENLACE CON EL PR
OGRAMA"CHR$(13)TAB(14)"MAESTRO, VERIFIQUE QUE EL DISCO DE PROGRAMAS SEA EL"CHR$(
13)TAB(9)"CORRECTO, O REINICIALICE EL SISTEMA...PRESIONE CUALQUIER TECLA"
3600 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
3610 IF INKEY$="" THEN 3610 ELSE RESUME 3130
```

B.5 PROGRAMA FOURIER.BAS

```

10 '#####
20 '
30 '          RUTINA PARA DETERMINACION DE LA TRANSFORMADA DE FOURIER
40 '
50 '          MARIA SOLEDAD JIMENEZ JIMENEZ
60 '
70 '#####
80 '
90 'DIMENSIONAMIENTO DE LAS VARIABLES
100 '
110 DEFDBL L
120 COMMON UNI$,NOMS$,NOM$,NUM1,NUM,N,NM,X,LU,SIGMA,LAMB,ALG$,BAND1,BAND2
130 DIM Q(2100),R(2100),S(2100),Y(2100),T(2),U(2),W(2)
140 PI=3.1415926#
150 '
160 KEY OFF
170 CLS:LOCATE 3,12:PRINT CHR$(34)*C A N C E L A C I O N   A D A P T I V A   D
E   R U I D O*CHR$(34)CHR$(13)TAB(25)*MARIA SOLEDAD JIMENEZ JIMENEZ*CHR$(13)CHR$
(13)TAB(25)*RUTINA DE ANALISIS DE FOURIER"
180 IF(UNI$<>"")AND(NOM$<>"")THEN 200 ELSE BEEP:LOCATE 12,16:PRINT"NO SE HAN DE
FINIDO LA UNIDAD DEL DISCO DE DATOS"CHR$(13)TAB(28)*NI EL NOMBRE DEL ARCHIVO*CHR
$(13)CHR$(13)TAB(26)*#### ESPERE POR FAVOR ####"
190 FOR I=1 TO 5000:NEXT I:GOTO 1310
200 IF BAND2=0 THEN EXT1$=".DIN" : EXT2$=".FIN":EXT3$="ENTRADA":GOTO 220
210 EXT1$=".DOT":EXT2$=".FOU":EXT3$="SALIDA"
220 LOCATE 7,13:PRINT"TRANSFORMADA DE FOURIER DEL ARCHIVO DE DATOS DE ";EXT3$
230 LOCATE 9,19:PRINT"NOMBRE DEL ARCHIVO DE DATOS: "UNI$+":"+NOMS$+EXT1$
240 IF NOMS$="" THEN NOMS$=NOM$
250 LOCATE 12,18:PRINT"EL ARCHIVO DE SALIDA SE LLAMA : "UNI$+":"+NOMS$+EXT2$
260 ON ERROR GOTO 1420
270 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
280 LOCATE 23,19:PRINT"#### ESPERE, ESTOY LEYENDO LOS DATOS ####"
290 '
300 IF BAND2=0 THEN 320 ELSE 410
310 '
320 'INGRESO DE DATOS DESDE UN ARCHIVO .DIN
330 '
340 CLOSE
350 OPEN UNI$+":"+NOMS$+".DIN" FOR INPUT AS #1
360 INPUT#1,NUM1,FM,V1
370 FOR I=1 TO NUM1
380 INPUT #1,LRUI,LSEN:Y(I)=LRUI+LSEN:NEXT I
390 NUMP=NUM1:GOTO 480
400 '
410 'INGRESO DE DATOS DESDE UN ARCHIVO .DOT
420 '

```

```
430 CLOSE
440 OPEN "B:"+NOMS$+".DOT" FOR INPUT AS #1
450 INPUT#1,NUMP,FM,V1
460 FOR I=1 TO NUMP
470 INPUT #1,LSENROI,LER,Y(I):NEXT I
480 CLOSE #1
490 LOCATE 23,10:PRINT"#### ESPERE, ESTOY CALCULANDO ####"
500 '#####
510 '
520 '          CALCULO DE LA TRANSFORMADA DE FOURIER
530 '
540 '#####
550 '
560 ' CALCULO DEL LOG. EN BASE 2 DE NUM
570 '
580 ON ERROR GOTO 1510
590 M=1:L1=0
600 M=M*2:L1=L1+1
610 IF M<NUMP THEN 600
620 IF M=NUMP THEN 670
630 M=M/2:L1=L1-1
640 FOR I=1 TO M
650 Y(I)=Y(NUMP-M+I)
660 NEXT I
670 NUMP=M:D=L1+1
680 FOR I=1 TO NUMP
690 R(I)=Y(I):Q(I)=Y(I)
700 NEXT I
710 '
720 ' TRANSFORMADA DISCRETA DE FOURIER
730 '
740 N2=NUMP/2:N1=NUMP-1:J=1
750 FOR I=1 TO N1
760 IF I >= J THEN 780
770 T(1)=R(J):T(2)=S(J):R(J)=R(I):S(J)=S(I):R(I)=T(1):S(I)=T(2)
780 K=N2
790 IF K >= J THEN 810
800 J=J-K:K=K/2:GOTO 790
810 J=J+K:NEXT I
820 FOR L=1 TO L1
830 L2=2^L:L3=L2/2:U(1)=1:U(2)=0
840 W(1)=COS(PI/L3):W(2)=-SIN(PI/L3)
850 FOR J=1 TO L3
860 FOR I=J TO NUMP STEP L2
870 G=I+L3
880 T(1)=R(G)*U(1)-S(G)*U(2)
890 T(2)=S(G)*U(1)+R(G)*U(2)
900 R(G)=R(I)-T(1):S(G)=S(I)-T(2)
910 R(I)=R(I)+T(1):S(I)=S(I)+T(2)
920 NEXT I
930 A1=U(1)*W(1)-U(2)*W(2)
940 A2=U(1)*W(2)+U(2)*W(1)
950 U(1)=A1:U(2)=A2
```

```
960 NEXT J
970 NEXT L
980 '
990 ' CALCULO DE LA MAGNITUD DEL ESPECTRO DE FRECUENCIA
1000 '
1010 FOR I=1 TO NUMP
1020 Y(I)=SQR(R(I)^2+S(I)^2)
1030 NEXT I
1040 '
1050 ' SUBROUTINA PARA CALCULO DE LA MAGNITUD MAX. Y MIN. DEL ESPECTRO
1060 '
1070 Y1=Y(I);Y2=Y(1)
1080 FOR I=2 TO NUMP
1090 IF Y(I) < Y1 THEN Y1=Y(I) ' EN Y1 SE GUARDA LA MAGNITUD MIN.
1100 IF Y(I) > Y2 THEN Y2=Y(I) ' EN Y2 SE GUARDA LA MAGNITUD MAX.
1110 NEXT I
1120 FOR J=1 TO NUMP
1130 Y(J)=Y(J)/Y2;Y(J)=20*LOG(Y(J))/LOG(10)
1140 NEXT J
1150 '
1160 '
1170 '#####
1180 '
1190 '           ALMACENAMIENTO EN ARCHIVO
1200 '
1210 LOCATE 23,13:PRINT"#### ESPERE,ESTOY ALMACENANDO LOS RESULTADOS ####"
1220 CLOSE:ON ERROR GOTO 1570
1230 OPEN UNI+*"+NOMS+EXT2$ FOR OUTPUT AS #2
1240 PRINT#2,NUMP/2,FM,V1
1250 FOR I=1 TO NUMP/2
1260 PRINT #2,Y(I):NEXT I
1270 CLOSE
1280 '
1290 '#####
1300 '
1310 '           RUTINA DE ENLACE CON EL PROGRAMA MAESTRO
1320 '
1330 '
1340 ON ERROR GOTO 1630
1350 CHAIN "TESIS"
1360 '
1370 '
1380 '#####
1390 '
1400 '           RUTINA DE INTERCEPCION DE ERRORES
1410 '
1420 'INTERCEPCION DE ERRORES EN EL ACCESO AL ARCHIVO DE DATOS
1430 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
1440 IF ERR=53 THEN 1480
1450 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ACCESO AL DISCO,
REVISE*CHR$(13)TAB(17)"LA COLOCACION DEL MISMO O QUE SEA EL CORRECTO*CHR$(13)TAB
(20)*PRESIONE CUALQUIER TECLA PARA CONTINUAR"
1460 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERR;SPC(15);"CODIGO DE ERROR = ";ERR;
```

