



DEPARTAMENTO DE
COMUNICACIONES



UNIVERSIDAD
POLITECNICA
DE VALENCIA

*Universidad Politécnica de Valencia
Departamento de Comunicaciones*

Filtrado adaptativo multicanal para control local
de campo sonoro basado en algoritmos de
proyección afín

Tesis Doctoral

Presentada por:

Miguel Ferrer Contreras

Dirigida por

Dr. Alberto González Salvador

Dra. María de Diego Antón

Valencia, Octubre de 2008



Instituto de Telecomunicaciones
y Aplicaciones Multimedia

A la memoria de mi padre
y a toda mi familia

Agradecimientos

Es difícil tratar de recoger en unas breves líneas el profundo sentimiento de agradecimiento que me invade al recordar a todas las personas cuyo apoyo y ayuda me ha sido de incalculable valor a lo largo de todos estos años de trabajo. Sé que este pequeño reconocimiento puede parecer un formalismo, pero creo que nunca sería suficiente si tratase alguna vez de recompensarlos como ellos se merecen.

Quisiera comenzar agradeciendo a mi familia su confianza incondicional y sus ánimos tanto en los momentos buenos como en los menos buenos. Aún en la distancia puedo sentir su aliento y sobre todo sus oraciones, lo que ha hecho que me sienta siempre un auténtico privilegiado y me impulsa a tratar de superarme día a día sabiendo que cada cosa que hago en esta vida está dedicada a ellos. Así que a todos ellos, simplemente, gracias por existir.

Por supuesto, quisiera ser igual de agradecido a las personas que durante todo este tiempo han ejercido voluntaria o involuntariamente de mi familia, preocupándose y cuidando de mí, ofreciéndome su tiempo, sus palabras, sus consejos ... Especialmente me gustaría agradecer a las personas que me han tenido que sufrir más de cerca, como son todos los compañeros del GTAC con los que he tenido la fortuna de coincidir durante todos estos años, así como a mis amigos y compañeros del DCOM-G los que siempre me han brindado su inestimable ayuda y a los que espero no haber agotado su paciencia.

Gracias, también, a Germán Ramos, por su apoyo y ayuda en el desarrollo del sistema en tiempo real con DSP's y por sus constantes ánimos, así como a mi amigo Javier Galindo, por la minuciosa revisión realizada al documento final de esta tesis.

Finalmente, quisiera agradecer muy especialmente a Alberto González y María de Diego (directores de esta tesis doctoral) toda la ayuda brindada durante este tiempo. Ellos son los auténticos artífices de esta tesis doctoral y sin sus orientaciones no habría sido posible. Gracias por su infinita paciencia, y por haberme concedido la oportunidad de trabajar todo este tiempo junto a ellos.

Resumen

Esta Tesis doctoral se ha centrado en el desarrollo e implementación de algoritmos eficientes multicanal, basados en el algoritmo de proyección afín, aplicados al control activo de ruido. Para abordar esta cuestión primeramente se han estudiado diferentes algoritmos eficientes de proyección afín que han sido analizados y validados mediante simulación, finalizando con la implementación, en un recinto, de un sistema real de control activo de ruido multicanal ejecutado en un DSP controlado por dichos algoritmos.

En los últimos años, los algoritmos de proyección afín han sido propuestos como algoritmos de control en sistemas adaptativos, que pretenden mejorar la velocidad de convergencia de los algoritmos basados en el LMS, siendo una alternativa eficiente, robusta y estable frente a estos algoritmos, cuya limitación principal es, precisamente, la velocidad de convergencia. Los algoritmos de proyección afín pueden ser considerados como una extensión natural del algoritmo NLMS, ya que éste actualiza sus coeficientes basándose en un único vector de datos de la señal de entrada mientras que los algoritmos de proyección afín actualizan los coeficientes de los filtros adaptativos usando N vectores de datos de la señal de entrada (siendo N el orden de proyección). Se han dedicado muchos esfuerzos para tratar de optimizar la eficiencia computacional de estos algoritmos aplicados al problema de la cancelación de eco, surgiendo diferentes versiones eficientes del algoritmo de proyección afín. Sin embargo, al aplicarlo al control activo de ruido, es necesario reducir aún más la complejidad computacional, teniendo en cuenta que, por lo general, la eficiencia computacional se consigue a costa de la degradación de alguna otra característica del algoritmo (generalmente la velocidad de convergencia). En este trabajo se presentan algunas alternativas a versiones eficientes existentes, que no degradan significativamente las prestaciones de dicho algoritmo, y se analiza cómo reducir aún más la carga computacional de estos algoritmos. Además, también se discute cómo implementar un algoritmo eficiente de proyección afín basado en la estructura de filtrado- x convencional con el ahorro computacional que dicha estructura conlleva comparada con la estructura de filtrado- x modificada, usada habitualmente cuando el algoritmo de proyección

afín se aplica al control activo de ruido. Aunque la estructura de filtrado-x modificada proporciona una velocidad de convergencia mayor que la convencional y permite trabajar con todas las señales que se necesitan en el algoritmo de proyección afín, también es cierto que posee un coste computacional más elevado puesto que necesita realizar un mayor número de filtrados que los realizados por la estructura de filtrado-x convencional, por lo que esta última estructura introduce, de por sí, una mejora en la eficiencia computacional. En este trabajo se demuestra que la versión eficiente del algoritmo de proyección afín presentada y que se basa en la estructura de filtrado-x convencional tiene tan buenas prestaciones como las de las versiones eficientes basadas en la estructura de filtrado-x modificada, proporcionando un importante ahorro computacional.

El estudio y análisis de las propiedades del algoritmo desarrollado se ha realizado mediante:

- El desarrollo de algoritmos de proyección afín multicanal aplicados al control activo de ruido y basados en la estructura de filtrado-x convencional, estudiando diferentes estrategias eficientes que optimizan el coste computacional. También se ha realizado un estudio teórico del comportamiento en estado estacionario y transitorio de estos algoritmos haciendo uso del principio de conservación de la energía.
- La obtención de resultados de simulación, lo que ha sido posible gracias a un software de simulación desarrollado para tal propósito.
- La implementación de estos algoritmos en un sistema real basado en DSP, donde se ha comprobado el buen funcionamiento de los mismos y que su coste computacional es abordable por los sistemas hardware actuales de gama media (al contrario que los algoritmos de proyección afín basados en la estructura modificada).

Resum

Esta Tesi doctoral està centrada en el desenvolupament i implementació d'algoritmes eficients multicanal, basats en l'algoritme de projecció afí, aplicats al control actiu de soroll. Per abordar esta qüestió en primer lloc s'han estudiat diferents algoritmes eficients de projecció afí que han sigut analitzats i validats per mitjà de simulacions. Finalment, s'ha implementat en un recinte un sistema real de control actiu de soroll multicanal executat en un DSP controlat pels mateixes algoritmes.

Als últims anys, els algoritmes de projecció afí han sigut proposats com a algoritmes de control en sistemes adaptatius, que pretenen millorar la velocitat de convergència dels algoritmes basats en el LMS, i són una alternativa eficient, robusta i estable enfront d'estos algoritmes, encara que la limitació principal és, precisament, la velocitat de convergència. Els algoritmes de projecció afí es podem considerar una extensió natural de l'algoritme NLMS, ja que este actualitza els seus coeficients basant-se en un únic vector de dades del senyal d'entrada mentre que els algoritmes de projecció afí actualitzen els coeficients dels filtres adaptatius usant N vectors de dades del senyal d'entrada (sent N l'ordre de projecció). S'han dedicat molts esforços per a tractar d'optimitzar la eficiència computacional d'estos algoritmes aplicats al problema de la cancel·lació d'eco, sorgint diferents versions eficients de l'algoritme de projecció afí. No obstant, al aplicar-ho al control actiu de soroll, és necessari reduir encara més la complexitat computacional, tenint en compte que, pel general, l'eficiència computacional s'aconsegueix a costa de la degradació d'alguna altra característica de l'algoritme (generalment la velocitat de convergència). En este treball es presenten algunes alternatives a versions eficients existents, que no degraden significativament les prestacions de tal algoritme, i s'analitza com reduir encara més la càrrega computacional d'estos algoritmes. A més, també es discutix com implementar un algoritme eficient de projecció afí basat en la estructura de filtrat-x convencional amb l'estalvi computacional que la dita estructura comporta comparada amb l'estructura de filtrat-x modificada, usada habitualment quan l'algoritme de projecció afí s'aplica al control actiu de soroll. Encara que la estructura de filtrat-x modificada proporciona una

velocitat de convergència major que la convencional i permet treballar amb tots els senyals que es necessiten en l'algoritme de projecció afí, també és cert que presenta un cost computacional més elevat ja que necessita realitzar un nombre més gran de filtrats que els realitzats per l'estructura de filtrat-x convencional, pel que esta última estructura introduïx, de per si, una millora en l'eficiència computacional. En este treball es demostra que la versió eficient de l'algoritme de projecció afí presentada i que es basa en l'estructura de filtrat-x convencional té tan bones prestacions com les de les versions eficients basades en l'estructura de filtrat-x modificada, proporcionant un important estalvi computacional.

L'estudi i anàlisi de les propietats de l'algoritme desenvolupat es realitza per mitjà de:

- El desenvolupament d'algoritmes de projecció afí multicanal aplicats al control actiu de soroll i basats en l'estructura de filtrat-x convencional, estudiant diferents estratègies eficients que optimitzen el cost computacional. També es realitza un estudi teòric del comportament en estat estacionari d'estos algoritmes a través del càlcul del seu error quadràtic mig.
- L'obtenció de resultats de simulació, mitjançant un programari de simulació desenvolupat per a tal propòsit.
- La implementació d'estos algoritmes en un sistema real DSP, on es comprova el bon funcionament dels mateixos i que el seu cost computacional es abordable pels sistemes actuals maquinari de gamma mitjana (al contrari que els algoritmes de projecció afí basats en l'estructura modificada).

Abstract

The aim of this thesis is the development and implementation of efficient affine projection algorithms for multichannel active noise control. Different approaches to this issue have been considered. First of all, various efficient affine projection algorithms have been developed; a specific software allows to validate by simulation the presented algorithms, and finally, the performance of the adaptive controllers considered have been validated in a real practical system by measurement of the acoustical sensation inside an enclosure.

In recent years, affine projection algorithms have been proposed for adaptive system applications as an efficient alternative to the slow convergence speed of least mean square (LMS) type algorithms, showing, then, good performance, robustness and stability. These algorithms update the weights based, instead of on the current input vector (as the NLMS algorithm), on N previous input vectors, being N the projection order. For that reason, the affine projection algorithm can be considered as a NLMS extension. Whereas much attention has been focused on the development of efficient versions of affine projection algorithms for echo cancellation applications, the similar adaptive problem presented by active noise control (ANC) systems has not been studied so deeply. This work is focused on the necessity to reduce even more the computational complexity of affine projection algorithms for real-time ANC applications. We present some alternative efficient versions of existing affine projection algorithms that do not significantly degrade performance in practice. Furthermore, while in the ANC context the commonly used affine projection algorithm is based on the modified filtered-x structure, an efficient affine projection algorithm based on the (non-modified) conventional filtered-x structure, as well as efficient methods to reduce its computational burden, are discussed and analyzed throughout this thesis. Although the modified filtered-x scheme exhibits better convergence speed than the conventional filtered-x structure and allows recovery of all the signals needed in the affine projection algorithm for ANC, the conventional filtered-x scheme provides a significant computational saving, avoiding the additional filtering needed by the modified filtered-x structure. In this work, it is shown that the proposed efficient

versions of affine projection algorithms based on the conventional filtered-x structure show good performance, comparable to the performance exhibited by the efficient approaches of modified filtered-x affine projection algorithms, and also achieve meaningful computational savings.

The commented points will be studied by means of the following experiment and analysis:

- Development of the multichannel affine projection algorithms for active noise control based on the conventional filtered-x structure and on the modified structure. Moreover, different efficient strategies for the affine projection algorithms have been studied. Finally, a theoretical study of the steady-state mean square error of the affine projection algorithms will be carried out.
- Simulation results have been obtained. For this purpose, a software has been specifically designed to validate the developed algorithms.
- Finally, the ability of the affine projection algorithms to reduce the sound field in a multichannel ANC system has been evaluated by means of a real practical implementation.

Notación y lista de acrónimos

x	Escalar.
\mathbf{x}	Vector.
\mathbf{X}	Matriz.
\mathbf{I}	Matriz identidad.
$(\cdot)^T$	Matriz traspuesta.
$ \cdot $	Valor absoluto.
$\ \cdot\ $	Norma vectorial o matricial.
$\ \cdot\ _{\Sigma}$	Norma ponderada.
\otimes	Producto de Kronecker.
$\det(\cdot)$	Determinante.
$E\{\cdot\}$	Valor medio.
$\text{Tr}(\cdot)$	Traza.
I	Número de señales de referencia.
J	Número de actuadores secundarios.
K	Número de señales de sensores de error.
L	Número de coeficientes de los filtros adaptativos.
M	Número de coeficientes de los filtros FIR usados para modelizar los caminos secundarios.

AP	Affine Projection (Proyección afín).
CAR	Control Activo de Ruido.
DSP	Digital Signal Processor.
MSE	Mean Square Error.
EMSE	Excess Mean Square Error.
FAP	Fast Affine Projection.
FIR	Finite Impulse Response.
IIR	Infinite Impulse Response.
LMS	Least Mean Square.
LS	Least Square.
NLMS	Normalized Least Mean Square.
RLS	Recursive Least Square.
FTF	Fast Transversal Filters.
MELMS	Multiple Error Least Mean Square.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos de la tesis	2
1.3. Estructuración de los contenidos	3
2. Principios básicos del filtrado adaptativo y control activo de ruido	5
2.1. Introducción	5
2.2. Control activo de ruido acústico	5
2.2.1. Motivación	5
2.2.2. Principios de funcionamiento del CAR	6
2.2.3. Evolución histórica	8
2.2.4. Descripción del CAR	10
2.2.5. Clasificación de los sistemas de control activo de ruido .	13
2.3. Sistemas adaptativos	16
2.4. Algoritmos de gradiente estocástico	20
2.4.1. Coeficientes que realizan el filtrado óptimo	20
2.4.2. Algoritmos de gradiente	21
2.4.3. Algoritmo LMS (<i>least mean square</i>)	23
2.5. Algoritmos de mínimos cuadrados	24
2.5.1. Minimización mediante mínimos cuadrados (LS)	25
2.5.2. Algoritmo RLS (<i>recursive least square</i>)	30
2.6. Filtrado de <i>Kalman</i>	34

2.7. Sistemas adaptativos aplicados al CAR	39
2.7.1. Estructura directa	42
2.7.2. Estructura clásica o de filtrado-x convencional	43
2.7.3. Estructura de filtrado-x modificada	47
2.7.4. Estructura adjunta	50
2.7.5. Elección de la estructura	52
3. Algoritmos de proyección afín y sus versiones eficientes	55
3.1. Introducción	55
3.2. Algoritmo LMS normalizado	56
3.3. Extensión del NLMS hasta el algoritmo de proyección afín.	59
3.4. Optimización computacional de los AP	62
3.4.1. Cálculo eficiente de la matriz inversa	62
3.5. Algoritmo eficiente de proyección afín (FAP)	64
3.5.1. Cálculo eficiente del vector de error	65
3.5.2. Cálculo eficiente de la actualización de los coeficientes	66
3.5.3. Cálculo eficiente del vector de error normalizado	72
4. Aplicación del algoritmo de proyección afín y sus versiones eficientes al CAR	77
4.1. Introducción	77
4.2. Algoritmos de proyección afín aplicados al CAR	79
4.2.1. Algoritmo de proyección afín multicanal usando la estructura de filtrado-x modificada	80
4.2.2. Algoritmo de proyección afín multicanal usando la estructura de filtrado-x convencional	84
4.3. Estudio de las prestaciones de los algoritmos propuestos	88
5. Estudio del error cuadrático medio de los algoritmos de proyección afín.	93
5.1. Introducción	93
5.2. Estudio del error cuadrático medio del algoritmo de proyección afín	94

5.3. Estudio del error cuadrático medio del algoritmo de proyección afín aplicado al CAR	104
5.3.1. MSE usando estructura modificada	106
5.3.2. MSE usando estructura de filtrado-x	114
5.4. Comparación de las diferentes estrategias y modelos	123
5.5. Estudio del error cuadrático medio del algoritmo de proyección afín multicanal aplicado al CAR	128
5.5.1. Estructura modificada multicanal	131
5.5.2. Estructura de filtrado-x multicanal	142
5.6. Resumen de relaciones matemáticas	148
5.6.1. Obtención de la relación de conservación de la energía	148
5.6.2. Manipulaciones matemáticas a la ecuación de conservación de la energía	148
5.6.3. Eliminación de la dependencia de $\mathbf{e}_N[n]$ en la ecuación de conservación de energía	149
5.6.4. Obtención del modelo del EMSE monocanal usando actualización eficiente del vector de error (estructura de filtrado-x modificada)	150
5.6.5. Obtención del modelo del EMSE (estructura de filtrado-x convencional)	155
5.6.6. Obtención del modelo del EMSE usando cálculo eficiente del vector de error (estructura de filtrado-x convencional)	158
5.6.7. Cálculo de la norma ponderada en el algoritmo multicanal	161
5.6.8. Justificación del valor aproximado de $E\{(\mathbf{V}^T[n]\mathbf{U}[n])\}$	162
6. Resultados de simulación	165
6.1. Introducción	165
6.2. Comparativa genérica	166
6.3. Comparativa para diferentes tipos de ruido	171
6.4. Efecto del retardo de los caminos secundarios en la velocidad de convergencia	172

7. Implementación práctica del algoritmo de proyección afín	179
7.1. Introducción	179
7.2. Descripción del prototipo y sistema de medidas	179
7.3. Análisis de las medidas	185
7.3.1. Ruido aleatorio	190
7.3.2. Tono 100 Hz	202
7.3.3. Ruido periódico	213
7.3.4. Robustez	224
7.4. Coste computacional real	229
8. Conclusiones y líneas futuras	231
8.1. Conclusiones	231
8.2. Líneas futuras	234
A. Comportamiento en régimen transitorio de los algoritmos de proyección afín.	237
A.1. Introducción	237
A.2. Análisis del transitorio del algoritmo de proyección afín con la estructura de filtrado-x modificada aplicado al CAR	238
A.3. Análisis del transitorio del algoritmo de proyección afín con la estructura de filtrado-x convencional aplicado al CAR	244
B. Estrategias eficientes aplicadas a los algoritmos LS	249
B.1. Introducción	249
B.2. Algoritmo <i>fast Kalman</i> o <i>fast transversal filter</i>	249
B.3. Algoritmo <i>fast Kalman</i> usando el error a posteriori	256
B.4. Algoritmos de ventana deslizante	260
B.5. Solución del problema mediante mínimos cuadrados (LS) con enventanado deslizante	261
B.6. Mínimos cuadrados recursivos con ventana deslizante	263
B.7. <i>Fast RLS</i> con enventanado deslizante	265

C. Recursión de <i>Levinson</i>	275
C.1. Introducción	275
C.2. Recursión de <i>Levinson</i>	275
C.3. Recursión de <i>Levinson</i> modificada	280
D. Simulador software de sistemas acústicos	287
D.1. Introducción	287
D.2. Breve descripción de la barra de menús	288
D.3. Definición de los algoritmos	299
E. Publicaciones relacionadas con la tesis	303
BIBLIOGRAFÍA	315

Capítulo 1

Introducción

1.1. Motivación

La realización de esta tesis doctoral ha formado parte de una línea de investigación abierta en el Grupo de Tratamiento de Audio y Comunicaciones (GTAC) del Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM) de la Universidad Politécnica de Valencia (UPV) centrada en el control de ruido acústico mediante métodos activos aplicando estrategias adaptativas, donde se han desarrollado hasta el momento trabajos relacionados, por ejemplo, con la cancelación de ruido acústico en un tubo o la ecualización de ruido periódico en recintos (usando algoritmos de control multicanal). Esta línea de investigación ha sido financiada, entre otros, por los proyectos CICYT TIC2000-1683-C03 *Desarrollo e implementación de algoritmos de procesamiento de señal y de computación de altas prestaciones para sistemas de emulación de entornos acústicos virtuales* y CTIDIA/2002/116 *Sistema de control local de campo sonoro para mejora subjetiva de la calidad del sonido en entornos cerrados*.

Aprovechando la experiencia del GTAC en temas de control activo de ruido (CAR), el presente trabajo se ha centrado también en el estudio de algoritmos adaptativos eficientes aplicados al control activo de ruido acústico. Concretamente, se ha estudiado el algoritmo de proyección afín (AP) y sus versiones eficientes (FAP), cuya aplicación había estado, hasta el momento de iniciar este estudio, principalmente enfocada a la cancelación de eco.

En la bibliografía actual, existen multitud de algoritmos y variantes que pretenden solucionar adaptativamente el problema de la minimización de una señal (que habitualmente se denomina de error) con la particularidad de que conforme mejores son las prestaciones de dichos algoritmos (sobre todo en

cuanto a velocidad de convergencia), mayor es el coste computacional del mismo. El reto es encontrar un algoritmo con una relación de compromiso óptima entre prestaciones y coste computacional.

Aunque los algoritmos estudiados han sido particularizados para la aplicación de control activo de ruido, sería prácticamente inmediato adaptarlos para cualquier otra aplicación de control donde se pretenda minimizar cualquier parámetro de la señal de error, por lo que las conclusiones en este trabajo alcanzadas tienen validez y aplicación en la solución de cualquier otro problema adaptativo.

1.2. Objetivos de la tesis

- Dada la multitud de métodos adaptativos existentes en la bibliografía, el objetivo de este trabajo ha sido tratar de encontrar y mejorar algunos algoritmos cuyas prestaciones sean óptimas y que además su coste computacional sea abordable por el hardware existente en la actualidad. Los algoritmos de proyección afín parecen reunir estas condiciones, puesto que pueden mejorar las propiedades de convergencia de los algoritmos adaptativos clásicos (como el LMS) manteniendo buenas propiedades de estabilidad, con un coste computacional moderado. Es por ello que esta familia de algoritmos se ha convertido en el objetivo de investigación central de esta tesis.
- Por las características propias de estos algoritmos, las estrategias basadas en los algoritmos de proyección afín han venido usándose principalmente para otro tipo de aplicaciones al margen del control activo de ruido (como la cancelación de eco), para las cuales se han desarrollado versiones computacionalmente eficientes (algoritmos *fast*) que reducen el número de operaciones a realizar. Por tanto, en este trabajo se tratarán de adaptar los algoritmos eficientes existentes (basados en el algoritmo de proyección afín) al problema del control activo de ruido y mejorar los esfuerzos realizados por otros autores en esta línea.
- Por supuesto, todas las variantes y nuevas aportaciones a desarrollar, serán estudiadas con detalle tratando de definir sus comportamientos tanto matemáticamente como por medio de simulaciones.
- Por último, se tratará de validar experimentalmente aquello que se haya desarrollado teóricamente, con la finalidad de dar sentido práctico a todo el proceso y sobre todo de constatar que el coste computacional de los algoritmos sobre los que versa este trabajo se encuentra dentro de

las capacidades del hardware actual y por tanto, podrían ser perfectos candidatos para el controlador de un sistema de CAR en una aplicación real.

1.3. Estructuración de los contenidos

La presentación del trabajo realizado en esta tesis doctoral se ha estructurado en los capítulos que se describen a continuación:

Capítulo 1. A modo de introducción a la tesis, se definen principalmente los objetivos de la misma.

Capítulo 2. Se presenta el CAR, definiendo brevemente su evolución histórica y las diferentes clasificaciones de los sistemas existentes, así como se introducen los sistemas adaptativos (como herramienta que se usará para los sistemas de control activo de ruido), centrándonos en los filtros adaptativos de estructura transversal y en los algoritmos adaptativos clásicos (algoritmos de gradiente, de mínimos cuadrados y filtrado de *Kalman*). Por último, se enumeran las diferentes estructuras existentes para aplicar los algoritmos adaptativos al control activo de ruido.

Capítulo 3. Se describe el algoritmo de proyección afín, y diferentes estrategias que permiten ejecutar eficientemente el algoritmo. Aunque estas estrategias ya han sido usadas en diversos trabajos propuestos por otros autores con anterioridad, se pretende en este capítulo realizar una revisión del estado del arte del algoritmo de proyección afín y sus versiones eficientes, aportando como novedad una estrategia eficiente para la optimización computacional de la inversión matricial que aparece en estos algoritmos.

Capítulo 4. En él se adapta el algoritmo de proyección afín y sus versiones eficientes, al problema de control activo de ruido multicanal, siendo original la adaptación de estas estrategias a la estructura de filtrado-x convencional.

Capítulo 5. Presenta un análisis en estado estacionario de los algoritmos considerados, mediante la obtención de una estimación del error cuadrático medio en régimen permanente, basada en el principio de conservación de la energía. Para mayor claridad de este estudio, inicialmente se reproducen los resultados ya conocidos del análisis del error cuadrático medio del algoritmo de proyección afín aplicado al problema de identificación de canal para posteriormente, usando la misma metodología, extrapolar este resultado a las estrategias aplicadas al control activo de ruido desarrolladas en este trabajo, aportando modelos matemáticos originales del comportamiento de estos algoritmos tanto en sus versiones monocanal como multicanal.

Capítulo 6. Se presentan los resultados obtenidos mediante simulación de las estrategias adaptativas presentadas en este estudio, pudiendo comparar las prestaciones de las mismas.

Capítulo 7. En él se resumen los resultados obtenidos en la implementación práctica en un recinto de un sistema multicanal de control activo de ruido usado para cancelación, empleando como algoritmo controlador el algoritmo de proyección afín.

Capítulo 8. Contiene las conclusiones alcanzadas en este trabajo y algunas de las posibles líneas futuras de investigación a las que ha dado lugar.

Anexo A. Presenta un modelo matemático del transitorio de los algoritmos de proyección afín monocanal aplicados al control activo de ruido, lo que resulta una aportación novedosa en el caso de dicho algoritmo funcionando bajo la estructura de filtrado-x convencional.

Anexo B. En este anexo se refleja la metodología que aparece en la bibliografía para implementar eficientemente algunos de los diferentes algoritmos clásicos discutidos en esta tesis y que podrán ser extrapoladas para realizar versiones eficientes del algoritmo de proyección afín.

Anexo C. Se describe el algoritmo recursivo de *Levinson*, y así poder identificar la analogía que algunas de las ecuaciones que aparecen en el anexo anterior presentan con dicho algoritmo.

Anexo D. Se presenta un simulador acústico software desarrollado como parte de la realización de la tesis y que ha sido una herramienta importante para realizar los estudios de las prestaciones de los diferentes algoritmos estudiados.

Anexo E. Enumera las diferentes publicaciones relacionadas con esta tesis doctoral.

Capítulo 2

Principios básicos del filtrado adaptativo y control activo de ruido

2.1. Introducción

Puesto que el presente trabajo se basa en el desarrollo y análisis de un algoritmo de filtrado adaptativo particularizado en el control activo de ruido, resulta adecuado empezar describiendo ambos conceptos (control activo de ruido (CAR) y sistemas adaptativos). Por lo tanto, en este capítulo se realizará una breve introducción al control activo de ruido [1], presentándose también, los sistemas adaptativos y los algoritmos clásicos usados en dichos sistemas [2]. Finalizaremos el capítulo con una descripción de las estrategias (estructuras) más frecuentes para aplicar los algoritmos adaptativos al CAR.

2.2. Control activo de ruido acústico

2.2.1. Motivación

El ruido acústico, aunque es un fenómeno existente desde el principio de los tiempos, no ha sido considerado realmente como un problema hasta el reciente desarrollo industrial y tecnológico donde los avances técnicos han venido casi siempre acompañados de efectos secundarios nocivos como la multiplicación de las molestas fuentes de ruido. Así, conforme el hombre ha ido evolucionando, ha desarrollado herramientas y ha diseñado métodos que le ayudarán a sobrevivir y mejorar su calidad de vida, dando lugar en muchos de los casos a la creación

de nuevas fuentes sonoras de origen artificial y de naturaleza generalmente desagradable.

Con el paso del tiempo, los sonidos debidos a los fenómenos de la naturaleza como el viento, la lluvia o las tormentas, han dejado de ser las únicas fuentes de ruido acústico, pasando a ser, en muchos casos, las de menor importancia. La generación del ruido de origen industrial, o el debido a los medios de locomoción, entre otros, han motivado la aparición del concepto de *contaminación acústica*.

La sociedad ha ido poco a poco pagando el precio de un desarrollo tecnológico a costa de convivir con una degradación del medio ambiente, hasta que ha podido darse cuenta del verdadero problema que esto supone, momento en el cual se han llevado a cabo iniciativas para controlar los posibles efectos nocivos de los agentes contaminantes. Hoy en día, esta preocupación ha llegado también a la contaminación acústica, puesto que está demostrado que estar expuesto a altos niveles de ruido, no solo provoca la degradación progresiva del aparato auditivo, sino que también provoca estrés, pudiendo llegar a dañar el sistema nervioso.

Es por esto, que las técnicas de reducción de ruido acústico cobran cada día mayor importancia, y el desarrollo de nuevos sistemas suele incluir un doble desafío: conseguir la mayor eficiencia con la menor contaminación posible (incluida por supuesto la contaminación acústica).