```
1470 IF INKEY$="" THEN 1470 ELSE RESUME 260
1480 BEEP:LOCATE 20,21:PRINT"NO EXISTE EL ARCHIVO DE DATOS INDICADO"CHR$(13)CHR$(13)TAB(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
1490 IF INKEY$="" THEN 1490 ELSE RESUME 1340
1500 '
1510 'INTERCEPCION DE ERRORES DURANTE EL PROCESAMIENTO
1520 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
1530 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL PROCESAMIENTO,"CHR$(13)TAB(11)"REVISE SUS DATOS...PRESIONE CUALQUIER TECLA PARA CONTINUAR";
1540 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
1550 IF INKEY$="" THEN 1550 ELSE RESUME 1340
1560 '
1570 'INTERCEPCION DE ERRORES EN EL ALMACENAMIENTO DE RESULTADOS
1580 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
1590 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ACCESO AL DISCO, REVISE"CHR$(13)TAB(18)"LA COLOCACION DEL MISMO O QUE NO ESTE LLENO"CHR$(13)TAB(20)"PRESIONE CUALQUIER TECLA PARA CONTINUAR"
1600 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
1610 IF INKEY$="" THEN 1610 ELSE RESUME 1220
1620 '
1630 'INTERCEPCION DE ERRORES DURANTE EL ENLACE CON EL PROGRAMA MAESTRO
1640 FOR I=20 TO 25:LOCATE I,1:PRINT STRING$(79," ");:NEXT I
1650 BEEP:LOCATE 20,11:PRINT"HA OCURRIDO ALGUN ERROR DURANTE EL ENLACE CON EL PROGRAMA"CHR$(13)TAB(14)"MAESTRO, VERIFIQUE QUE EL DISCO DE PROGRAMAS SEA EL"CHR$(13)TAB(9)"CORRECTO, O REINICIALICE EL SISTEMA...PRESIONE CUALQUIER TECLA"
1660 LOCATE 24,9:PRINT"LINEA DEL ERROR = ";ERL;SPC(15);"CODIGO DE ERROR = ";ERR;
1670 IF INKEY$="" THEN 1670 ELSE RESUME 1340
```

LISTADOS DE LOS PROGRAMAS

PROGRAMAS DE ENLACE

ARCHIVOS POR LOTES

B.6 ARCHIVO FILTRO.BAT

```
ECHO OFF
CLS
KEY-FAKE ""
TESIS
CLS
```

B.7 ARCHIVO MONO.BAT

```
ECHO OFF
CLS
COPY 123MONO.SET 123.SET
ECHO EL PROGRAMA ESTA DEFINIDO PARA UN MONITOR MONOCROMATICO
```

B.8 ARCHIVO COLOR.BAT

```
ECHO OFF
CLS
COPY 123COLOR.SET 123.SET
ECHO EL PROGRAMA ESTA DEFINIDO PARA UN MONITOR A COLOR
```


LISTADOS DE LOS PROGRAMAS

P R O G R A M A S D E G R A F I C O S

ARCHIVOS PARA TRABAJO EN LOTUS123

GRAFICO.WK1

TESIS DE GRADO
"CANCELACION ADAPTIVA DE RUIDO"

RUTINA DE GRAFIZACION DE RESULTADOS

AUTOR : MARIA SOLEDAD JIMENEZ JIMENEZ
DIRECTOR : Ing. GUALBERTO HIDALGO

EPN MARZO 1988

TABLA DE UBICACION DE MACROS

\A	I6..I12
\MENU	Q3..U26
\MENU1A	Q90..T108
\MENU1D	Q31..T38
\MENU1R	Q60..U68
\MENU2A	Q111..R118
\MENU2D	Q45..R48
\MENU2R	Q76..R78

MACRO INICIAL DE ARRANQUE
{HOME}
{GOTO}I21~{WINDOWSOFF}/FDB:~
/GRGTXOFGLQ
TFCANCELACION ADAPTIVA DE RUIDO~
TXNUMERO DE MUESTRAS~TYAMPLITUD~
TSGRAFICOS~LANADA~LBNADA~QQ
{MENUBRANCH \MENU}

MENU PRINCIPAL (MENU)

UNIDAD
Unidad del disco de datos
/FD(?)~
(BRANCH I12)

DATOS

Grafico de la senal de entrada
(HOME)
/RE.(CEND)<COUHD>(RIGHT)>(RIGHT)>
(RIGHT)>(RIGHT)>(RIGHT)>(RIGHT)>
/FIN*.DIN"(?)~
CGOTO)02~
+CLEFT>(LEFT)+CLEFT)~
/C".(LEFT)>(END)<COUHD>(RIGHT)>~
CGOTO)02~/DFD2~
/DF.(LEFT)>(LEFT)>(CEND)>
(COUHD)>(RIGHT)>(RIGHT)>~
CGOTO)02~
/GR:ABCDEFQX.(CEND)<COUHD>~Q
CGOTO)121~
(MENUBRANCH \MENU10)

RESULTADOS

Grafico de la senal de salida
(HOME)
/RE.(CEND)<COUHD>(RIGHT)>(RIGHT)>
(RIGHT)>(RIGHT)>(RIGHT)>(RIGHT)>
/FIN*.00T"(?)~
CGOTO)02~/0FD2~
/OF.(LEFT)>(LEFT)>(CEND)>
(COUHD)>(RIGHT)>(RIGHT)>~
CGOTO)G1~MIN<(HOME)<COUHD>.
(CEND)<COUHD>~(COUHD)
@MAX<(HOME)>(RIGHT)>(RIGHT)>
(COUHD).(CEND)<COUHD>~(CUP)
/RV.(CEND)<COUHD>~
/C"<COUHD>.(LEFT)>(CEND)<COUHD>
(RIGHT)>~(CGOTO)F2~
+C2-1.5*(CABS)~
/C"<COUHD>.(LEFT)>(CEND)<COUHD>
(RIGHT)>~(HOME)
CGOTO)02~
/GR:ABCDEFQX.(CEND)<COUHD>~Q
CGOTO)121~
(MENUBRANCH \MENU10)

ANALISIS

Curva de aprendizaje , etc.
(HOME)
/RE.(CEND)<COUHD>(RIGHT)>(RIGHT)>
(RIGHT)>(RIGHT)>(RIGHT)>(RIGHT)>
(MENUBRANCH \MENU10)

SALIR

Terminar la rutina
(HOME)
CGOTO)121~/0C(?)~

SUBMENU DE GRAFICOS DE LOS DATOS (MENU1D)

RUIDO
 Ruido blanco o coloreado /GRABQAB2"A.(CEND)COOIND"
 OTSCESC)RUIDO"LADESC)
 RUIDO BLANCO"QVQ
 (GOTO)I21"CHINDOHSOND
 CHINDOHSOFF)
 (MENUBRANCH \MENU2D)

SEÑAL
 Señal pura /GRABQAB2"A.(CEND)COOIND"
 OTSCESC)SEÑAL SIN RUIDO"
 LADESC)SEÑAL PURA"QVQ
 (GOTO)I21"CHINDOHSOND
 CHINDOHSOFF)
 (MENUBRANCH \MENU2D)

SUBMENU DE ALMACENAMIENTO DE GRAFICOS (MENU2D)

CONTINUAR
 Continuar el programa /GSC?)"RADESC)CESC)
 (MENUBRANCH \MENU1D)

ALMACENAR
 Almacenar el grafico /GSC?)"RADESC)CESC)
 (MENUBRANCH \MENU1D)

SUBMENU DE GRAFICOS DE LOS RESULTADOS (MENU1R)

SEÑAL
 Entrada al filtro /GRABQAB2"A.(CEND)COOIND"
 OTSCESC)ENTRADA AL FILTRO"
 LADESC)ENTRADA"QVQ
 (GOTO)I21"CHINDOHSOND
 CHINDOHSOFF)
 (MENUBRANCH \MENU2R)

ERROR
 Señal de error /GRABQAB2"A.(CEND)COOIND"
 OTSCESC)SEÑAL DE ERROR"
 LADESC)ERROR"QVQ
 (GOTO)I21"CHINDOHSOND
 CHINDOHSOFF)
 (MENUBRANCH \MENU2R)

SUBMENU DE ALMACENAMIENTO DE GRAFICOS (MENU2R)

CONTINUAR
 Continuar con el programa /GSC?)"RADESC)CESC)
 (MENUBRANCH \MENU1R)

ALMACENAR
 Almacenar el grafico /GSC?)"RADESC)CESC)
 (MENUBRANCH \MENU1R)

TOTAL
 Señal y ruido /GRABQAB2"A.(CEND)COOIND"
 OTSCESC)SEÑAL + RUIDO"
 LADESC)SEÑAL MAS RUIDO"QVQ.
 (GOTO)I21"CHINDOHSOND
 CHINDOHSOFF)
 (MENUBRANCH \MENU2D)

MENU
 Regresar al Menu anterior
 (MENUBRANCH \MENU)

RESULTADO
 Salida del filtro /GRABQAB2"A.(CEND)COOIND"
 OTSCESC)SALIDA DEL FILTRO"
 LADESC)SALIDA"QVQ
 (GOTO)I21"CHINDOHSOND
 CHINDOHSOFF)
 (MENUBRANCH \MENU2R)

COMPARACION
 Grafico de la entrada y salida
 Regresar al menu anterior
 /GRABQAB2"A.(CEND)COOIND"
 (MENUBRANCH \MENU)

BF2"B.(CEND)COOIND)OTSCESC)
 ENTRADA Y SALIDA DEL FILTRO"LA
 CESC)ENTRADA"LB(CESC)SALIDA"QVQ
 (GOTO)I21"CHINDOHSOND
 CHINDOHSOFF)
 (MENUBRANCH \MENU2R)

```

SUBMENU DE GRAFICOS DEL ANALISIS (MENU1A)
CURVA
Graf. Curva de aprendizaje
CHOME)
/FIN*.CUR*(?)~(GOTO)B1~
/DFB1~-----/OF.CLEFT)
CEND)COO)ND-CRIG)HT)-----
CGOTO)B1~
/GR*ARBCDEFQX.CEND)COO)ND~
RABQAR2~A.CEND)COO)ND~
LACESC)C.CURVA DE APRENDIZAJE~
CGOTO)I21~CHINDO)NSO)N)
CHINDO)NSOFF)
(MENUBRANCH \MENU2A)

ENTRADA)FOURIER)
Analisis fourier de s. entrada)
CHOME)
/FIN*.FIN*(?)~(GOTO)B2~
/DFB2~-----/OF.CLEFT)
CEND)COO)ND-CRIG)HT)-----
CGOTO)C2~+B1CRBS)
/2/A1CRBS)*B2~/C~COO)ND
.CLEFT)CEND)COO)ND-CRIG)HT)~
/GR*ARBCDEFQXC2~
X.CEND)COO)ND~
RABQAR2~A.CEND)COO)ND~
OTS(ESC)ESPECTRO DE FOURIER~
TX(ESC)FRECUENCIA (Hz)~TY(ESC)
MAGNITUD RELATIVA (dB)~
LACESC)ENTRADA)QVQ
CGOTO)I21~CHINDO)NSO)N)
CHINDO)NSOFF)
(MENUBRANCH \MENU2A)

SALIDA)FOURIER)
Analisis fourier de s. salida
CHOME)
/FIN*.FOU*(?)~(GOTO)B2~
/DFB2~-----/OF.CLEFT)
CEND)COO)ND-CRIG)HT)-----
CGOTO)C2~+B1CRBS)
/2/A1CRBS)*B2~/C~COO)ND
.CLEFT)CEND)COO)ND-CRIG)HT)~
/GR*ARBCDEFQXC2~
X.CEND)COO)ND~
RABQAR2~A.CEND)COO)ND~
OTS(ESC)ESPECTRO DE FOURIER~
TY(ESC)FRECUENCIA (Hz)~TY(ESC)
MAGNITUD RELATIVA (dB)~
LACESC)SALIDA)QVQ
CGOTO)I21~CHINDO)NSO)N)
CHINDO)NSOFF)
(MENUBRANCH \MENU2A)

```

```

SUBMENU DE ALMACENAMIENTO DE GRAFICOS (MENU2A)
CONTINUAR
Almacenar el grafico
/GO)X(ESC)
NUMERO DE MUESTRAS~
TY(ESC)
AMPLITUD)QQ
(MENUBRANCH \MENU)