2.2.2. Principios de funcionamiento del CAR

Tradicionalmente, el control de ruido ha venido realizándose mediante las denominadas técnicas pasivas [3], consistentes en la introducción de barreras físicas (llamadas silenciadores) que intentan aislar la fuente de ruido del entorno que le rodea. Estas técnicas se basan en los siguientes principios:

- Principio de reflexión: la introducción de determinados materiales o pantallas acústicas, provocan una disminución en la potencia acústica debido a las distintas reflexiones a las que se ve sometida la onda sonora al atravesar diferentes medios con impedancias distintas.
- Principio de dispersión: existen determinados materiales denominados absorbentes que provocan pérdidas de energía debidas a la expansión del sonido al incidir sobre la superficie de dichos materiales.

Las técnicas pasivas de control de ruido, son hoy en día la primera opción a la hora de luchar contra la contaminación acústica y proporcionan buenos resultados a frecuencias medias y altas. Sin embargo, a medida que se reduce

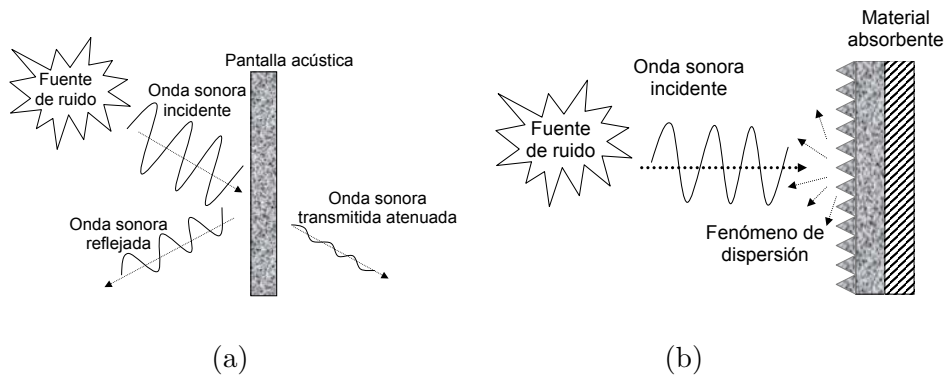


Figura 2.1: Actuación de los métodos pasivos para el control de ruido (a) principio de reflexión, (b) principio de dispersión.

la frecuencia de la señal ruidosa que se desea silenciar, las técnicas pasivas suelen ser más ineficientes y resultan bastante más caras, siendo necesario un significativo incremento del volumen en los materiales usados, provocando en muchos casos que el tamaño del material silenciador necesario impida implementar el sistema total.

A diferencia de las técnicas pasivas, las técnicas activas para el control de ruido, se basan en la generación de otras ondas sonoras de forma que el nuevo campo acústico generado sumado con el campo acústico original ruidoso, se cancele o al menos reduzca considerablemente su nivel. La idea del principio del control activo de ruido se muestra en la figura 2.2, y con más detalle, puede consultarse en uno de los artículos originales del tema como [4].

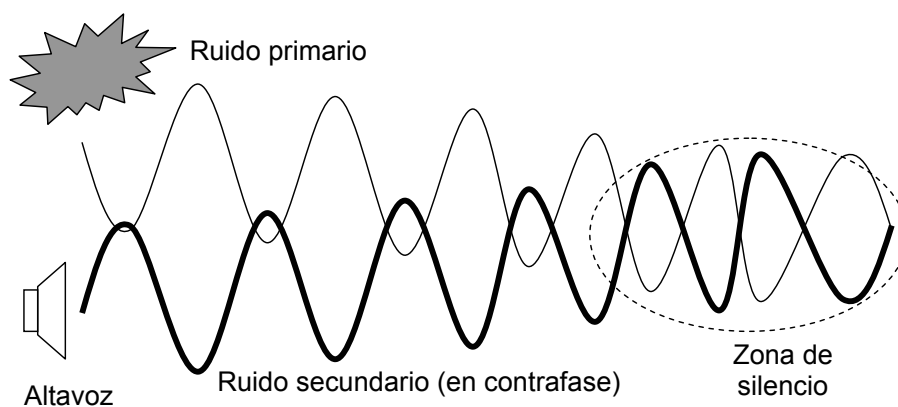


Figura 2.2: Principio en el que se basa el control activo de ruido.

Como se desprende de la figura 2.2, el control activo de ruido se basa en la idea de crear una zona donde se produzca interferencia destructiva entre la señal original y la generada.

Las técnicas activas obtienen mejores resultados conforme las longitudes de ondas de las señales de ruido son más grandes. Por lo tanto, resultan ideales para ejercer control de ruido a bajas frecuencias.

Así pues, podemos pensar que las técnicas de control activo de ruido pueden ser el complemento ideal para las técnicas pasivas, a la hora de implementar sistemas de control en todo el margen frecuencial. Sin embargo, no es solo esto lo que hace realmente interesantes a las técnicas de control activo. Si pensamos en el modo en el que actúan las técnicas pasivas de ruido, podemos darnos cuenta que conseguimos atenuar la señal acústica sin tener un excesivo control sobre el campo acústico final que queda. Además, no es un sistema modificable a posteriori, sino que suele ser un sistema poco flexible. En cambio, el hecho de que el control activo esté basado en la creación de una zona de interferencia y que tengamos la posibilidad de generar el campo acústico interferente, nos concede la posibilidad no solo de cancelar la señal existente sino de conformar el campo acústico final. Es decir, los sistemas de control activo de ruido, además de conseguir resultados bastante buenos a bajas frecuencias, ofrecen como valor añadido la posibilidad de ecualizar la señal de ruido. Además, mediante el empleo conjunto de técnicas adaptativas, los sistemas activos tienen la capacidad de ajustarse a las posibles variaciones del entorno cambiante.

Esto proporciona un inmenso abanico de posibilidades para tratar el ruido, ya que no solo podemos disminuir los niveles de ruido existentes, sino que podemos modificar su espectro buscando otra serie de propiedades (sobre todo de carácter psicoacústico) con las que conseguiríamos modelar el ruido de forma que puedan ser otros criterios (incluso de índole subjetiva) los que decidan cómo actuar frente al ruido, buscando el disponer no únicamente de mayor silencio, sino de una sensación de mayor confort acústico o poder modificar el ruido conforme a criterios subjetivos tales como la deportividad o musicalidad de los sonidos (por enumerar alguno de ellos).

Por todo esto, el control activo de ruido alcanza hoy un gran interés y se presenta como una herramienta válida para la lucha contra la contaminación acústica.

2.2.3. Evolución histórica

Los principios en los que se basa el control activo de ruido son conocidos desde hace mucho tiempo atrás, pero no es hasta 1934 cuando se tiene

constancia de los primeros estudios realizados para cancelar ruido mediante sistemas activos [5]. Dichos trabajos fueron realizados por Paul Lueg y han quedado recogidos en su patente presentada en 1936 y llamada “*Process of Silencing Sound Oscillations*”. Con su sistema, Lueg pretendía cancelar un tono mediante la creación de otro tono de fase opuesta. Su sistema usaba una línea de retardo para ajustar la fase de la señal y poder lograr de ese modo la interferencia destructiva. En [6] podemos encontrar una discusión sobre esta patente. Desafortunadamente, la tecnología electrónica de los años treinta no estaba lo suficientemente desarrollada como para favorecer las investigaciones acerca de este tipo de temas y el sistema era muy simple y de dudosa aplicación práctica, pues solamente se podía controlar una frecuencia. Pero de esta forma, Lueg se convirtió en el pionero de la investigación del control activo de ruido.

Hubo que esperar hasta el año 1953 para que Harry Olson, siguiendo los pasos de Paul Lueg, le diera un nuevo impulso al control activo de ruido, ideando varios sistemas como el “absorbente electrónico de sonido” o el “auricular cancelador de ruido” [7]. Aunque ya en esta época la electrónica había avanzado mucho, y se desarrolló un primer sistema de control de ruido local que introducía métodos activos para la cancelación, los sistemas analógicos existentes seguían estando bastante limitados a la hora de controlar la fase de las señales.

Casi de forma simultánea, un ingeniero de la General Electric llamado Conover, trató de desarrollar un sistema basado en el control activo de ruido para tratar de minimizar el ruido en los transformadores, poniendo una fuente sonora muy cerca del transformador [8]. Los resultados que se obtuvieron no fueron del todo satisfactorios puesto que la reducción de ruido que se logró no era uniforme en todas las direcciones del espacio (incluso para algunos ángulos aumentaba el nivel de ruido) por lo que el proyecto fue abandonado en 1956.

Pero no fue hasta mediados de los sesenta, cuando los estudios de control activo de ruido tomaron gran relevancia debido principalmente a los trabajos de Jessel, sobre ruido en conductos, y de Kido sobre ruido de transformadores, junto con los avances producidos en la tecnología. De hecho, los pioneros en la introducción de las técnicas digitales en el control activo de ruido fueron Kido [9] y Chaplin [10].

Otros muchos autores como Hesselman, Ross, Angevine, Bschorr o Davidson también contribuyeron con sus publicaciones a desarrollar la incipiente teoría sobre el control activo de ruido.

El extraordinario avance de los componentes electrónicos de estado sólido que proporcionó el boom de la electrónica y dio paso a la llamada “era digital”, facilitó el desarrollo de las computadoras abriendo nuevas posibilidades a las

técnicas de control activo de ruido. En paralelo, se había ido desarrollando la teoría de tratamiento digital de la señal, lo que conjuntamente con la aparición del hardware adecuado para el procesado digital ha supuesto el poder disponer de una poderosa herramienta para desarrollar nuevos sistemas de control activo de ruido mucho más eficientes y ha conseguido que este campo de investigación resulte mucho más atractivo.

En nuestros días, el tratamiento digital de la señal y más concretamente, la teoría desarrollada de filtros adaptativos [2], supone el principal soporte teórico para implementar los sistemas de control activo, y los sistemas DSP (Digital Signal Processor) son la herramienta hardware ideal.

Los sistemas de control de ruido en recintos desarrollados en la actualidad permiten ejercer control sobre ruido de banda ancha (aunque hasta frecuencias no demasiado elevadas, como 500 Hz) [11], [12], e incluso permiten el poder ecualizar ruido cuando éste es ruido periódico o de banda estrecha (compuesto únicamente por un conjunto de tonos de fase inicial aleatoria) [13], [14].

2.2.4. Descripción del CAR

Como ya se ha mencionado con anterioridad, las estrategias para el control activo de ruido están basadas en el principio de superposición de señales en sistemas lineales.

Realmente, un sistema de control activo de ruido local no es más que un sistema capaz de generar las ondas acústicas apropiadas para que, sumadas con las ondas ruidosas originales, se cancelen en unos determinados puntos del espacio donde el sistema puede monitorizar el nivel del campo acústico existente. Es decir, lo que se consigue es crear una zona de interferencia destructiva en el entorno de unos determinados puntos del espacio, tanto más grande como mayor sea la longitud de onda de la señal a cancelar. Se ha podido demostrar [15] que puede conseguirse una considerable atenuación (más de 10 dB) en una zona alrededor del punto donde se ejerce el control de aproximadamente un décimo de la longitud de onda ($\lambda/10$) de la señal a cancelar, lo que supone unos 34 cm a 100 Hz. En el resto del espacio no tiene porque haber reducción del nivel, pudiendo incluso aumentar.

Por tanto, un sistema de control acústico activo de ruido típico está compuesto básicamente por un conjunto de altavoces o fuentes sonoras, un conjunto de micrófonos o sensores acústicos y un sistema controlador que monitoriza la señal de ruido existente y genera lo que podemos denominar como el campo acústico cancelador.

Para que un sistema de control activo de ruido cumpla la misión para la que fue diseñado, en el entorno donde se le hace trabajar debe haber, al menos, una

fuente de ruido y, en el caso de un sistema con control *feedforward* (sistemas en los que se centrará este trabajo), debemos tener un sensor adecuado que nos de información acerca de dicho ruido. En la mayoría de los casos dicho sensor podría ser un simple transductor electroacústico, pero también puede tratarse de cualquier otro tipo de sensor que nos proporcione la información necesaria del ruido que deseamos cancelar.

En la figura 2.3 se muestra lo que podría ser un sistema de control activo de ruido básico *feedforward*.

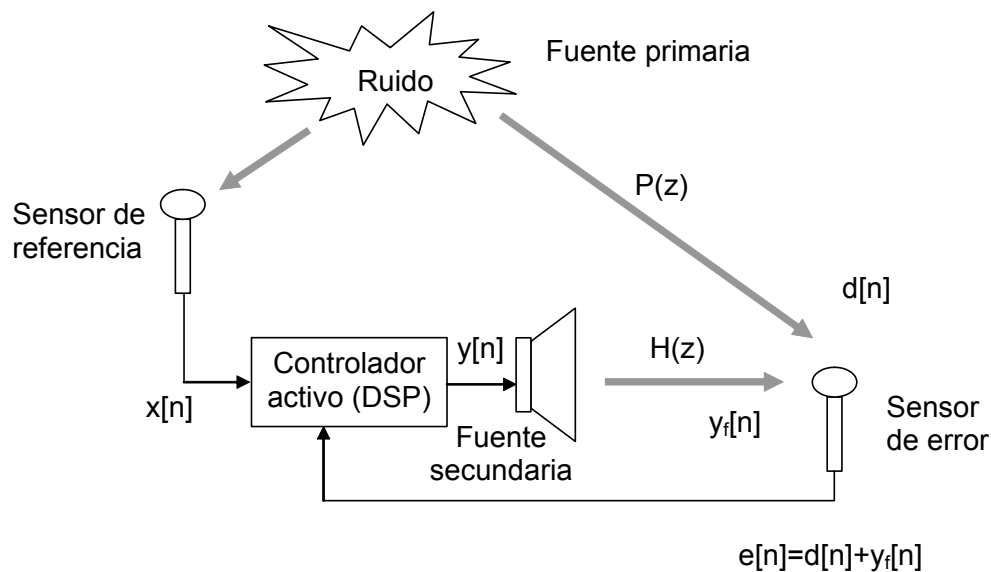


Figura 2.3: Componentes de un sistema de control activo local de ruido monocanal *feedforward*.

En dicha figura podemos distinguir los diferentes componentes elementales de un sistema CAR que a continuación se enumeran siguiendo la nomenclatura clásica:

- Sensor de error: se trata de un sensor acústico (normalmente un micrófono) que recoge información del campo acústico existente en un determinado punto del espacio donde se desea ejercer control activo de ruido.
- Fuente primaria: es la fuente generadora del ruido y su naturaleza puede ser muy diversa.
- Fuente secundaria: es un altavoz o cualquier tipo de dispositivo capaz de generar ondas sonoras mediante el cual creamos la señal que está desti-

nada a sumarse con el ruido existente y de esta forma modificar el campo acústico original intentando cancelarlo o ecualizarlo si es el caso.