ALMACENAR
Almacenar el grafico
/GO)X(ESC)
NUMERO DE MUESTRAS~
TY(ESC)
AMPLITUD)QQ
(MENUBRANCH \MENU)

```

ANEXO C

GENERACION DE RUIDO BLANCO

C.1 CONCEPTOS BASICOS

El teorema del límite central, establece que una variable aleatoria que depende de la suma de cierto número de variables aleatorias independientes tiende a ser gaussiana cuando el número de variables aleatorias crece adquiriendo valores muy grandes.

Un proceso $X(t)$ se dice que es gaussiano si las variables aleatorias $X(t_1), X(t_2), \dots, X(t_n)$ son conjuntamente gaussianas para todo n y para todo grupo de (t_1, t_2, \dots, t_n) .

Un proceso gaussiano se especifica completamente por su función de autocorrelación y su valor medio. Además, si la función de autocorrelación $R(\tau)$ y el valor medio $E[X(t)]$ no varían al producirse un desplazamiento del origen de tiempo, el proceso es estacionario, es decir:

$$R(\tau) = E[X(t)X(t-\tau)] \quad , \text{ es independiente del } \quad (C.1) \\ \text{tiempo y solamente depende de } \tau.$$

$$E[X(t)] = \overline{X(t)} \quad , \text{ constante para todo } t \quad (C.2)$$

El ruido blanco es un proceso cuyo espectro tiene todas las componentes de frecuencia en igual proporción, tal como lo muestra la figura C.1(a). A los procesos de ruido blanco que son gaussianos se les conoce como ruido blanco

gaussiano y les corresponde una densidad espectral similar a la de la figura C.1(a).

De modo que para un proceso de ruido blanco, la densidad espectral de potencia vendrá dada en general por:

$$p(\omega) = A/2 \tag{C.3}$$

Una vez conocida la densidad espectral de potencia, es posible determinar la función de autocorrelación, la cual se obtiene mediante la transformada inversa de Fourier, y se muestra gráficamente en la figura C.1(b).

$$R(\tau) = \frac{A}{2} \delta(\tau) \tag{C.4}$$

donde $\delta(\tau)$ es la función delta de Dirac definida como:

$$\delta(\tau) = 0, \tau \neq 0 \tag{C.5}$$

$$\int_{-\infty}^{+\infty} \delta(t) dt = 1$$

De la ecuación (C.4) se desprende que la autocorrelación es cero para todo τ excepto para $\tau=0$, lo que implica que las variables aleatorias $X(t_1)$ y $X(t_2)$ serán no correlacionadas si $t_1 \neq t_2$; en consecuencia dos muestras diferentes cualesquiera de ruido blanco gaussiano serán no correlacionadas.

El ruido blanco constituye una idealización, puesto que no existe en la práctica debido a que su valor cuadrático medio es infinito, efectivamente usando la relación de Wiener Kinchine se tiene [5].

$$R(0) = E[X^2(t)] = \frac{1}{2\pi} \int_{-\infty}^{+\infty} p(\omega) d\omega = \infty \tag{C.6}$$

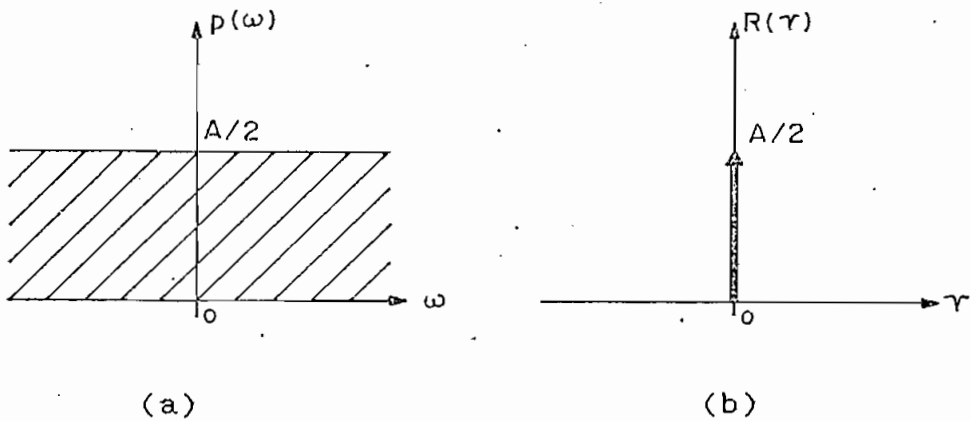


Fig. C.1 Señal de ruido blanco gaussiano. (a) espectro de potencia. (b) función de autocorrelación.

De manera que para todos los casos prácticos se considera al ruido de banda ancha como ruido blanco, y se lo trata como tal, cometiéndose un error despreciable para todo fin práctico.

El ruido blanco gaussiano de banda limitada, es un proceso aleatorio, cuya densidad espectral de potencia es constante en una banda limitada $[-W, W]$ y cero fuera de ella, como lo muestra la figura C.2(a), es decir:

$$p(\omega) = \begin{cases} A/2 & |\omega| < W \\ 0 & \text{otro caso} \end{cases} \quad (C.7)$$

La función de autocorrelación será nuevamente la transformada inversa de Fourier de $p(\omega)$.

$$R(\tau) = AB \text{ Sam}(W\tau) \quad (C.8)$$

donde $B = W/2\pi$ corresponde al ancho de banda en Hz., y $\text{Sam}(W\tau) = \text{Sen}(W\tau)/W\tau$ es la función Sampling.

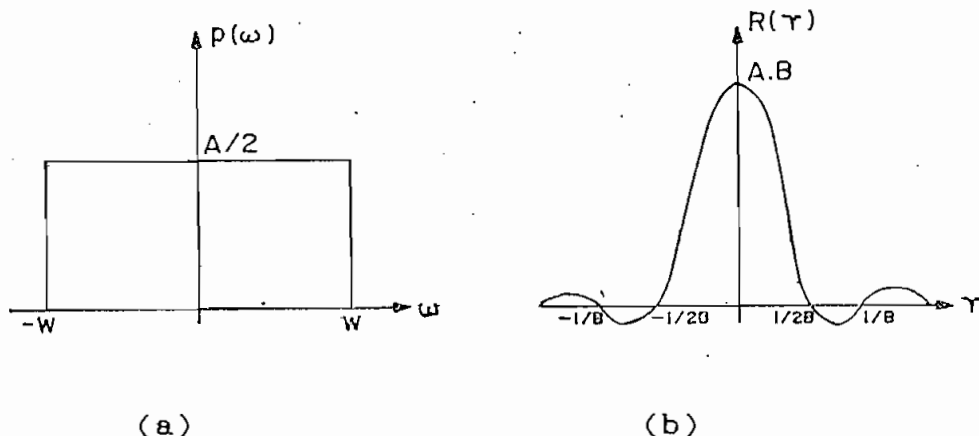


Fig. C.2 Ruido blanco gaussiano de banda limitada. (a) espectro de densidad de potencia. (b) autocorrelación.

Dado que $R(\tau) = E[X(t)X(t+\tau)]$, cuando $\tau=0$ tendremos:

$$R(0) = E[X(t)X(t)] = \overline{X^2} \quad (C.8)$$

Consecuentemente el valor cuadrático medio será:

$$\overline{X^2} = R(0) = AB \quad (C.10)$$

Además se cumple:

$$\lim_{\tau \rightarrow \infty} R(\tau) = (\overline{X})^2 \quad (C.11)$$

pues si suponemos que $X(t)$ no contiene componentes periódicas, para grandes valores de τ , $X(t)$ y $X(t+\tau)$ pierden toda similitud o relación y tienden a ser no correlacionadas, por tanto:

$$\lim_{\tau \rightarrow \infty} R(\tau) = \overline{X(t)X(t+\tau)} = \overline{X(t)} \overline{X(t+\tau)} = (\overline{X})^2 \quad (C.12)$$

De la figura C.2(b) se observa que cuando $\tau \rightarrow \infty$ la función

de autocorrelación $R(\tau) \rightarrow 0$, y por lo tanto el valor medio de este proceso $(\bar{X})^2$ será cero.

De acuerdo con el teorema del muestreo una señal de banda limitada a B [Hz] (es decir una señal que no tiene componentes espectrales más allá de los B Hz.) puede construirse en base a muestras tomadas a intervalos uniformes menores que $1/2B$ seg. De esta manera la señal se expresa matemáticamente como:

$$x(t) = \sum_{k=-\infty}^{\infty} x_k \text{Sam}(Wt - k\pi) \quad (C.13)$$

donde $x_k = x(kT)$ son las muestras tomadas, y $T = 1/2B$ el intervalo de muestreo. El coeficiente x_k es diferente para cada función de muestreo y puesto que X_k ($k = 0, \pm 1, \pm 2, \dots$) es una variable aleatoria. Es posible expresar la ecuación (C.13) para el proceso como [5]:

$$X(t) = \sum_{k=-\infty}^{\infty} X_k \text{Sam}(Wt - k\pi) \quad (C.14)$$

Donde $X_1, X_2, \dots, X_k, \dots$ son variables aleatorias. De manera que el proceso completo se puede especificar por un grupo de variables aleatorias X_k , $k = 0, \pm 1, \dots$ etc.; esto permite reemplazar señales continuas por medio de una secuencia discreta de variables aleatorias sin pérdida de información.

De la figura C.2(b) se puede observar que la función de autocorrelación es cero para $T = n/2B$, con $n = 1, 2, \dots$. Por lo tanto, las muestras tomadas en $T = 1/2B$ ($X_1, X_2, \dots, X_k, \dots$), esto es las muestras de Nyquist, son no correlacionadas; dado que el proceso es gaussiano estas muestras son también gaussianas, y dado que las variables gaussianas son no correlacionadas serán también independientes.