- Controlador activo de ruido: se trata de un procesador (normalmente un DSP) donde se ejecutan los algoritmos de control activo de ruido.
- Sensor de referencia: es otro transductor mediante el cual el controlador activo de ruido obtiene la información necesaria de la señal de ruido que se pretende controlar cuando existe control *feedforward*. Según la fuente de ruido que se trate y las necesidades del sistema, dicho sensor suele ser de tipo electroacústico (micrófono) o electromecánico (acelerómetro, tacómetro,...).

En la figura 2.3 también podemos apreciar las diferentes señales que intervienen en un sistema de control activo de ruido y que generalmente se denominan como:

- $e[n]$ (Señal de error): es la señal recogida por el sensor de error y sobre la que actuarán los algoritmos de control, que en caso de ser adaptativos intentarán minimizar alguno de sus parámetros (normalmente su potencia media).
- $d[n]$ (Señal primaria): también conocida como ruido primario y representa a la señal de ruido original a cancelar.
- $y[n]$ (Señal secundaria): también denominada señal canceladora o de antirruído o ruido secundario, por ser la señal procedente de la fuente secundaria.
- $x[n]$ (Señal de referencia): es una señal necesaria en los sistemas *feedforward* para que el algoritmo de control activo de ruido realice su labor correctamente, y se genera a partir de la información procedente del sensor de referencia.

Por último, para definir por completo la nomenclatura de un sistema de control activo de ruido, también tenemos que enumerar los siguientes conceptos:

- Caminos primarios: son los caminos acústicos que debe recorrer la señal primaria desde la fuente de ruido hasta cada uno de los sensores de error.
- Caminos secundarios: son los caminos acústicos que debe recorrer cada una de las señales secundarias desde que sale de la fuente secundaria

hasta que llega a los sensores de error. A diferencia de los caminos primarios, el conocimiento del comportamiento de los caminos secundarios es transcendental para el buen funcionamiento de los algoritmos de control activo de ruido.

Estos caminos acústicos se comportan como un filtro acústico que altera las amplitudes de las ondas sonoras e introducen un retardo dependiente de la longitud física del camino. En la mayoría de los sistemas adaptativos aplicados al CAR, necesitaremos estimarlos adecuadamente, pues necesitamos reproducir los efectos de dicho filtro acústico en los algoritmos, como ya veremos en los capítulos sucesivos.

Un detalle a tener en cuenta para la implementación de los sistemas de control activo de ruido es la linealidad. Si el sistema no es lineal no va a ser posible que los algoritmos lineales de control activo funcionen bien, si bien es cierto que podríamos conseguir un sistema eficiente mediante el uso de algoritmos no lineales a costa de mayor complejidad. Sin embargo, los algoritmos no lineales no son objeto de estudio en este trabajo.

Cuando se aplica control activo de ruido en recintos, hay que tener en cuenta que el campo acústico no se propaga libremente, sino que sufre reflexiones en las paredes dando lugar a la creación de ondas estacionarias a ciertas frecuencias dependientes de la geometría del recinto y que se conocen como modos acústicos del recinto.

La existencia de dichos modos provoca que haya determinados puntos en el recinto donde dichos modos presentan nulos de presión. Cada uno de esos nulos resulta crítico puesto que en los puntos donde existen es imposible excitar dicho modo. Por tanto, hay que intentar evitar colocar las fuentes generadoras o los sensores acústicos necesarios para el control activo en los nulos, si queremos controlar dichos modos acústicos.

2.2.5. Clasificación de los sistemas de control activo de ruido

Los sistemas de control activo de ruido pueden clasificarse atendiendo a diversos criterios:

1. Según el número de elementos del sistema:
 - Sistema monocanal: cuando solo existe un único sensor de error y una fuente secundaria y por tanto un solo camino secundario.
 - Sistema multicanal: se trata de la extensión natural en el espacio de los sistemas monocanal, y son sistemas dotados de más de un sensor

de error y/o más de una fuente secundaria. Tratan con esto de ampliar la zona de actuación del sistema de control activo de ruido mediante la superposición de las diferentes zonas de atenuación que se crean alrededor de cada uno de los sensores individuales.

En general, cuando nos referimos a un sistema de control activo de ruido, éste viene definido en cuanto a su número de elementos mediante tres cifras separadas por dos puntos, haciendo referencia al número de fuentes primarias la primera de ellas, al número de fuentes secundarias el segundo dígito, y al número de sensores de error el último. Así, un sistema monocanal donde existe una única fuente de ruido suele ser conocido como un sistema 1:1:1.

2. Según la naturaleza de la señal de ruido:
 - Sistemas de banda ancha: son los sistemas especialmente diseñados para actuar sobre señales con contenido espectral en toda la banda de trabajo (normalmente ruido aleatorio). Estos sistemas, en general, tratan a la señal perturbadora como si fuera ruido aleatorio independientemente de la naturaleza de dicha señal, por lo que no explotan la información adicional que podamos conocer sobre la misma.
 - Sistemas de banda estrecha: son los sistemas destinados a actuar sobre señales de ruido periódico o de banda estrecha. Estos sistemas aprovechan el conocimiento de la naturaleza de este tipo de señales de ruido para actuar sobre ellas de forma mucho más eficiente que los sistemas de banda ancha. Aunque a priori parecen mucho más limitados que los sistemas de banda ancha capaces de actuar sobre toda clase de ruidos, el campo de aplicación de estos sistemas de banda estrecha es bastante interesante, pues son muchos los dispositivos electromecánicos (motores, transformadores ...) que generan ruido de esta naturaleza.
3. Según el modo de actuación sobre la señal de ruido:
 - Sistemas canceladores: son los que tienen como finalidad intentar minimizar (idealmente, cancelar) algún parámetro de la señal de ruido existente (normalmente su nivel de potencia).
 - Sistemas ecualizadores: son mucho más versátiles que los canceladores, puesto que ofrecen la posibilidad de ecualizar o conformar el campo acústico final.
4. Según la zona de actuación:

- Sistemas de control local: son los sistemas que tratan de actuar en una zona localizada dentro de un recinto y en el entorno de los sensores de error.
 - Sistemas de control global: tratan de controlar el campo acústico existente en el interior de un recinto tridimensional, actuando sobre unos pocos modos dominantes y mediante una distribución estratégica de los sensores de error y las fuentes secundarias por todo el recinto. Sobre todo son eficientes a bajas frecuencias.
5. Según la señal de referencia empleada:
- Sistemas *feedforward*: son los que necesitan una señal de referencia correlada con el ruido primario e independiente de la señal de error. Estos sistemas son los más robustos. Hay que diferenciar entre los sistemas *feedforward* puros, que son aquellos donde la señal de referencia no es susceptible de realimentación, y los sistemas que sí son susceptibles de realimentación, por usar, por ejemplo, un micrófono para recoger la señal de referencia y estar al alcance de las fuentes secundarias.
 - Sistemas *feedback*: obtienen la señal de referencia necesaria para su buen funcionamiento de la misma señal de error. Son sistemas más sencillos pero también más sensibles a inestabilidades producidas por las posibles realimentaciones positivas del sistema.
6. Según la naturaleza de las señales en el proceso de control:
- Analógicos: cuando en todo momento las señales que intervienen en el sistema de control son analógicas (así como el propio sistema de control).
 - Digitales: cuando las señales que intervienen en el proceso de control se muestrean y discretizan pasando a operar en el dominio digital, siendo por tanto el sistema de control un sistema definido en el dominio discreto.
7. Según la flexibilidad del proceso de control:
- Fijos: cuando el sistema de generación de las señales, después de haberse ajustado mediante algún método conocido o una etapa de *setup*, permanece invariante durante el periodo de control.
 - Adaptativos: cuando el sistema usa algún tipo de algoritmo o proceso adaptativo que utiliza para la generación de señales durante el proceso de control.

En este trabajo nos centraremos en los sistemas CAR multicanal de control local *feedforward*, digitales y adaptativos, destinados a cancelar ruido de banda ancha.

Por tanto, a continuación se describirá brevemente el funcionamiento de los filtros adaptativos, así como los algoritmos clásicos de control de los mismos.

2.3. Sistemas adaptativos

Un sistema adaptativo, como su propio nombre indica, es un sistema capaz de modificar sus características para adaptarse a la consecución de ciertos objetivos para los que está en un principio diseñado. Se basa en los filtros digitales adaptativos, que tienen la propiedad de actualizar continuamente sus coeficientes en función de una serie de datos de entrada y controlados por un algoritmo desarrollado para tal fin.

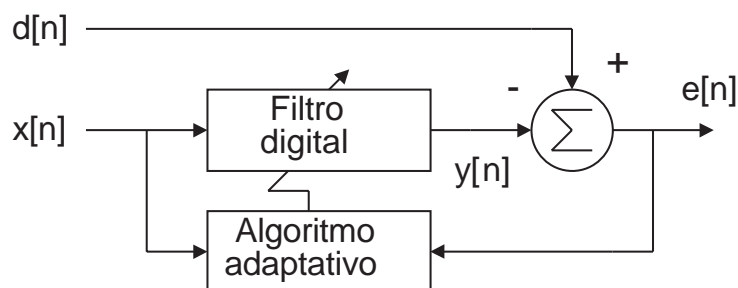


Figura 2.4: Sistema adaptativo típico.

El esquema típico de un sistema adaptativo se muestra en la figura 2.4 donde al margen del filtro adaptativo y del algoritmo que lo controla, podemos apreciar la nomenclatura clásica de las señales que intervienen en dichos sistemas, relatada a continuación:

- $d[n]$: señal deseada.
- $x[n]$: Señal de entrada al filtro adaptativo (o de referencia).
- $y[n]$: señal generada (salida del filtro adaptativo).

- $e[n]$: señal de error diferencia entre la señal deseada y la generada. Sobre esta señal suelen actuar los sistemas adaptativos minimizando alguno de sus parámetros.

Respecto al filtro digital hay que destacar que pueden usarse tanto filtros de respuesta impulsional infinita (IIR) como los de respuesta impulsional finita (FIR), aunque debido a que los IIR no siempre son estables, pues presentan una estructura recursiva que se traduce en la introducción de polos en su función de transferencia, se suelen usar filtros FIR que se caracterizan por tener un comportamiento siempre estable y que pueden realizarse mediante una sencilla estructura de filtro transversal.

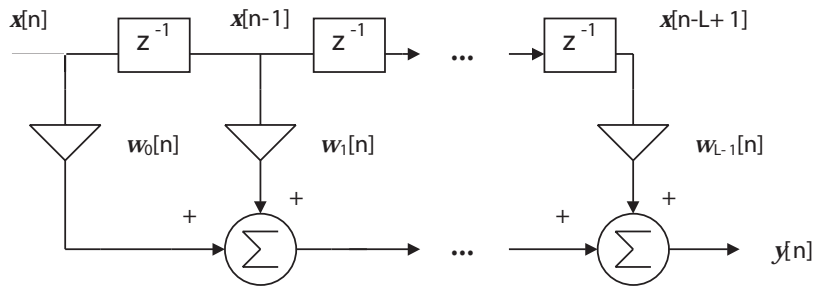


Figura 2.5: Estructura de filtro transversal (FIR).

La salida de un filtro transversal de L componentes como el de la figura 2.5 vendrá dada por la expresión (2.1):

$$y[n] = \sum_{i=0}^{L-1} w_i[n]x[n-i]. \quad (2.1)$$

Normalmente se suele usar notación matricial y de esta forma se puede reescribir la expresión (2.1) como un producto escalar:

$$y[n] = \mathbf{x}_L^T[n] \mathbf{w}_L[n] = \mathbf{w}_L^T[n] \mathbf{x}_L[n] \quad (2.2)$$

donde $\mathbf{x}_L[n]$ es un vector columna que contiene las L muestras de entrada de un filtro transversal de L coeficientes en el instante n y $\mathbf{w}_L[n]$ es el vector de pesos del filtro en ese mismo instante.

$$\mathbf{x}_L[n] = (x[n], x[n-1], x[n-2], \dots, x[n-L+1])^T \quad (2.3)$$

$$\mathbf{w}_L[n] = (w_0[n], w_1[n], w_2[n], \dots, w_{L-1}[n])^T. \quad (2.4)$$

El vector de coeficientes $\mathbf{w}_L[n]$ se actualiza periódicamente, controlado por el algoritmo adaptativo, el cual normalmente suele ser programado para minimizar algún parámetro de la señal de error (generalmente relacionado con su potencia) que se denomina función de coste. Es decir, lo que normalmente intenta conseguir el algoritmo adaptativo es tratar de encontrar los coeficientes óptimos del filtro adaptativo que minimizan algún parámetro de la potencia de la señal de error. Estos coeficientes se denominan coeficientes del filtrado óptimo.

Aunque la estructura de filtrado transversal (introducida por *Kalman* en 1940 [16]) es la usada más frecuentemente, no es la única puesto que también puede usarse la estructura en celosía (*lattice*), introducida por Itakura y Saito en 1972 [17], o la estructura de vector sistólico (*systolic array*) introducida por Kung y Leiserson en 1978, [18].

Hay que hacer notar que los sistemas adaptativos son sobre todo útiles en dos circunstancias:

- Cuando las características del filtro a usar deba variar con el tiempo porque la señal a filtrar así lo necesite.
- Cuando no sepamos qué filtro debemos usar a priori y por tanto necesitemos un proceso de adaptación hasta dar con el adecuado.

Estos sistemas se ajustan por tanto a las necesidades del control activo de ruido donde inicialmente no sabemos cuál es el filtro que realiza el filtrado óptimo (por lo tanto estamos ante un caso de no saber que filtro debemos usar), y posteriormente debemos tener un sistema lo suficientemente flexible como para que se readapte ante las posibles variaciones de la señal de ruido a minimizar o ante cambios en las condiciones del entorno donde se ejerce el control (caso de filtro variante).

Para el cálculo de los coeficientes del filtro adaptativo en el esquema mostrado en la figura 2.4, habitualmente se usan dos estrategias diferenciadas:

- Aproximación de gradiente estocástico: en este caso, la función de coste a minimizar es el error cuadrático medio ($\xi = E\{|e[n]|^2\}$), donde el operador $E\{\cdot\}$ denota el valor medio o esperado. Se podría obtener una solución directa si se conocieran las propiedades estadísticas de las señales que intervienen en el sistema aunque es muy frecuente calcular una solución recursiva, como en los algoritmos de gradiente de paso

mínimo de entre los que podemos destacar el LMS [19]. Los algoritmos de gradiente estocástico suelen usar la estructura de filtrado transversal.