Luego de este análisis para generar ruido blanco gaussiano, deberán generarse las muestras, esto es los valores de las variables aleatorias con distribución gaussiana o normal, que tengan un valor medio cero y varianza σ^2 ; el método que se utilizará para generar este tipo de variables aleatorias es el método del límite central. Previamente se establecerán algunos conceptos básicos acerca de la distribución normal.

C.1.2 DISTRIBUCION NORMAL

Una variable aleatoria que tome valores reales entre $-\infty < x < \infty$, tendrá una distribución normal o gaussiana, con valor esperado μ y desviación estándar σ , si su función de densidad de probabilidad (fdp) está dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right] \quad (\text{C.15})$$

Si los valores de μ y σ son 0 y 1 respectivamente, la función recibe el nombre de distribución normal o estándar; sin embargo, cualquier distribución normal puede convertirse a la forma estándar haciendo la sustitución:

$$y = \frac{x - \mu}{\sigma} \quad (\text{C.16})$$

de manera que la función densidad de probabilidad quedaría:

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp[-y^2/2] \quad (\text{C.17})$$

esta función de densidad de probabilidad también cumple:

$$\int_{-\infty}^{+\infty} f(x) dx = 1 \quad (\text{C.18})$$

C.1.3 METODO DEL LIMITE CENTRAL

Este método se fundamenta en el teorema del límite central, el cual establece que:

Si $X_1, X_2, \dots, X_n, \dots$ es una sucesión de variables aleatorias independientes pero idénticamente distribuidas, con $E[X_i] = \mu_i$ y $V(X_i) = \sigma_i^2$, $i=1, 2, \dots$. Entonces la variable aleatoria suma, esto es $X = X_1 + X_2 + \dots + X_n$, se aproxima asintóticamente a una distribución normal a medida que n se hace muy grande, y dicha distribución tendrá como media y varianza respectivamente a:

$$\mu = \sum_{i=1}^n \mu_i, \quad \sigma^2 = \sum_{i=1}^n \sigma_i^2 \quad (\text{C.19})$$

A fin de simular una distribución normal con media μ y varianza σ^2 , se debe proporcionar la siguiente interpretación matemática al teorema del límite central. Si r_1, r_2, \dots, r_n representan variables aleatorias independientes, cada una de las cuales posee la misma distribución de probabilidad (μ, σ^2) , entonces:

$$\lim_{n \rightarrow \infty} P \left[a < \frac{\sum_{i=1}^n r_i - n\mu}{\sqrt{n} \sigma} < b \right] = \frac{1}{\sqrt{2\pi}} \int_a^b \exp[-y^2/2] dy \quad (\text{C.20})$$

donde

$$\sum_{i=1}^n \mu_i = n\mu \quad (\text{C.21})$$

$$\sum_{i=1}^n \sigma_i^2 = n\sigma^2 \quad (\text{C.22})$$

$$Y = \frac{X - \sum_{i=1}^n \mu_i}{\sqrt{\sum_{i=1}^n \sigma_i^2}} \quad (C.23)$$

tanto de la distribución normal estándar como de la ecuación (C.16) , se deduce que Y es una variable aleatoria con distribución normal estándar.

El procedimiento para simular valores normales mediante una computadora requiere el uso de la suma de K valores de variables aleatorias distribuidas uniformemente; esto es, la suma de r_1, r_2, \dots, r_k , con cada r_i definida en el intervalo $0 < r_i < 1$. Puesto que para una distribución uniforme se cumple que:

$$\mu = \frac{a+b}{2} = \frac{0+1}{2} = \frac{1}{2} \quad (C.24)$$

$$\sigma = \frac{b-a}{\sqrt{12}} = \frac{1-0}{\sqrt{12}} = \frac{1}{\sqrt{12}} \quad (C.25)$$

Como estamos considerando k valores de variables aleatorias, la expresión para (C.23) queda:

$$Y = \frac{\sum_{i=1}^k r_i - k/2}{\sqrt{k/12}} \quad (C.26)$$

Pero por definición Y es una variable aleatoria con distribución normal estándar, la cual se puede expresar en la forma sugerida por (C.16), donde x es el valor de variable aleatoria distribuida en forma normal que se va a generar, con media μ y varianza σ . Igualando las dos expresiones (C.16) y (C.23) se tendrá:

$$\frac{x - \mu}{\sigma} = \frac{\sum_{i=1}^k r_i - k/12}{\sqrt{(k/12)}} \quad (C.27)$$

por lo tanto

$$x = \sigma(12/k)^{1/2} \left\{ \sum_{i=1}^k r_i - k/2 \right\} + \mu \quad (C.28)$$

El valor de k depende del objetivo de la generación de los valores. Si se requiere un menor tiempo de cómputo un valor adecuado es k=12, pues con éste valor se evita una división y una raíz cuadrada, pero nótese que se limitan los valores obtenidos entre +6 y -6, lo cual no importa para el nivel de ruido que deseamos generar, puesto que no se va a superar tales límites.

De manera que la ecuación (C.28), para k=12 quedará:

$$x = \sigma \sum_{i=1}^{12} r_i - 6 \quad (C.29)$$

donde $\mu=0$, pues recuérdese que el proceso de ruido blanco gaussiano tenía media cero.

Filtro Butterworth Pasabajos:

El filtro de Butterworth constituye un diseño mas o menos estándar de un filtro pasabajos. A fin de visualizar en mejor forma su estructura y funcionamiento, será necesario definir algunos parámetros.

Si grafizamos la ganancia de potencia (cuadrado de la respuesta de amplitud) en función de la frecuencia, obtendremos una señal de la forma como se indica en la figura D-2, nótese que la ganancia de potencia es una función par de w , y en el gráfico constan únicamente las frecuencias positivas. En general podemos expresar la ganancia de potencia de un filtro como:

$$\begin{aligned} \text{ganancia de potencia} &= |H(jw)|^2 \\ \text{ganancia de potencia en dB} &= 10\log_{10}|H(jw)|^2 \end{aligned} \quad (D.1)$$

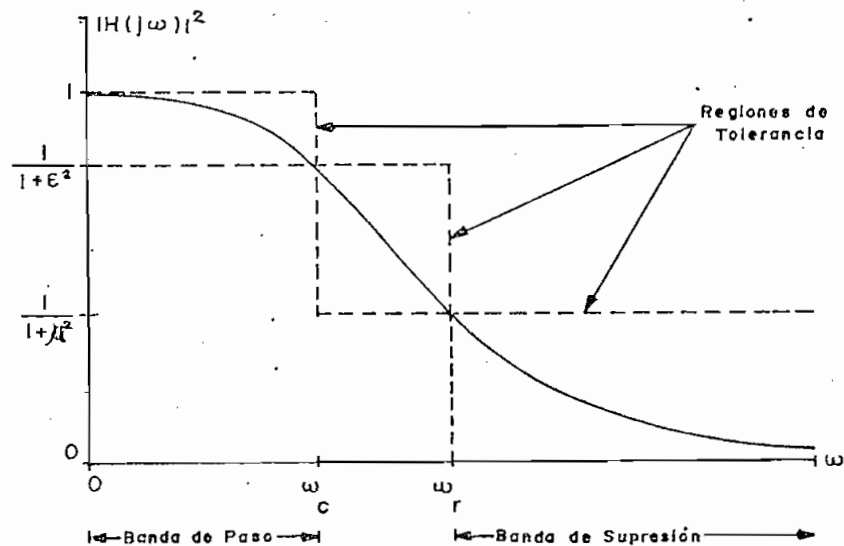


Fig. D.2 Características de ganancia de potencia de un filtro pasabajos.

Además, considerando que las características de un filtro pasabajos ideal no se pueden realizar en forma finita, se definen algunos parámetros y regiones útiles en el diseño de filtros, tal como lo muestra la figura D.2:

ω_c : es la frecuencia de corte, la cual marca el límite superior de la banda de paso, desde 0 hasta ω_c .

ω_r : es la frecuencia de rechazo, la cual es mayor que ω_c , y marca el inicio de la banda eliminada, desde ω_r hasta infinito.

entre la banda de paso y la banda eliminada, esto es en el intervalo $\omega_c < |\omega| < \omega_r$, la ganancia de potencia decrece rápidamente.

En base a los parámetros μ y ϵ , podemos expresar la ganancia de potencia para cada una de las bandas como:

$$\text{banda de paso: } |\omega| < \omega_c ; \quad |H(j\omega)|^2 > \frac{1}{1 + \epsilon^2} \quad (D.2)$$

$$\text{banda eliminada: } |\omega| > \omega_r ; \quad |H(j\omega)|^2 < \frac{1}{1 + \mu^2}$$

Por tanto si hacemos que ϵ se aproxime a cero, ω_r tienda a ω_c , y μ tienda al infinito; tendremos una característica rectangular ideal para el filtro pasabajos.

En general, el filtro análogo de Butterworth tiene una ganancia de potencia dada por la siguiente expresión:

$$|H_B(j\omega)|^2 = \frac{1}{1 + \epsilon^2(\omega/\omega_c)^{2N}} \quad (D.3)$$

donde N corresponde al orden del filtro. Tal filtro se dice que es máximamente plano cerca de $\omega=0$ y $\omega=\infty$. Si hacemos $\omega=\omega_r$ en la ec. (D.3) podemos obtener una expresión para determinar el orden del filtro de acuerdo a:

$$|H_B(j\omega_r)|^2 = \frac{1}{1 + \mu^2} = \frac{1}{1 + \epsilon^2(\omega_r/\omega_c)^{2N}}$$

entonces:

$$(\omega_r/\omega_c)^{2N} = \mu^2/\epsilon^2 = (\mu/\epsilon)^2$$

sacando logaritmos a ambos lados de la ecuación anterior

$$\log [(\omega_r/\omega_c)^{2N}] = \log [(\mu/\epsilon)^2]$$

de modo que:

$$2N \log(\omega_r/\omega_c) = 2 \log(\mu/\epsilon)$$

por tanto la expresión para N quedará:

$$N \geq \frac{\log(\mu/\epsilon)}{\log(\omega_r/\omega_c)} \quad (D.4)$$

de acuerdo a esta expresión, se debe establecer un compromiso entre la cuadratura de las características de ganancia de potencia y el menor valor de N. Las mejoras en las características de ganancia de potencia cuando N se incrementa se muestran en la figura D.3.

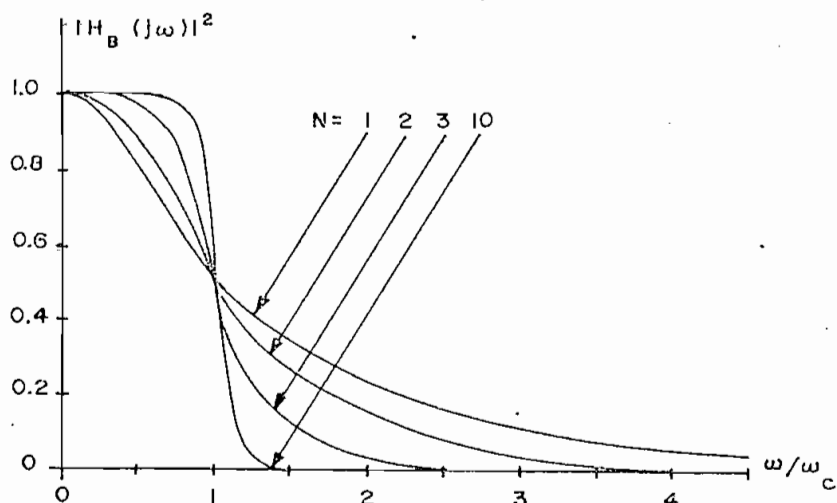


Fig. D.3 Ganancia de potencia para un filtro Butterworth, con $\epsilon=1$

Los polos en el plano s de la función $|H_B(s)|^2$ se encuentran igualando a cero el denominador de la ecuación (D.3), con $s=j\omega$, esto es:

$$1 + \epsilon^2 (s/j \omega_c)^{2N} = 1 + \epsilon^2 (-s^2/\omega_c^2)^N = 0$$

$$1 + \epsilon^2 (s^2/\omega_c^2)^N (-1)^N = 1 + \epsilon^2 (s^2/\omega_c^2)^N (e^{-j\pi})^N = 0$$

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

entonces

$$\epsilon^2 (s^2/\omega_c^2)^N (e^{-j\pi})^N = (e^{-j\pi})$$

despejando

$$(s/\omega_c)^{2N} = \frac{1}{\epsilon^2} e^{-j\pi} e^{j\pi N} = \frac{1}{\epsilon^2} e^{j\pi(N-1)}$$

sacando las raíces de la ecuación anterior, se puede determinar los polos de la función, los mismos que estarán localizados en:

$$s_n = \omega_c \epsilon^{-1/N} e^{j\pi(N-1+2n)/2N} \quad (D.5)$$

con $n = 1, 2, 3, \dots, 2N$

de modo que los polos de la función de ganancia de potencia caen sobre un círculo de radio $\omega_c \epsilon^{-1/N}$ en el plano s , como lo ilustra la figura D.4; los polos de la función de transferencia sola, $H_B(s)$, estarán localizados en el semiplano izquierdo puesto que:

$$|H_B(s)|^2 = H_B(s) H_B(-s) \quad (D.6)$$

y por tanto la ec. (D.5) da los polos de la función de transferencia cuando n es igual a $1, 2, \dots, N$.

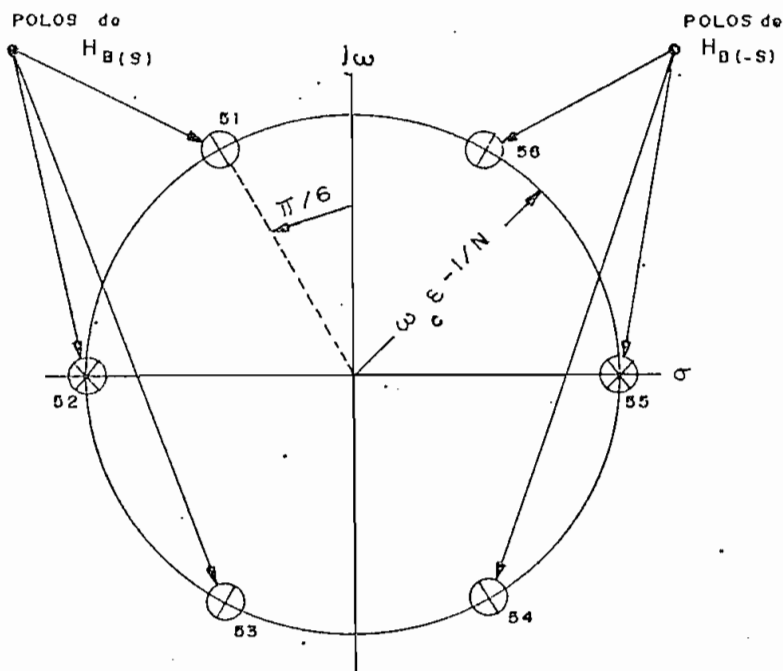


Fig. D.4 Polos del filtro Butterworth sobre el plano s .
 $N=3$

La función de transferencia del filtro vendrá dada por:

$$H_B(s) = \frac{1}{(1-s/s_1)(1-s/s_2)\dots(1-s/s_N)} \quad (D.7)$$

tal ecuación se puede expresar de la siguiente forma:

$$H_B(s) = \frac{(-1)^N s_1 s_2 \dots s_N}{(s-s_1)(s-s_2)\dots(s-s_N)} \quad (D.8)$$

D.2 CONVERSION DE FILTROS ANALOGOS EN DIGITALES POR MEDIO DE LA TRANSFORMACION BILINEAL

La transformación bilineal es considerada como una transformación desde el plano s hacia el plano z , lo cual permite la conversión de polos y ceros análogos en polos y ceros digitales.

El objetivo fundamental no es una simulación exacta, se trata más bien de obtener un filtro digital con mejores características respecto de su original el análogo (por ejemplo más rectangular). Un método que nos permite cumplir satisfactoriamente con tales requerimientos es la transformación bilineal, mediante la cual, la variable s en $H(s)$ se sustituye por una función de z para obtener $H^*(z)$. La función de z para dicha transformación está dada por:

$$s \leftarrow A \frac{z - 1}{z + 1} \quad (D.9)$$

donde $A = \text{constante}$

La transformación bilineal con $A=1$ es la transformación más simple del eje $j\omega$ al círculo unitario sobre el plano z , de manera que la ecuación (D.9) quedará:

$$s \leftarrow \frac{z - 1}{z + 1} \quad (D.10)$$

la misma que al ser transformada al dominio de la frecuencia da una relación:

$$\begin{aligned} j\omega' &= \frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} \\ &= \frac{e^{j\omega T/2} - e^{-j\omega T/2}}{e^{j\omega T/2} + e^{-j\omega T/2}} \end{aligned}$$

de modo que

$$\omega' = \tan(\omega T/2) \quad (D.11)$$

La figura D.5 muestra la relación entre ω' y ωT .

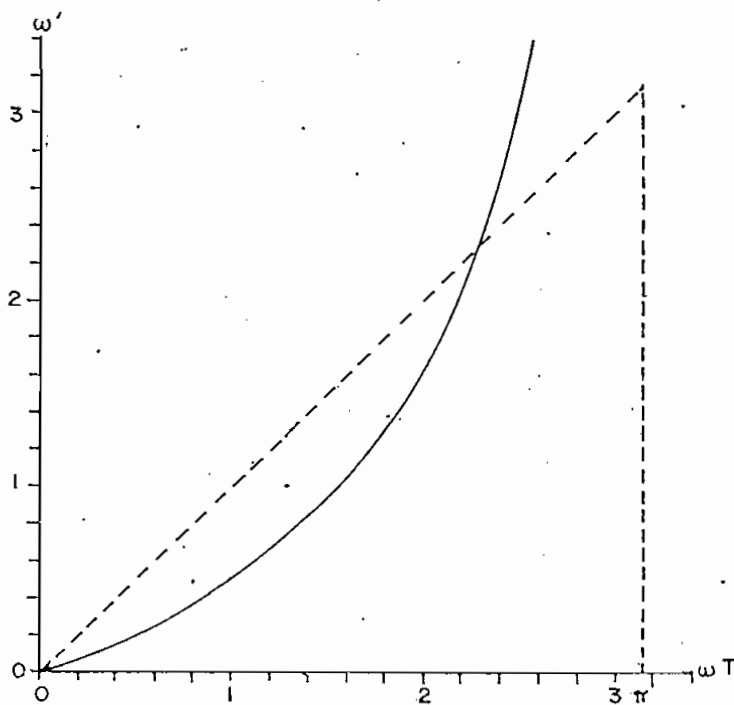


Fig. D.5 La transformación de frecuencia bilineal, $\omega' = \tan(\omega T/2)$

A continuación se detallará el procedimiento a seguir para usar la transformación bilineal.

1. Determinar los valores transformados de las frecuencias críticas del filtro, w_c' y w_r' , usando la ecuación (D.11).

$$w' = \tan (wT/2)$$

2. Diseñar un filtro análogo con función de transferencia $H_A(s)$, pero usando w_c' y w_r' en lugar de w_c y w_r .

3. Transformar $H_A(s)$ en $H^*(z)$. Para el efecto usar la relación (D.10).

$$s \leftarrow \frac{z - 1}{z + 1}$$

donde $H^*(z)$ vendrá a constituir la función de transferencia del filtro digital deseado.

Por lo que de acuerdo con la ecuación (D.8), la función de transferencia del filtro queda:

$$H^*(z) = \frac{(-1)^N s_1 s_2 \dots s_N}{\left[\frac{z-1}{z+1} - s_1 \right] \left[\frac{z-1}{z+1} - s_2 \right] \dots \left[\frac{z-1}{z+1} - s_N \right]} \quad (D.12)$$