- Estimación de mínimos cuadrados: en este caso, la función de coste es la suma del error cuadrático $\left(\xi[n] = \sum_{i=1}^n e[i]^2\right)$ al que generalmente se le añade una ponderación (o constante λ) llamada factor de olvido para dar diferente peso a las señales recientes con respecto de las más antiguas. De esta forma, la función de coste más general puede escribirse como:

$$\xi[n] = \sum_{i=1}^n \lambda^{n-i} e[i]^2. \quad (2.5)$$

Para obtener los coeficientes que proporcionan la solución al problema podemos trabajar por bloques de señal, o bien mediante algoritmos recursivos muestra a muestra con menor coste computacional. En esta segunda opción nos encontramos con las siguientes posibilidades:

- RLS estándar: como puede comprobarse en [2] se basa en el lema de inversión matricial. Usa una estructura transversal y posee alta complejidad computacional y poca robustez.
- Algoritmos de raíz cuadrada: son más estables y robustos y se basan en la descomposición QR [20].
- Algoritmos *Fast* RLS: tratan de minimizar el coste computacional aprovechando la estructura de la matriz de datos de entrada, usando filtros de predicción lineal [21]. Si usan estructuras de filtros transversales obtendremos la familia de algoritmos FTF (*Fast Transversal Filters*) aunque también pueden usarse estructuras en celosía para los filtros predictores (filtros adaptativos de *order-recursive*).

Los algoritmos de gradiente estocástico y de mínimos cuadrados han sido las soluciones clásicas usadas históricamente en los algoritmos de control activo de ruido cuando se dan condiciones de estacionariedad en el sistema. Además de estas estrategias, también podemos encontrar los algoritmos basados en el filtrado de *Kalman* [22], que proporcionan una estimación óptima por mínimos cuadrados tanto en condiciones de estacionariedad como de no estacionariedad. Todos ellos se describirán con detalle a continuación, pues son el punto de partida de la mayoría de las estrategias adaptativas.

2.4. Algoritmos de gradiente estocástico

Los algoritmos de gradiente estocástico tratan de encontrar los coeficientes del filtro transversal que minimizan el error cuadrático medio de la señal $e[n]$ que aparece en la figura 2.4. Para ello, intentan acercarse a la solución de una forma recursiva siguiendo la dirección del gradiente de lo que se conoce como la función de error o curva de superficie de error. El algoritmo más ampliamente conocido y usado de esta familia es el LMS (*least mean square*) que será descrito posteriormente.

2.4.1. Coeficientes que realizan el filtrado óptimo

Según se puede comprobar en la figura 2.4 la señal de error puede escribirse como:

$$e[n] = d[n] - y[n] = d[n] - \mathbf{w}_L^T[n] \mathbf{x}_L[n]. \quad (2.6)$$

Vamos a tratar de encontrar cuáles son los coeficientes \mathbf{w}_L del filtro que minimizan el valor cuadrático medio de la señal de error, y que por tanto realizan el filtrado óptimo. Para ello partimos de la expresión de dicho valor cuadrático medio:

$$\xi = E\{e[n]^2\} = E\{d[n]^2\} + \mathbf{w}_L^T E\{\mathbf{x}_L[n] \mathbf{x}_L^T[n]\} \mathbf{w}_L - 2E\{d[n] \mathbf{x}_L^T[n]\} \mathbf{w}_L. \quad (2.7)$$

Para alcanzar el resultado expuesto en (2.6), se ha considerado que los elementos del vector $\mathbf{x}_L[n]$ son independientes entre sí, al igual que de las muestras $\mathbf{d}[n]$, y que los coeficientes del filtro adaptativo varían muy lentamente en el tiempo hasta el punto de poder considerarlos como constantes (aunque esto, como ya se sabe, no es del todo correcto, y lo que pasa es que se readaptan buscando el mínimo de la función de error). Considerando también, que las secuencias formadas por las muestras de $x[n]$ y $d[n]$ son estacionarias, podemos denominar \mathbf{R}_L a la matriz de autocorrelación de las muestras de la señal de entrada $x[n]$ y \mathbf{p} al vector de correlación cruzada entre la señal de entrada y la señal $d[n]$, con lo que podemos reescribir la ecuación (2.7) de la forma:

$$\xi = E\{e[n]^2\} = E\{d[n]^2\} + \mathbf{w}_L^T \mathbf{R}_L \mathbf{w}_L - 2\mathbf{p}^T \mathbf{w}_L \quad (2.8)$$

donde:

$$\mathbf{R}_L = E\{\mathbf{x}_L[n] \mathbf{x}_L^T[n]\}$$

$$= E \begin{pmatrix} x[n]^2 & x[n]x[n-1] & \cdots & x[n]x[n-L+1] \\ x[n-1]x[n] & x[n-1]^2 & \cdots & x[n-1]x[n-L+1] \\ \vdots & \vdots & \ddots & \vdots \\ x[n-L+1]x[n] & x[n-L+1]x[n-1] & \cdots & x[n-L+1]^2 \end{pmatrix} \quad (2.9)$$

y \mathbf{p} se define como:

$$\begin{aligned} \mathbf{p} &= E\{d[n]\mathbf{x}_L[n]\} \\ &= E\{d[n]x[n], d[n]x[n-1], d[n]x[n-2], \dots, d[n]x[n-L+1]\}^T. \end{aligned} \quad (2.10)$$

Como puede comprobarse en la ecuación (2.8), el error cuadrático medio es una función de los coeficientes del filtro transversal \mathbf{w}_L . La curva que representa dicha función se conoce como superficie de error y se trata de un hiperparaboloide elíptico del que se pueden encontrar los valores de los coeficientes que logran minimizar dicha función. Dichos valores son los que proporcionan el filtrado óptimo y se denominarán como \mathbf{w}_{Lopt} .

Puesto que la superficie de error es una función cuadrática, sabemos que ha de existir un punto crítico y que debe ser un mínimo. El valor de ese mínimo y por tanto de los coeficientes \mathbf{w}_{Lopt} que proporcionan el filtrado óptimo se obtiene a partir del gradiente de la superficie de error.

$$\nabla \xi = \frac{\partial E\{e[n]^2\}}{\partial \mathbf{w}_L[n]} = 2\mathbf{R}_L \mathbf{w}_L - 2\mathbf{p}. \quad (2.11)$$

El vector de coeficientes que consigue realizar el filtrado óptimo se puede hallar calculando los coeficientes que consiguen anular el valor del gradiente, obteniendo:

$$\mathbf{w}_{Lopt} = \mathbf{R}_L^{-1} \mathbf{p}. \quad (2.12)$$

Ya sabemos, por tanto, cuáles son los coeficientes que proporcionan el filtrado óptimo. El valor mínimo del error cuadrático medio de la función de error (valor de dicha función cuando se realiza el filtrado óptimo) vendrá dado por la siguiente expresión:

$$\xi_{min} = E\{e[n]^2\}_{|\mathbf{w}_L=\mathbf{w}_{Lopt}} = E\{d[n]^2\} - \mathbf{p}^T \mathbf{w}_{Lopt}. \quad (2.13)$$

En la figura 2.6 se puede comprobar la forma típica que tendría la superficie de error para el caso de un filtro de dos coeficientes.

2.4.2. Algoritmos de gradiente

Como se ha comentado en el punto anterior, la superficie de error es una función cuadrática de los coeficientes del filtro adaptativo cuyo mínimo puede

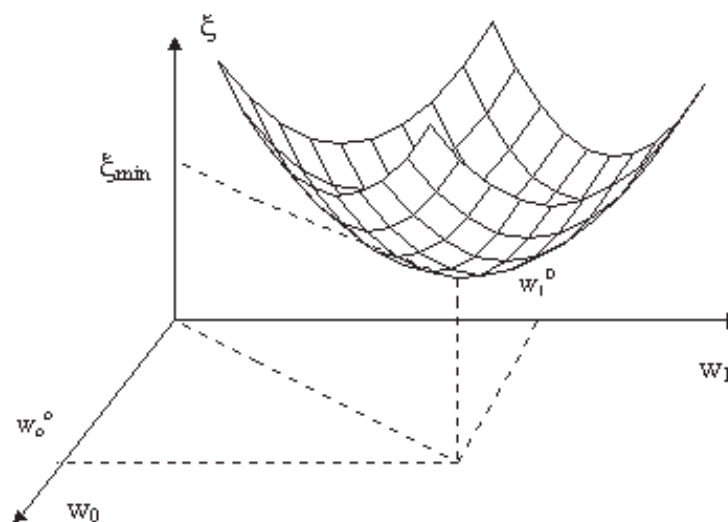


Figura 2.6: Superficie de error.

hallarse a partir del cálculo de los coeficientes que realizan el filtrado óptimo y que, como ha quedado reflejado en la ecuación (2.12) dependen de las características estadísticas de las señales formadas por las secuencias $x[n]$ y $d[n]$.

Pero normalmente no podemos calcular los coeficientes que realizan el filtrado óptimo a partir de estos conocimientos puesto que en la mayoría de los casos no se pueden conocer a priori tales parámetros estadísticos, por lo que hay que hacer uso de estrategias alternativas como los métodos basados en el gradiente.

Los métodos de gradiente son algoritmos iterativos que intentan calcular el mínimo de una determinada función desplazándose por la misma según la dirección de máxima variación (dirección del gradiente) y en sentido opuesto a la misma.

Estos métodos, necesitan por tanto el conocimiento del gradiente de la función de error, aunque según se puede comprobar en la ecuación (2.11) dicho gradiente depende igualmente de los parámetros estadísticos de las señales $x[n]$ y $d[n]$.

Realmente, lo que se suele usar en este tipo de métodos iterativos de gradiente, no es el gradiente propiamente dicho, sino una estimación del mismo ($\hat{\nabla}$), y dependiendo de dicha estimación se pueden distinguir diferentes métodos de gradiente. Sin embargo, todos ellos tienen en común un cálculo de los

pesos de forma iterativa [19] tal y como se expone en la siguiente ecuación:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] - \frac{\mu}{2} \widehat{\nabla} \xi[n], \quad (2.14)$$

siendo μ un factor de convergencia conocido como paso del algoritmo y cuyo valor determina la velocidad de convergencia. Una vez alcanzado el mínimo, entonces $\mathbf{w}_L[n] \approx \mathbf{w}_{Lopt}$ y $\widehat{\nabla} \xi \approx 0$, con lo que la actualización de pesos se detiene.

El principal inconveniente de esta estrategia es su baja velocidad de convergencia cuando existe alta correlación entre los datos de la señal de referencia, aunque existen estrategias que permiten mejorar esta deficiencia como la presentada en [23].

2.4.3. Algoritmo LMS (*least mean square*)

El algoritmo de gradiente de uso más extendido es el LMS [19]. Dicho algoritmo usa el error cuadrático instantáneo ($e[n]^2$) como aproximación del error cuadrático medio en la estimación del gradiente.

$$\xi[n] = E\{e[n]^2\} \approx e[n]^2 = \widehat{\xi}[n] = (d[n] - \mathbf{x}_L^T[n] \mathbf{w}_L[n])^2. \quad (2.15)$$

De esta forma podemos escribir:

$$\nabla \xi = \frac{\partial E\{e[n]^2\}}{\partial \mathbf{w}_L[n]} \approx \frac{\partial e[n]^2}{\partial \mathbf{w}_L[n]} = 2 \frac{\partial e[n]}{\partial \mathbf{w}_L[n]} e[n] = -2e[n] \mathbf{x}_L^T[n]. \quad (2.16)$$

Podemos comprobar cómo, en la estimación usada, solo intervienen los valores de las muestras de entrada al filtro y el valor de la señal de error, lo que demuestra la sencillez del método.

La actualización de pesos usando el algoritmo LMS quedaría:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu e[n] \mathbf{x}_L[n]. \quad (2.17)$$

La forma práctica en la que se podría realizar el algoritmo LMS en un sistema adaptativo podría ser:

1. Obtener las muestras de la señal de entrada $x[n]$ y $d[n]$.
2. Actualizar el vector de entrada $\mathbf{x}_L[n]$.
3. Calcular la salida $y[n] = \mathbf{w}_L^T[n-1] \mathbf{x}_L[n]$.
4. Calcular la señal $e[n] = d[n] - y[n]$.

5. Actualizar los pesos $\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + 2\mu e[n]\mathbf{x}_L[n]$.

Este algoritmo necesitaría solo $2L + 1$ multiplicaciones por iteración.

La estabilidad, convergencia y propiedades en régimen estacionario del LMS dependen del número de coeficientes seleccionado para el filtro adaptativo, de la constante de convergencia y propiedades estadísticas de la señal de entrada al filtro o de referencia.

Existen muchas versiones y variantes de este algoritmo que tratan de mejorar algunas de sus prestaciones, ya sea la complejidad computacional, la velocidad de convergencia o cualquier otra de sus propiedades. De entre estas versiones, se pueden destacar el algoritmo LMS normalizado [19] (el cual se describirá con detalle al exponer el algoritmo de proyección afín, que es el eje de este trabajo), el algoritmo LMS de paso variable, donde el paso de convergencia se reajusta en cada iteración [24], el algoritmo LMS con desvanecimiento [25], donde se introduce un factor que pondera el peso de los coeficientes antiguos en el cálculo de los nuevos, los algoritmos LMS de bloque [26], y muchos más.

2.5. Algoritmos de mínimos cuadrados

Dado el sistema adaptativo presentado en la figura 2.4 donde $d[n]$ es el valor de la muestra en el instante n -ésimo de una señal desconocida que llamaremos señal deseada y $x[n]$ el valor de la muestra en el instante n -ésimo de una señal conocida y correlada con la anterior que llamaremos señal de referencia, trataremos de generar la señal deseada a partir de la señal de referencia haciendo uso de un filtro FIR adaptativo. Los coeficientes óptimos de dicho filtro FIR se calculan minimizando una determinada función de coste relacionada con la señal de error.

Como ya se ha comentado, los algoritmos adaptativos clásicos suelen usar como función de coste o bien una aproximación del valor cuadrático medio de la señal de error (para el caso de los algoritmos que usan aproximaciones de gradiente estocástico, y sus versiones iterativas basadas en el método de gradiente entre los que se encuentra el algoritmo LMS comentado anteriormente), o la suma de los cuadrados de dicha señal de error, dando lugar a los algoritmos de estimación de mínimos cuadrados de entre los que podemos destacar el RLS (*recursive least square*).

Veremos a continuación cómo se resuelve el problema mediante mínimos cuadrados.

2.5.1. Minimización mediante mínimos cuadrados (LS)

Si llamamos $y[n]$ a la salida de nuestro filtro $\mathbf{w}_L[n]$ de L coeficientes, tendremos:

$$y[n] = \sum_{i=0}^{L-1} w_i[n]x[n-i] = \mathbf{w}_L^T[n]\mathbf{x}_L[n] \quad (2.18)$$

donde $\mathbf{x}_L[n]$ y $\mathbf{w}_L[n]$ representan los vectores ya definidos en (2.3) y (2.4), respectivamente.