procesando la ecuación anterior es posible llegar a obtener una expresión de la forma:

$$H^*(z) = \frac{(-1)^N s_1 s_2 \dots s_N (z+1)^N}{[(1-s_1)z - (1+s_1)] \dots [(1-s_N)z - (1+s_N)]} \quad (D.13)$$

La exactitud del "ajuste" de ω_c y ω_r , puede demostrarse mediante el siguiente argumento: De acuerdo al paso 3,

$$H^*(z) = H_A(z-1/z+1) \quad (D.14)$$

pero la función de transferencia del filtro digital es:

$$H^{\wedge}(j\omega) = H^*(e^{j\omega T}) \quad (D.15)$$

a partir de la ecuación (D.11), y usando (D.15) tendremos:

$$H^{\wedge}(j\omega) = H_A(j\omega') \quad (D.16)$$

de manera que el paso 2 del procedimiento genera los valores de ganancia deseados $H^{\wedge}(j\omega_c)$ y $H^{\wedge}(j\omega_r)$ a las frecuencias ω_c y ω_r respectivamente.

La ecuación (D.13) nos permite encontrar la ganancia de potencia del filtro digital, para lo cual debemos sustituir $e^{j\omega T}$ por z y encontrar la magnitud al cuadrado de dicha ecuación; sin embargo un método mucho más simple sería usar (D.16) y (D.11), de modo que se tendrá:

$$\begin{aligned} |H^{\wedge}(j\omega)|^2 &= |H_A(j\omega')|^2 \\ &= |H_A[j \tan(\omega T/2)]|^2 \end{aligned} \quad (D.17)$$

con lo que únicamente restaría sustituir ω_c' por ω_c y ω_r' por ω_r en la ecuación (D.3) de la ganancia de potencia de Butterworth, para obtener:

$$|H^{\wedge}(j\omega)|^2 = \frac{1}{1 + \epsilon^2 \left[\frac{\tan(\omega T/2)}{\tan(\omega_c T/2)} \right]^{2N}} \quad (D.18)$$

La función de transferencia digital $H^*(z)$ dada por la ecuación (D.13) se puede realizar en forma de cascada, para lo cual se debe tomar en consideración que s_n y s_{N+1-n} , dados por la ecuación (D.5), son complejos conjugados, esto es:

$$s_n = R \exp(j\theta_n) \tag{D.19}$$

$$s_{N+1-n} = R \exp(-j\theta_n)$$

donde

$$R = \omega_c' e^{-1/N} \qquad \theta_n = \frac{2n+N-1}{2N} \pi \tag{D.20}$$

Si consideramos que N (orden del filtro) es un número par, podremos descomponer al filtro en secciones de segundo orden, donde el número de secciones vendría dado por $N/2$. De manera que la función de transferencia del filtro será:

$$H^*(z) = H_1^*(z) H_2^*(z) \dots H_{N/2}^*(z) \tag{D.21}$$

donde

$$\begin{aligned} H_n^*(z) &= \frac{s_n s_{N+1-n} (z+1)^2}{[(1-s_n)z - (1+s_n)] [(1-s_{N+1-n})z - (1+s_{N+1-n})]} \\ &= \frac{R^2 (z+1)^2}{\{[1-R\exp(j\theta_n)]z - [1+R\exp(j\theta_n)]\} \{[1-R\exp(-j\theta_n)]z - [1+R\exp(-j\theta_n)]\}} \\ &= \frac{R^2 (z^2 + 2z + 1)}{(1 + R^2 - 2R\cos\theta_n)z^2 - 2(1 - R^2)z + (1 + R^2 + 2R\cos\theta_n)} \end{aligned} \tag{D.22}$$

con n variando desde 1 hasta $N/2$. La realización del filtro digital en cascada dada por las ecuaciones (D.21) y (D.22) se muestra en los diagramas de la figura D.5.

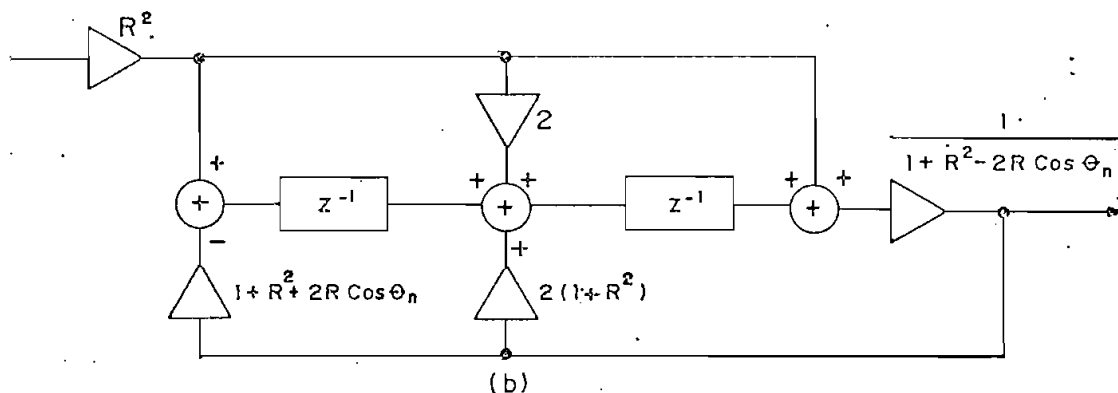
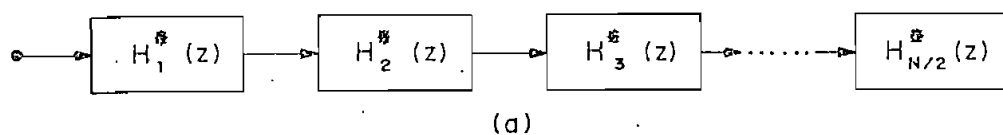


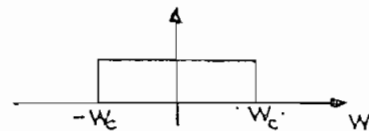
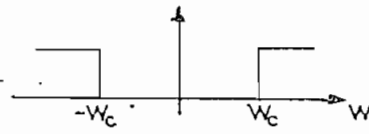
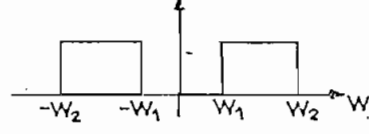
Fig. D.5 (a) Realización en cascada de un filtro digital de Butterworth. (b) Diagrama de flujo para una sección del filtro $H_n^*(z)$.

D.3 TRANSFORMACION EN FRECUENCIA

La conversión de un filtro pasabajos en un filtro pasaalto o pasabanda, involucra la sustitución de una función de s , $f'(s)$, dentro de la función de transferencia análoga del pasabajos, $H(s)$, para proveer una nueva función de transferencia, $H(f')$. La conversión tiene las siguientes propiedades importantes:

1. La nueva función de transferencia obtenida luego de la transformación, $H(jw')$, tiene exactamente el mismo valor que la función del pasabajos original, $H(jw)$, pero a valores diferentes de w .
2. Los filtros digitales diseñados mediante la transformación bilineal pueden convertirse fácilmente, convirtiendo su versión analoga al nuevo tipo de filtro deseado, y posteriormente aplicando la sustitución $s \leftarrow (z-1)/(z+1)$.

Las fórmulas de conversión constan en la tabla D-1. Sobre la derecha de la tabla están los resultados de las transformaciones del dominio de la frecuencia, obtenidas con $s=jw$.

Pasabajos	$s \leftarrow s$	
Pasaaltos	$s \leftarrow \frac{w_0^2}{s}$	
Pasabanda	$s \leftarrow \frac{s^2 + w_1w_2}{s}$	

donde $w_0 = w_2 - w_1$

Tabla D.1 Conversión de filtros pasa bajos.

Conversión de un filtro pasabajos analógico en un filtro pasaaltos digital:

Para el caso del filtro pasaaltos, la banda de paso del filtro pasabajos original $|w| < w_c$, es llevada al dominio donde $|w| > w_c$, de esta manera se genera la nueva banda de paso deseada a frecuencias mayores que w_c .

El procedimiento para convertir un filtro pasabajos análogo en un filtro pasaaltos digital, consistirá en partir de la ecuación de la función de transferencia del pasabajos, dada por la ec. (D.8).

$$H_L(s) = \frac{(-1)^N s_1 s_2 \dots s_N}{(s-s_1)(s-s_2)\dots(s-s_N)} \quad (D.23)$$

donde los polos de dicha función vendrán dados por:

$$s_n = \omega_c' e^{-1/N} e^{j\pi(2n+N-1)/2N} \quad ; \quad n=1,2,\dots,N \quad (D.24)$$

Es decir, ya se realizó la transformación correspondiente de ω_c en ω_c' de acuerdo con la ec. (D.11), luego, antes de aplicar la transformación bilineal para convertir el filtro análogo en un filtro digital, se debe introducir la relación dada por la tabla D.1; donde s es reemplazada por ω_c^2/s en la fórmula de $H_L(s)$, de esta manera se obtiene la nueva función de transferencia para el filtro pasaaltos, $H_H(s)$:

$$\begin{aligned} H_H(s) &= H_L(\omega_c'^2/s) \\ &= \frac{(-1)^N s_1 s_2 \dots s_N}{(\omega_c'^2/s - s_1)(\omega_c'^2/s - s_2)\dots(\omega_c'^2/s - s_N)} \end{aligned}$$

trabajando sobre esta ecuación se llega a la siguiente expresión:

$$= \frac{(-1)^N s^N}{(s - \omega_c'^2/s_1)(s - \omega_c'^2/s_2)\dots(s - \omega_c'^2/s_N)} \quad (D.25)$$

como se puede deducir del análisis de ésta ecuación, los polos originales dados por la ecuación (D.24) son transformados uno a uno sobre el plano- s , introduciéndose ade-