Supongamos que en el instante n -ésimo conocemos los valores de la señal de referencia y de la señal deseada desde su origen (podemos asumir que se cumple la hipótesis de preeventanado, que en este caso implica que para instantes de tiempo anteriores a cero ($n < 0$), las señales tienen valor cero). Lo que se pretende minimizar mediante mínimos cuadrados es la suma total del error cuadrático cometido, o lo que es lo mismo:

$$\xi[n] = \sum_{i=0}^n e^2[i] \quad (2.19)$$

siendo:

$$e[i] = d[i] - \mathbf{w}_L^T[n]\mathbf{x}_L[i]. \quad (2.20)$$

Como se puede apreciar en la ecuación (2.20), vamos a suponer que los coeficientes del filtro, no varían con el tiempo, sino que permanecen invariantes hasta la iteración n -ésima. Esta condición no se suele cumplir en sistemas adaptativos, pero se asume con frecuencia puesto que simplifica mucho las operaciones a realizar y es justificable cuando se da una variación lenta de los coeficientes. En realidad, el problema matemático que intentaremos resolver difiere un poco del problema práctico, puesto que en este último vamos a tratar de encontrar los coeficientes del filtro de forma adaptativa, mientras que matemáticamente hablando lo que tratamos de resolver es qué coeficientes fijos desde la primera iteración minimizarían la señal de error total hasta la iteración n -ésima.

Operando con la expresión (2.19), obtenemos:

$$\xi[n] = \left(\sum_{i=0}^n d^2[i] \right) + \mathbf{w}_L^T[n] \left(\sum_{i=0}^n \mathbf{x}_L[i]\mathbf{x}_L^T[i] \right) \mathbf{w}_L[n] - 2\mathbf{w}_L^T[n] \left(\sum_{i=0}^n d[i]\mathbf{x}_L[i] \right). \quad (2.21)$$

Para simplificar la notación, llamamos:

$$\mathbf{R}_L[n] = \sum_{i=0}^n \mathbf{x}_L[i]\mathbf{x}_L^T[i] = \mathbf{X}_L^T[n]\mathbf{X}_L[n] \quad (2.22)$$

y

$$\mathbf{r}_L[n] = \sum_{i=0}^n \mathbf{x}_L[i] d[i] = \mathbf{X}_L^T[n] \mathbf{d}_L[n] \quad (2.23)$$

donde

$$\mathbf{X}_L[n] = \begin{pmatrix} x(0) & 0 & 0 & 0 \\ x(1) & x(0) & 0 & 0 \\ \cdot & x(1) & \dots & 0 \\ \cdot & & \dots & x(0) \\ \cdot & & \dots & \cdot \\ x(n) & x(n-1) & \dots & x(n-L+1) \end{pmatrix} \quad (2.24)$$

y $\mathbf{d}_L[n] = [d[n], d[n-1], \dots, d[n-L+1]]^T$.

Entonces, podemos reescribir la ecuación (2.21) como:

$$\xi[n] = \left(\sum_{i=0}^n d^2[i] \right) + \mathbf{w}_L^T[n] \mathbf{R}_L[n] \mathbf{w}_L[n] - 2\mathbf{w}_L^T[n] \mathbf{r}_L[n]. \quad (2.25)$$

Derivando $\xi[n]$ con respecto $\mathbf{w}_L[n]$ obtenemos:

$$\frac{\partial(\xi[n])}{\partial \mathbf{w}_L[n]} = 2\mathbf{R}_L[n] \mathbf{w}_L[n] - 2\mathbf{r}_L[n], \quad (2.26)$$

e igualando a cero, obtenemos que los valores de los coeficientes $\mathbf{w}_L[n]$ que minimizan la suma del error cuadrático total satisfacen las siguientes ecuaciones:

$$\mathbf{R}_L[n] \mathbf{w}_{Lopt}[n] = \mathbf{r}_L[n] \quad (2.27)$$

$$\mathbf{w}_{Lopt}[n] = \mathbf{R}_L[n]^{-1} \mathbf{r}_L[n], \quad (2.28)$$

obteniendo para el valor mínimo de dicho error:

$$\begin{aligned} \xi[n]_{min} &= \left(\sum_{i=0}^n d^2[i] \right) + \mathbf{w}_{Lopt}^T[n] \mathbf{R}_L[n] \mathbf{w}_{Lopt}[n] - 2\mathbf{w}_{Lopt}^T[n] \mathbf{r}_L[n] \\ &= \left(\sum_{i=0}^n d^2[i] \right) - \mathbf{w}_{Lopt}^T[n] \mathbf{R}_L[n] \mathbf{w}_{Lopt}[n] = \left(\sum_{i=0}^n d^2[i] \right) - \mathbf{w}_{Lopt}^T[n] \mathbf{r}_L[n]. \end{aligned} \quad (2.29)$$

El problema presentado resuelve, por tanto, el cálculo de los coeficientes en la iteración n -ésima que minimizan el error cuadrático total acumulado entre una señal deseada y una señal generada a partir de otra señal conocida y correlada con la original de la cual se disponen de sus muestras hasta el instante n -ésimo así como de esas mismas muestras de la señal deseada. Sería interesante plantear cómo obtener los coeficientes óptimos en la iteración n +

1 a partir del conocimiento de dichos coeficientes en la iteración n -ésima y conocidas también las nuevas muestras de la señal deseada y de referencia en dicha iteración, o sea, en el instante $n + 1$ ($d[n + 1]$ y $x[n + 1]$). Para ello, tenemos que tener en cuenta las siguientes relaciones fáciles de demostrar a partir de las ecuaciones (2.22) y (2.23):

$$\mathbf{R}_L[n + 1] = \mathbf{R}_L[n] + \mathbf{x}_L[n + 1]\mathbf{x}_L^T[n + 1] \quad (2.30)$$

$$\mathbf{r}_L[n + 1] = \mathbf{r}_L[n] + \mathbf{x}_L[n + 1]d[n + 1]. \quad (2.31)$$

Considerando que los coeficientes del filtro obtenidos por mínimos cuadrados en la iteración $n + 1$ vendrán dados por la siguiente ecuación:

$$\mathbf{R}_L[n + 1]\mathbf{w}_{Lopt}[n + 1] = \mathbf{r}_L[n + 1], \quad (2.32)$$

despejando de (2.30) y (2.31) las expresiones de $\mathbf{R}_L[n]$ y $\mathbf{r}_L[n]$, y sustituyendo en (2.27) tendremos:

$$(\mathbf{R}_L[n + 1] - \mathbf{x}_L[n + 1]\mathbf{x}_L^T[n + 1])\mathbf{w}_{Lopt}[n] = \mathbf{r}_L[n + 1] - \mathbf{x}_L[n + 1]d[n + 1]. \quad (2.33)$$

Operando con dicha expresión, se obtiene:

$$\mathbf{R}_L[n + 1]\mathbf{w}_{Lopt}[n] + \mathbf{x}_L[n + 1](d[n + 1] - \mathbf{x}_L^T[n + 1]\mathbf{w}_{Lopt}[n]) = \mathbf{r}_L[n + 1], \quad (2.34)$$

donde aparece la cantidad conocida como error *a priori* (para diferenciarlo del error *a posteriori* definido en (2.20) que es el que normalmente se intenta minimizar), que se define como:

$$e_{pri}[n + 1] = d[n + 1] - \mathbf{x}_L^T[n + 1]\mathbf{w}_{Lopt}[n] \quad (2.35)$$

pudiendo reescribir la ecuación (2.34) como:

$$\mathbf{R}_L[n + 1]\mathbf{w}_{Lopt}[n] + \mathbf{x}_L[n + 1]e_{pri}[n + 1] = \mathbf{r}_L[n + 1]. \quad (2.36)$$

Si definimos el vector de longitud L , $\mathbf{G}_L^*[n + 1]$ (al que llamaremos vector de ganancia de *Kalman* igual que en el anexo B o en [27]) como aquel que cumple la relación:

$$\mathbf{R}_L[n + 1]\mathbf{G}_L^*[n + 1] = \mathbf{x}_L[n + 1] \quad (2.37)$$

e introducimos la ecuación (2.37) en (2.36), por comparación con la ecuación (2.32) obtenemos:

$$\mathbf{w}_{Lopt}[n + 1] = \mathbf{w}_{Lopt}[n] + \mathbf{G}_L^*[n + 1]e_{pri}[n + 1]. \quad (2.38)$$

El hecho del que el vector $\mathbf{G}_L^*[n + 1]$ reciba el nombre de ganancia de *Kalman*, se justifica intuitivamente si consideramos la ecuación (2.38) como la

que rige el valor estimado de un estado (en este caso serían nuestros coeficientes $\mathbf{w}_{Lopt}[n+1]$) a partir de una predicción ($\mathbf{w}_{Lopt}[n]$, en esta ocasión) según el filtrado de *Kalman*, que aplica una corrección gobernada por el vector de ganancia de *Kalman*. En este caso, dicho vector se elige de forma que el valor medio del valor estimado tienda a los coeficientes del filtrado óptimo expresados en (2.32).

Sin embargo, la recursión mostrada en (2.38) no suele usarse con mucha frecuencia, ya que el cálculo del vector de ganancia de *Kalman* lleva implícita una inversión matricial que a menudo se evita como veremos en el algoritmo de mínimos cuadrados recursivos (RLS).

Si sustituimos en la expresión (2.38) el valor del vector de ganancia de *Kalman*, obtendremos una expresión de actualización de los coeficientes muy parecida a la que se obtiene para los filtros de proyección afín como veremos más adelante. Esta expresión sería:

$$\mathbf{w}_{Lopt}[n+1] = \mathbf{w}_{Lopt}[n] + \mathbf{R}_L^{-1}[n+1]\mathbf{x}_L[n+1]e_{pri}[n+1]. \quad (2.39)$$

También nos puede ser útil obtener una relación recursiva para el mínimo error total expresado en (2.29). Usando dicha expresión, podemos decir que:

$$\xi[n+1]_{min} = \left(\sum_{i=0}^{n+1} d^2[i] \right) - \mathbf{w}_{Lopt}^T[n+1]\mathbf{r}_L[n+1] \quad (2.40)$$

que, haciendo uso de (2.38) puede escribirse como:

$$\begin{aligned} \xi[n+1]_{min} &= \left(\sum_{i=0}^n d^2[i] \right) + d^2[n+1] \\ &\quad - \mathbf{w}_{Lopt}^T[n]\mathbf{r}_L[n+1] - \mathbf{r}_L^T[n+1]\mathbf{G}_L^*[n+1]e_{pri}[n+1]. \end{aligned} \quad (2.41)$$

Teniendo en cuenta la ecuación (2.31) e introduciendo el valor del error *a priori* definido en (2.35), se puede reescribir el término de la parte derecha de la expresión (2.41) como:

$$\begin{aligned} &\left(\sum_{i=0}^n d^2[i] \right) + d^2[n+1] - \mathbf{w}_{Lopt}^T[n]\mathbf{r}_L[n+1] \\ &= \left(\sum_{i=0}^n d^2[i] \right) + d[n+1]d[n+1] - \mathbf{w}_{Lopt}^T[n]\mathbf{r}_L[n] \\ &\quad - \mathbf{w}_{Lopt}^T[n]\mathbf{x}_L[n+1]d[n+1] \\ &= \left(\sum_{i=0}^n d^2[i] \right) - \mathbf{w}_{Lopt}^T[n]\mathbf{r}_L[n] + d[n+1](d[n+1] - \mathbf{w}_{Lopt}^T[n]\mathbf{x}_L[n+1]) \\ &= \xi[n]_{min} + d[n+1]e_{pri}[n+1]. \end{aligned} \quad (2.42)$$

Introduciendo esta última relación en la ecuación (2.41), obtenemos:

$$\begin{aligned}
\xi[n+1]_{min} &= \xi[n]_{min} + d[n+1]e_{pri}[n+1] - \mathbf{r}_L^T[n+1]\mathbf{G}_L^*[n+1]e_{pri}[n+1] \\
&= \xi[n]_{min} + e_{pri}[n+1](d[n+1] - \mathbf{r}_L^T[n+1]\mathbf{G}_L^*[n+1]) \\
&= \xi[n]_{min} + e_{pri}[n+1](d[n+1] - \mathbf{r}_L^T[n+1]\mathbf{R}_L^{-1}[n+1]\mathbf{R}_L[n+1]\mathbf{G}_L^*[n+1]) \\
&= \xi[n]_{min} + e_{pri}[n+1](d[n+1] - \mathbf{w}_{Lopt}^T[n+1]\mathbf{x}_L[n+1])
\end{aligned} \tag{2.43}$$

donde se ha hecho uso de (2.32) y (2.37). Finalmente, si nos damos cuenta de que el valor que aparece entre paréntesis no es más que el valor del error *a posteriori* (2.20), podemos escribir de una forma compacta que:

$$\xi[n+1]_{min} = \xi[n]_{min} + e_{pri}[n+1]e[n+1]. \tag{2.44}$$

Cálculo eficiente de la matriz $\mathbf{R}_L[n]$

Llegado a este punto, merece la pena detenerse en el cálculo recursivo eficiente de la matriz $\mathbf{R}_L[n]$, puesto que matrices con estructura similar han de obtenerse en los algoritmos de proyección afín posteriormente estudiados. La matriz $\mathbf{R}_L[n]$ puede calcularse como se expresa en la fórmula (2.22) y más eficientemente según (2.30). Dicha matriz, representa una especie de matriz de autocorrelación instantánea de la señal de referencia. Su estructura está detallada en (2.45).

$$\mathbf{R}_L[n] = \begin{pmatrix} R_x(n, n) & R_x(n, n-1) & \dots & R_x(n, n-L+1) \\ R_x(n-1, n) & R_x(n-1, n-1) & \dots & R_x(n-1, n-L+1) \\ \vdots & \ddots & & \vdots \\ R_x(n-L+1, n) & R_x(n-L+1, n-1) & \dots & R_x(n-L+1, n-L+1) \end{pmatrix} \tag{2.45}$$

siendo:

$$R_x(i, j) = \mathbf{x}[i]^T \mathbf{x}[j], \tag{2.46}$$

con $\mathbf{x}[k]$ igual al vector columna de tamaño n con las k muestras de la señal de referencia y el resto de valores igual a cero tal y como se describe en (2.47)

$$\mathbf{x}[k] = (0, \dots, 0, x[0], x[1], \dots, x[k-1])^T. \tag{2.47}$$

De la expresión (2.46) podemos deducir que la matriz $\mathbf{R}_L[n]$ es una matriz simétrica. Sin embargo, si observamos atentamente la estructura de dicha matriz, podemos notar que la mayoría de los datos de una iteración se repiten a la siguiente desplazando su posición. Por ello, no es necesario calcular todos

los datos de dicha matriz en la iteración n -ésima, si disponemos de sus valores en las iteración anterior, sino que obtendríamos máxima eficiencia calculando solamente aquellos valores novedosos y recolocando aquellos que cambian su posición. Así, si llamamos $\bar{R}_{L-1}[n]$ a la matriz $((L-1) \times (L-1))$ que contiene la parte superior izquierda de la matriz $R_L[n]$, entonces, podemos poner que:

$$\mathbf{R}_L[n+1] = \begin{pmatrix} \mathbf{x}[n+1]^T \mathbf{x}[n+1] & \mathbf{x}_v^T[n+1] \\ \mathbf{x}_v[n+1] & \bar{R}_{L-1}[n] \end{pmatrix}, \quad (2.48)$$

siendo $\mathbf{x}_v[n+1] = \mathbf{x}_v[n] + x(n+1) \cdot \bar{\mathbf{x}}_L[n]$, y $\bar{\mathbf{x}}_L[n]$ los $L-1$ valores más recientes del vector $\mathbf{x}_L[n]$.

De esta forma, solo realizamos $2L-1$ multiplicaciones para el cálculo de la matriz $\mathbf{R}_L[n]$, frente a las L^2 que se necesitarían aplicando (2.30) o las L^2n aplicando (2.22).

2.5.2. Algoritmo RLS (*recursive least square*)

El algoritmo RLS o de mínimos cuadrados recursivos, resuelve el problema de minimización mediante mínimos cuadrados acercándose a la solución iterativamente (siendo más eficiente, computacionalmente hablando, que el LS). Se presenta como una alternativa práctica al clásico LMS para encontrar los coeficientes óptimos en un proceso adaptativo. Tiene la ventaja de que su tasa de convergencia es más rápida que la del LMS pero también es cierto que es a costa de una mayor complejidad computacional. Los orígenes del RLS podemos encontrarlos en los trabajos de Plackett [28], aunque más recientemente otros autores como Hastings-Jame y Sage lo volvieron a reformular con éxito [29]. Una completa descripción de dicho algoritmo puede ser consultada, entre otros, en [2]. Este algoritmo nos conducirá a actualizar eficientemente los coeficientes de los pesos óptimos de forma similar a como se expresa en la ecuación (2.39) pero con un cálculo eficiente de la matriz $\mathbf{R}^{-1}[n]$ sin tener que calcular la matriz $\mathbf{R}[n]$.

Por tanto, el algoritmo RLS, trata de minimizar la función de coste dada por la expresión (2.49)

$$\xi[n] = \sum_{i=0}^n \lambda^{n-i} e^2[i], \quad (2.49)$$

donde se ha introducido el factor λ con respecto a la ecuación (2.19) representando un factor de olvido ($0 < \lambda < 1$) que conforme más pequeño sea, más peso dará a los datos más recientes. Esto es útil cuando la estadística de la señal sufre variaciones a lo largo del tiempo, y necesitamos despreciar los

valores más antiguos que ya no nos dan información útil de la señal actual. En caso de $\lambda = 1$ tendremos la función de coste dada por (2.19) donde todos los errores tienen el mismo peso (o lo que es lo mismo, trataría de minimizar la señal de error incluyendo la totalidad de su historia pasada).

Así, teniendo en cuenta el esquema típico de un sistema adaptativo (figura 2.4), tendríamos para el RLS que la señal de error en el instante i -ésimo vendría dada por (2.50)

$$e[i] = d[i] - \mathbf{w}_L^T[n] \mathbf{x}_L[i], \quad 0 \leq i \leq n \quad (2.50)$$

donde se presupone que el vector de pesos del filtro adaptativo $\mathbf{w}_L[n]$ permanece constante durante el intervalo $1 \leq i \leq n$. Introduciendo la información de (2.50) en (2.49), podemos reescribir:

$$\begin{aligned} \xi[n] = \sum_{i=1}^n \lambda^{n-i} d^2[i] - 2\mathbf{w}_L^T[n] \left(\sum_{i=1}^n \lambda^{n-i} d[i] \mathbf{x}_L[i] \right) \\ + \mathbf{w}_L^T[n] \left(\sum_{i=1}^n \lambda^{n-i} \mathbf{x}_L^T[i] \mathbf{x}_L[i] \right) \mathbf{w}_L[n] \end{aligned} \quad (2.51)$$

que, para simplificar nomenclatura se suele formular:

$$\xi[n] = \sum_{i=1}^n \lambda^{n-i} d^2[i] - 2\rho_L^T[n] \mathbf{w}_L[n] + \mathbf{w}_L^T[n] \Phi_L[n] \mathbf{w}_L[n] \quad (2.52)$$

siendo: $\rho_L[n] = \sum_{i=1}^n \lambda^{n-i} d[i] \mathbf{x}_L[i]$, un vector que hace las veces de la correlación

cruzada entre la señal deseada y la señal de error y $\Phi_L[n] = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}_L^T[i] \mathbf{x}_L[i]$, una matriz que hace el papel de la autocorrelación de los datos de entrada.

Para encontrar los coeficientes óptimos que minimizan la función de coste $\xi[n]$, derivamos la expresión (2.52) respecto a $\mathbf{w}_L[n]$ e igualamos a cero como hicimos en el caso de mínimos cuadrados, obteniendo una solución similar:

$$\mathbf{w}_{Lopt}[n] = \Phi_L[n]^{-1} \rho_L[n]. \quad (2.53)$$

Dado que a menudo las matrices de datos $\rho_L[n]$ y $\Phi_L[n]$ resultan muy costosas de calcular o son desconocidas, el método de RLS evita calcular directamente dichas matrices usando una sencilla relación iterativa para su cálculo tal y como se expresa a continuación:

$$\Phi_L[n] = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}_L^T[i] \mathbf{x}_L[i] + \sum_{i=1}^{n-1} \lambda^{n-i} \mathbf{x}_L^T[i] \mathbf{x}_L[i] + \mathbf{x}_L^T[n] \mathbf{x}_L[n]$$

$$= \lambda \sum_{i=1}^{n-1} \lambda^{[(n-1)-i]} \mathbf{x}_L^T[i] \mathbf{x}_L[i] + \mathbf{x}_L^T[n] \mathbf{x}_L[n] = \lambda \Phi_L[n-1] + \mathbf{x}_L^T[n] \mathbf{x}_L[n]$$

y de la misma forma:

$$\rho_L[n] = \lambda \rho_L[n-1] + d[n] \mathbf{x}_L^T[n]. \quad (2.54)$$

Ya resuelto el problema del cálculo iterativo de las matrices de datos $\rho_L[n]$ y $\Phi_L[n]$, queda por resolver, el cálculo de la inversa de la matriz $\Phi_L^{-1}[n]$ de forma eficiente. Sin embargo, la matriz de autocorrelación suele tener ciertas propiedades (como el hecho de ser definida positiva) que nos va a permitir aplicar el *lema de inversión matricial* que dice:

“Dadas dos matrices definidas positivas \mathbf{A} y \mathbf{B} , de dimensiones $M \times M$ y que cumplen la siguiente relación: $\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H$, donde \mathbf{D} es otra matriz definida positiva, de dimensiones $N \times N$, y \mathbf{C} es una matriz $M \times N$, entonces, la inversa de la matriz \mathbf{A} puede escribirse como: $\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B}$ ”.

El lema de inversión puede aplicarse a $\Phi_L[n]$ realizando las siguientes identificaciones:

$$\begin{aligned} \mathbf{A} &= \Phi_L[n] \\ \mathbf{B}^{-1} &= \lambda \Phi_L[n-1] \\ \mathbf{C} &= \mathbf{x}_L[n] \\ \mathbf{D} &= 1. \end{aligned}$$

Entonces, aplicando el lema, podemos decir que:

$$\Phi_L^{-1}[n] = \lambda^{-1} \Phi_L^{-1}[n-1] - \frac{\lambda^{-1} \Phi_L^{-1}[n-1] \mathbf{x}_L[n] \mathbf{x}_L^T[n] \lambda^{-1} \Phi_L^{-1}[n-1]}{\mathbf{x}_L^T[n] \lambda^{-1} \Phi_L^{-1}[n-1] \mathbf{x}_L[n] + 1}. \quad (2.55)$$

Haciendo uso de la expresión recursiva (2.55), podemos prescindir de calcular $\Phi_L[n]$ y trabajar siempre con $\Phi_L^{-1}[n]$ que es justo lo que nos interesa. De esta forma, definiremos un matriz cuadrada $L \times L$ (siendo L el número de coeficientes de los filtros adaptativos).

$$\mathbf{Q}_L[n] = \Phi_L^{-1}[n] \quad (2.56)$$

y también definiremos un vector de ganancia $L \times 1$ como:

$$\mathbf{k}_L[n] = \frac{\lambda^{-1} \mathbf{Q}_L[n-1] \mathbf{x}_L[n]}{\mathbf{x}_L^T[n] \lambda^{-1} \mathbf{Q}_L[n-1] \mathbf{x}_L[n] + 1}. \quad (2.57)$$

De este modo la ecuación (2.55) quedaría,

$$\mathbf{Q}_L[n] = \lambda^{-1} \mathbf{Q}_L[n-1] - \lambda^{-1} \mathbf{k}_L[n] \mathbf{x}_L^T[n] \mathbf{Q}_L[n-1]. \quad (2.58)$$

Por otro lado, desarrollando la ecuación (2.57) del vector de ganancia, tenemos:

$$\mathbf{k}_L[n] = \lambda^{-1} \mathbf{Q}_L[n-1] \mathbf{x}_L[n] - \lambda^{-1} \mathbf{k}_L[n] \mathbf{x}_L^T[n] \mathbf{Q}_L[n-1] \mathbf{x}_L[n] \quad (2.59)$$

$$\mathbf{k}_L[n] = (\lambda^{-1} \mathbf{Q}_L[n-1] - \lambda^{-1} \mathbf{k}_L[n] \mathbf{x}_L^T[n] \mathbf{Q}_L[n-1]) \mathbf{x}_L[n]. \quad (2.60)$$

Por lo que haciendo uso de (2.58), tendremos:

$$\mathbf{k}_L[n] = \mathbf{Q}_L[n] \mathbf{x}_L[n]. \quad (2.61)$$

Con todas estas expresiones, e introduciendo (2.54) y (2.56) en (2.53) podemos formular unas ecuaciones recursivas para el cálculo de los coeficientes óptimos según:

$$\mathbf{w}_L[n] = \mathbf{Q}_L[n] \rho_L[n] = \mathbf{Q}_L[n] (\lambda \rho_L[n-1] + d[n] \mathbf{x}_L[n]) \quad (2.62)$$

$$\mathbf{w}_L[n] = \lambda \mathbf{Q}_L[n] \rho_L[n-1] + d[n] \mathbf{Q}_L[n] \mathbf{x}_L[n]. \quad (2.63)$$

Usando la ecuación (2.58) en (2.63) e identificando y recordando que $\mathbf{w}_L[n-1] = \mathbf{Q}_L[n-1] \rho_L[n-1]$ obtenemos:

$$\mathbf{w}_L[n] = \mathbf{Q}_L[n-1] \rho_L[n-1] - \mathbf{k}_L[n] \mathbf{x}_L^T[n] \mathbf{Q}_L[n-1] \rho_L[n-1] + d[n] \mathbf{Q}_L[n] \mathbf{x}_L[n] \quad (2.64)$$

$$\mathbf{w}_L[n] = \tilde{\mathbf{w}}_L[n-1] - \mathbf{k}_L[n] \mathbf{x}_L^T[n] \tilde{\mathbf{w}}_L[n-1] + d[n] \mathbf{k}_L[n] \quad (2.65)$$

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mathbf{k}_L[n] (d[n] - \mathbf{w}_L^T[n-1] \mathbf{x}_L[n]) \quad (2.66)$$

Es de notar que la expresión $(d[n] - \mathbf{w}_L^T[n-1] \mathbf{x}_L[n])$ incluida en la ecuación (2.66) representa una estimación de la señal de error (definido en (2.35) y llamado anteriormente error *a priori*) que en nuestro sistema adaptativo puede sustituirse por la propia señal de error (que en la mayoría de los sistemas adaptativos es una señal conocida o fácilmente calculable). Así, la ecuación (2.66) deriva en la siguiente expresión:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mathbf{k}_L[n] e[n]. \quad (2.67)$$

Con todo lo dicho hasta ahora, podríamos elaborar un simple algoritmo adaptativo RLS siguiendo los pasos descritos a continuación:

En el instante n -ésimo disponemos de los valores de $\mathbf{w}_L[n-1]$ y $\mathbf{Q}_L[n-1]$, y nos llega la siguiente nueva información: $d[n]$ y $x[n]$. Entonces:

- $e[n] = d[n] - \mathbf{w}_L^T[n-1] \mathbf{x}_L[n]$

- $\mathbf{z}[n] = \mathbf{Q}_L[n-1]\mathbf{x}_L[n]$
- $\mathbf{k}_L[n] = \frac{\mathbf{z}[n]}{\mathbf{x}_L^T[n]\mathbf{z}[n] + \lambda}$
- $\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mathbf{k}_L[n]e[n]$
- $\mathbf{Q}_L[n] = \lambda^{-1}\mathbf{Q}_L[n-1] - \lambda^{-1}\mathbf{k}_L[n]\mathbf{z}^T[n]$.

Hay que tener en cuenta que, puesto que el cálculo de la matriz $\mathbf{Q}_L[n]$ y de los coeficientes adaptativos $\mathbf{w}_L[n]$ es recursivo, necesitamos partir de un valor inicial que habitualmente suele ser un vector de ceros para $\mathbf{w}_L[n]$ y una matriz identidad (tamaño $L \times L$) ponderada por alguna constante para $\mathbf{Q}_L[n]$ [2], aunque existen estrategias para optimizar esta inicialización a partir del conocimiento previo de alguna característica de las señales que intervienen en el algoritmo [30].

El coste computacional de este algoritmo es excesivamente elevado comparado con el LMS puesto que se precisan $2L^2 + 5L + 1$ multiplicaciones por iteración, pero evita la realización de las inversiones matriciales (de coste computacional más elevado) para obtener una solución del problema mediante mínimos cuadrados.

En el anexo B se exponen algunas de las estrategias que pueden optimizar computacionalmente la implementación de estos algoritmos, puesto que podrían extrapolarse para la optimización de los algoritmos de proyección afín posteriormente descritos.