más un cero múltiple en $s=0$. La función de ganancia de potencia del filtro pasaaltos de acuerdo con la ecuación (D.3) vendrá dada por:

$$\begin{aligned} |H_H(j\omega)|^2 &= |H_L(\omega_c'/\omega)|^2 \\ &= \frac{1}{1 + \epsilon(\omega_c'/\omega)^2} \end{aligned} \quad (D.26)$$

restará ahora aplicar la transformación bilineal para obtener finalmente el filtro digital pasaaltos, este procedimiento se tratará mas en detalle para el caso de conversión de un filtro pasabajos en un filtro pasabanda digital.

Conversión de un filtro Pasabajos análogo en un filtro Pasabanda digital:

La transformación a un filtro pasabanda es un tanto más complicada, y genera un filtro de orden igual al doble del original, puesto que s^2 está implícito en la transformación. Como se muestra en la tabla, el filtro pasabajos original se debe diseñar con una frecuencia de corte:

$$\omega_c = \omega_2 - \omega_1 \quad (D.27)$$

Donde la frecuencia de corte es igual al ancho de banda del filtro pasabanda, lo anteriormente expuesto se puede demostrar mediante el siguiente argumento: Sean las funciones de transferencia de los filtros pasabajos y pasabanda $H_L(j\omega)$ y $H_B(j\omega)$ respectivamente. Sustituyendo de acuerdo a la línea 3 de la tabla tendremos:

$$j\omega \leftarrow \frac{-\omega^2 + \omega_1\omega_2}{j\omega}$$

Consecuentemente

$$H_B(j\omega) = H_L \left[j \frac{\omega^2 - \omega_1\omega_2}{\omega} \right] \quad (D.28)$$

$$H_B(\pm j\omega_1) = H_L[\pm j(\omega_1 - \omega_2)] = H_L(\mp j\omega_c)$$

$$H_B(\pm j\omega_2) = H_L[\pm j(\omega_2 - \omega_1)] = H_L(\pm j\omega_c)$$

De modo que la ganancia del filtro pasabajos para $|\omega| < \omega_c$ es igual a la ganancia del filtro pasabanda para $\omega_1 < |\omega| < \omega_2$.

Para realizar la conversión de un filtro pasabajos análogo en un filtro pasabanda digital procedemos de acuerdo al método ya indicado anteriormente, es decir primero convertimos el filtro pasabajos en un filtro análogo pasabanda apropiado; y, luego usamos la transformación bilineal para obtener finalmente el filtro digital correspondiente.

De acuerdo al procedimiento señalado debemos primero hacer la transformación de ω_1 en ω_1' y ω_2 en ω_2' , de acuerdo a la ec.(D.11), esto es

$$\omega_1' = \tan(\omega_1 T/2) \quad (D.29)$$

$$\omega_2' = \tan(\omega_2 T/2)$$

posteriormente se determinará ω_c' , mediante la relación dada por la tabla D.1:

$$\omega_c' = \omega_2' - \omega_1' \quad (D.30)$$

La ganancia de potencia del pasabajos será por consiguiente:

$$|H_L(j\omega)|^2 = \frac{1}{1 + \epsilon^2 (\omega/\omega_c')^{2N}} \quad (D.31)$$

De manera que la ganancia de potencia del filtro pasabanda análogo será:

$$\begin{aligned} |H_B(j\omega)|^2 &= \left| H_L j \left(\frac{\omega^2 - \omega_1' \omega_2'}{\omega} \right) \right|^2 \\ &= \frac{1}{1 + \epsilon^2 \left[\frac{\omega^2 - \omega_1' \omega_2'}{\omega \omega_c'} \right]^{2N}} \end{aligned} \quad (D.32)$$

A partir de esta ecuación la ganancia de potencia del filtro pasabanda digital se puede determinar tomando en consideración la ec.(D.17) y vendrá dada por:

$$\begin{aligned} |H^*(j\omega)|^2 &= |H_B[j \tan(\omega T/2)]|^2 \\ &= \frac{1}{1 + \epsilon^2 \left[\frac{\tan^2(\omega T/2) - \omega_1' \omega_2'}{\omega_c' \tan(\omega T/2)} \right]^{2N}} \end{aligned} \quad (D.33)$$

Podemos encontrar los polos de la función de transferencia $H^*(z)$ transformando los polos del filtro pasabajos análogo (ec.D.23), primero en polos de un pasabanda análogo, y luego en polos sobre el plano z de un filtro pasabanda digital.

Para obtener los polos del pasabanda análogo, de acuerdo a la tabla D-1 tendremos:

$$s_L = \frac{s_B^2 + w_1'w_2'}{s_B} \quad (D.34)$$

donde s_L son los polos del filtro pasabajos y s_B son los polos del filtro pasabanda, por consiguiente:

$$s_B = 1/2 [s_L \pm (s_L^2 - 4w_1'w_2')^{1/2}] \quad (D.35)$$

de manera que por cada polo del pasabajos análogo obtendremos dos polos del pasabanda análogo.

Aplicando luego la transformación bilineal

$$s = (z-1)/(z+1) \quad (D.36)$$

nos dará los polos del filtro pasabanda digital

$$z_B = (1+s_B)/(1-s_B) \quad (D.37)$$

El filtro pasabanda digital también puede implementarse en forma de cascada, donde el número de secciones vendrá dado igualmente por la mitad del orden del filtro; y, cada sección tendrá una función de transferencia de la forma:

$$H^*_k(z) = \frac{A_k(z^4 - 2z^2 + 1)}{z^4 + B_kz^3 + C_kz^2 + D_kz + E_k} \quad (D.38)$$

BIBLIOGRAFIA

- [1] Bernard Widrow and others, Adaptive noise cancelling: Principles and Applications, Proceedings of the IEEE, Vol 63, No. 12, December 1975, pp 1692-1718.
- [2] Luis Rodriguez, Simulación Estadística, EPN, 1983.
- [3] Samuel D. Stearns, Digital Signal Analysis, Hayden Book Company, Inc.
- [4] C.F.N Cowan and P.M. Grant, Adaptive Filters, Prentice Hall, 1985.
- [5] B.P. Lathi, An introduction to Random Signals and Communication Theory. International Textbook Company, 1968.
- [6] S.M. Bozic, Digital and Kalman Filtering, 1981.
- [7] Earl R. Ferrara and Bernard Widrow, The time-sequenced Adaptive Filters, IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 6, June 1981, pp 519-523.
- [8] Bernard Widrow and others, Adaptive Antenna Systems, Proceedings of the IEEE, Vol. 55, No. 12, December 1967, pp 2143-2159.
- [9] Bernard Widrow and others, Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter, Proceedings of the IEEE, Vol. 64, No. 8, August 1976, pp 1151-1162.
- [10] M. Honing and D.G. Messerschmitt, Adaptive Filters: Structures, Algorithms and Applications, Kluwer Aca-

demic Publishers Group, Dordrecht, 1984.

- [11] Alan V. Oppenheim and Ronald W. Schaffer, Digital Signal Processing, Prentice - Hall Inc. , New Jersey, 1975, pp 8-18.
- [12] T.A.C.M. Claasen and W.F.G. Mecklenbräuker, Adaptive Techniques for Signal Processing in Communications. IEEE Communications Magazine, November 1985, Vol. 23, No. 11.