2.6. Filtrado de *Kalman*

El filtrado de *Kalman* [31], [32], es una potente herramienta para estimar el filtro óptimo tanto en procesos aleatorios estacionarios como no estacionarios. El objetivo del filtrado de *Kalman* es, en el caso general, encontrar de forma iterativa un estimador lineal, óptimo e insesgado del estado de un sistema en el instante n a partir de la información disponible hasta el instante anterior, actualizando con la información adicional del instante n (habitualmente denominada medida) dicha estimación. Para ello se supone que el sistema puede ser descrito mediante un modelo lineal estocástico de estados, donde tanto el error que pueda cometerse en el modelo del sistema como en la medida de la información adicional se consideran variables aleatorias incorreladas de media cero y varianza conocida.

Así, un sencillo sistema de estados que podría modelar al sistema adaptativo presentado en la figura 2.4, considerando que los coeficientes del filtro

adaptativo en el instante $n + 1$ son el estado del sistema a estimar, podría ser:

$$\mathbf{w}[n + 1] = \mathbf{A}[n]\mathbf{w}[n] + \mathbf{B}[n]\mathbf{u}[n] \quad (2.68)$$

donde el vector $\mathbf{w}[n]$ estaría formado por los L coeficientes del filtro adaptativo y la matriz $\mathbf{A}[n]$ contendría la información para la transición entre cada estado. En el caso más general, dicha matriz de transición puede variar con el tiempo (no estacionariedad del sistema). El modelo de la transición entre estados incorpora un error representado en $\mathbf{B}[n]\mathbf{u}[n]$, donde $\mathbf{u}[n]$ es un vector cuyos elementos proceden de una variable aleatoria, y $\mathbf{B}[n]$ la matriz que controla el error que se comete en el modelo de estados.

Por otro lado, la señal $d[n]$, puede ser considerada como la información adicional del sistema de estados (medida) en el instante n -ésimo, que según se desprende de la figura 2.4, sería:

$$d[n] = \mathbf{x}^T[n]\mathbf{w}[n] + r[n]. \quad (2.69)$$

$r[n]$ es una secuencia aleatoria que modela el error en la medida y $\mathbf{x}[n]$ un vector con los L últimos valores de la señal $x[n]$.

Con las ecuaciones (2.68) y (2.69), se tendría descrito el sistema de estados, pero se trata de encontrar iterativamente la mejor estimación del estado actual, conocidas las estimaciones de los estados anteriores. Así, si denominamos $\hat{d}[n|n-1]$, $\hat{\mathbf{w}}[n|n-1]$ y $\hat{r}[n|n-1]$ a las estimaciones de las variables $d[n]$, $\mathbf{w}[n]$ y $r[n]$ a partir del conocimiento de las estimaciones de dichas variables hasta el instante anterior, se cumple que la estimación de la medida es:

$$\hat{d}[n|n-1] = \mathbf{x}^T[n]\hat{\mathbf{w}}[n|n-1] + \hat{r}[n|n-1]. \quad (2.70)$$

Si consideramos que la secuencia aleatoria $r[n]$ está incorrelada, entonces $\hat{r}[n|n-1] = 0$.

Conocida la medida, el error en la estimación de la misma será:

$$e[n] = d[n] - \hat{d}[n|n-1] = d[n] - \mathbf{x}^T[n]\hat{\mathbf{w}}[n|n-1]. \quad (2.71)$$

Por tanto, el problema se reduce a encontrar el estimador $\hat{\mathbf{w}}[n|n-1]$.

La variable aleatoria que modela la señal de error en la estimación está incorrelada con ella misma (es decir, su matriz de autocorrelación es una matriz diagonal). Aprovechando esta circunstancia, tratemos de encontrar el estimador lineal de $\mathbf{w}[n]$ por mínimos cuadrados a partir del conocimiento de la señal $e[n]$, o lo que es lo mismo, la matriz K que cumple $\hat{\mathbf{w}}[n]_{|e[n]} = K\mathbf{e}[n]$, donde $\mathbf{e}[n]$, almacenaría el historial de valores desde $n = 0$ hasta el instante actual. Sabemos que K ha de ser la solución de la ecuación normal

$$K\mathbf{R}_e[n] = \mathbf{R}_{we}[n], \quad (2.72)$$

siendo $\mathbf{R}_e[n]$, la matriz de autocorrelación de $e[n]$ (matriz diagonal, que contendría en su diagonal los elementos: $R_e[0], R_e[1], \dots, R_e[n]$, donde $R_e[i] = E\{e[i]e[i]\}$), y $\mathbf{R}_{we}[n]$ la matriz de correlación cruzada entre $\mathbf{w}[n]$ y $\mathbf{e}[n]$. De esta forma:

$$\hat{\mathbf{w}}[n]_{|\mathbf{e}[n]} = \mathbf{R}_{we}[n]\mathbf{R}_e^{-1}[n]\mathbf{e}[n], \quad (2.73)$$

y puesto que $\mathbf{R}_e^{-1}[n]$ es diagonal,

$$\hat{\mathbf{w}}[n]_{|\mathbf{e}[n]} = \sum_{i=0}^n E\{\mathbf{w}[n]e[i]R_e^{-1}[i]e[i]\} = \sum_{i=0}^n \hat{\mathbf{w}}[n]_{|\mathbf{e}[i]}. \quad (2.74)$$

Es decir, que $\hat{\mathbf{w}}[n|n]$ (estimador de $\mathbf{w}[n]$ usando las n observaciones anteriores, es:

$$\begin{aligned} \hat{\mathbf{w}}[n|n] &= \hat{\mathbf{w}}[n]_{|\mathbf{e}[n]} + \sum_{i=0}^{n-1} E\{\mathbf{w}[n]e[i]\}R_e^{-1}[i]e[i] \\ &= E\{\mathbf{w}[n]e[n]\}R_e^{-1}[n]e[n] + \hat{\mathbf{w}}[n|n-1]. \end{aligned} \quad (2.75)$$

Como el estimador buscado es $\hat{\mathbf{w}}[n|n-1]$, ayudándonos de la relación expresada en (2.75), podemos escribir:

$$\hat{\mathbf{w}}[n+1|n] = \hat{\mathbf{w}}[n+1|n-1] + E\{\mathbf{w}[n+1]e[n]\}R_e^{-1}[n](d[n] - \mathbf{x}^T[n]\hat{\mathbf{w}}[n|n-1]), \quad (2.76)$$

donde ahora $R_e^{-1}[n] = E\{e[n]e[n]\}$ y se ha introducido el resultado hayado en (2.71).

Haciendo uso de la ecuación (2.68)

$$\hat{\mathbf{w}}[n+1|n-1] = \mathbf{A}[n]\hat{\mathbf{w}}[n|n-1] + \mathbf{B}[n]\mathbf{u}[n|n-1], \quad (2.77)$$

y considerando que $\mathbf{u}[n|n-1] = 0$ (puesto que $u[n]$ es una variable aleatoria incorrelada con ella misma) podemos introducir dicho resultado en (2.76), obteniendo finalmente:

$$\begin{aligned} \hat{\mathbf{w}}[n+1|n] &= \mathbf{A}[n]\hat{\mathbf{w}}[n|n-1] + E\{\mathbf{w}[n+1]e[n]\}R_e^{-1}[n](d[n] - \mathbf{x}^T[n]\hat{\mathbf{w}}[n|n-1]) \\ &= (\mathbf{A}[n] - E\{\mathbf{w}[n+1]e[n]\}R_e^{-1}[n]\mathbf{x}^T[n])\hat{\mathbf{w}}[n|n-1] \\ &\quad + E\{\mathbf{w}[n+1]e[n]\}R_e^{-1}[n]d[n] \\ &= \mathbf{K}_p[n]\hat{\mathbf{w}}[n|n-1] + \mathbf{G}[n]d[n], \end{aligned} \quad (2.78)$$

siendo $\mathbf{G}[n] = E\{\mathbf{w}[n+1]e[n]\}R_e^{-1}[n]$ y $\mathbf{K}_p[n] = \mathbf{A}[n] - \mathbf{G}[n]\mathbf{x}^T[n]$. $\mathbf{G}[n]$ es conocido como la ganancia de *Kalman* y regula la influencia de la medida en

la regla de variación de la estimación de un estado a partir de la estimación del anterior.

Para calcular $\mathbf{G}[n]$ primero hallamos el valor medio $E\{\mathbf{w}[n+1]e[n]\}$

$$\begin{aligned} E\{\mathbf{w}[n+1]e[n]\} &= E\{(\mathbf{A}[n]\mathbf{w}[n] + \mathbf{B}[n]\mathbf{u}[n])e[n]\} \\ &= \mathbf{A}[n]E\{\mathbf{w}[n]e[n]\} + \mathbf{B}[n]E\{\mathbf{u}[n]e[n]\} \end{aligned} \quad (2.79)$$

Para obtener el valor de las dos esperanzas matemáticas que aparecen en (2.79), nos ayudamos de la autocorrelación del vector de error de la estimación del estado (coeficientes estimados), definido como:

$$\tilde{\mathbf{w}}[n|n-1] = \mathbf{w}[n] - \hat{\mathbf{w}}[n|n-1], \quad (2.80)$$

y su autocorrelación

$$\mathbf{P}[n|n-1] = E\{\tilde{\mathbf{w}}[n|n-1]\tilde{\mathbf{w}}^T[n|n-1]\}. \quad (2.81)$$

Así,

$$E\{\mathbf{w}[n]e[n]\} = E\{(\tilde{\mathbf{w}}[n|n-1] + \hat{\mathbf{w}}[n|n-1])e[n]\}. \quad (2.82)$$

Si consideramos que $e[n]$ está incorrelada con $\hat{\mathbf{w}}[n|n-1]$ y que $r[n]$ lo está con $\tilde{\mathbf{w}}[n|n-1]$, la ecuación (2.82) quedaría:

$$\begin{aligned} E\{\mathbf{w}[n]e[n]\} &= E\{\tilde{\mathbf{w}}[n|n-1]e[n]\} = E\{\tilde{\mathbf{w}}[n|n-1](\mathbf{x}^T[n]\tilde{\mathbf{w}}[n|n-1] + r[n])\} \\ &= E\{\tilde{\mathbf{w}}[n|n-1]\mathbf{x}^T[n]\tilde{\mathbf{w}}[n|n-1]\} = \mathbf{P}[n|n-1]\mathbf{x}^T[n], \end{aligned} \quad (2.83)$$

donde se ha tenido en cuenta que

$$\begin{aligned} e[n] &= d[n] - \mathbf{x}^T[n]\hat{\mathbf{w}}[n|n-1] = \mathbf{x}^T[n]\mathbf{w}[n] - \mathbf{x}^T[n]\hat{\mathbf{w}}[n|n-1] + r[n] \\ &= \mathbf{x}^T[n]\tilde{\mathbf{w}}[n|n-1] + r[n]. \end{aligned} \quad (2.84)$$

Considerando que $r[n]$ y $\tilde{\mathbf{w}}[n|n-1]$ están incorrelados, encontramos por último, la relación:

$$E\{\mathbf{u}[n]e[n]\} = E\{\mathbf{u}[n](\mathbf{x}^T[n]\tilde{\mathbf{w}}[n|n-1] + r[n])\} = E\{\mathbf{u}[n]r[n]\} = R_{ur}[n]. \quad (2.85)$$

Ahora, podemos expresar el vector de ganancia de Kalman en función de los resultados calculados:

$$\mathbf{G}[n] = (\mathbf{A}[n]\mathbf{P}[n|n-1]\mathbf{x}^T[n] + \mathbf{B}[n]R_{ur}[n])R_e^{-1}[n] \quad (2.86)$$

El valor de $R_e[n]$, puede deducirse de:

$$\begin{aligned} E\{e[n]e[n]\} &= E\{e[n]^2\} = E\{(\mathbf{x}^T[n]\tilde{\mathbf{w}}[n|n-1] + r[n])^2\} \\ &= \mathbf{x}^T[n]\mathbf{P}[n|n-1]\mathbf{x}[n] + R_r[n], \end{aligned} \quad (2.87)$$

donde se ha considerado la incorrelación entre $r[n]$ y $\tilde{\mathbf{w}}[n|n-1]$, y se ha llamado $R_r[n]$ a $E\{r[n]r[n]\}$.

Volviendo a la ecuación (2.86), para calcular la ganancia de *Kalman* en el instante n -ésimo se precisan de datos desconocidos en dicho instante como $\mathbf{P}[n|n-1]$. Sin embargo, este valor puede calcularse recursivamente como se describe a continuación.

Sabemos que $\mathbf{w}[n] = \tilde{\mathbf{w}}[n|n-1] + \hat{\mathbf{w}}[n|n-1]$, y que $\tilde{\mathbf{w}}[n|n-1]$ y $\hat{\mathbf{w}}[n|n-1]$ han de estar incorrelados. Así, si calculamos la autocorrelación de $\mathbf{w}[n]$:

$$\begin{aligned} \mathbf{\Pi}[n] &= E\{\mathbf{w}[n]\mathbf{w}^T[n]\} = E\{\hat{\mathbf{w}}[n|n-1]\hat{\mathbf{w}}^T[n|n-1]\} \\ &\quad + E\{\tilde{\mathbf{w}}[n|n-1]\tilde{\mathbf{w}}^T[n|n-1]\} = \mathbf{\Sigma}[n] + \mathbf{P}[n|n-1], \end{aligned} \quad (2.88)$$

de donde se deduce que $\mathbf{P}[n+1|n] = \mathbf{\Pi}[n+1] - \mathbf{\Sigma}[n+1]$. Es fácil deducir que:

$$\mathbf{\Pi}[n+1] = \mathbf{A}[n]\mathbf{\Pi}[n]\mathbf{A}^T[n] + \mathbf{B}[n]\mathbf{R}_u[n]\mathbf{B}^T[n], \quad (2.89)$$

con $\mathbf{R}_u[n] = E\{\mathbf{u}[n]\mathbf{u}^T[n]\}$, y

$$\mathbf{\Sigma}[n+1] = \mathbf{A}[n]\mathbf{\Sigma}[n]\mathbf{A}^T[n] + \mathbf{G}[n]R_r[n]\mathbf{G}^T[n]. \quad (2.90)$$

Por tanto,

$$\begin{aligned} \mathbf{P}[n+1|n] &= \mathbf{A}[n]\mathbf{\Pi}[n]\mathbf{A}^T[n] + \mathbf{B}[n]\mathbf{R}_u[n]\mathbf{B}^T[n] - \mathbf{A}[n]\mathbf{\Sigma}[n]\mathbf{A}^T[n] \\ &\quad - \mathbf{G}[n]R_r[n]\mathbf{G}^T[n] = \mathbf{A}[n](\mathbf{\Pi}[n] - \mathbf{\Sigma}[n])\mathbf{A}^T[n] \\ &\quad + \mathbf{B}[n]\mathbf{R}_u[n]\mathbf{B}^T[n] - \mathbf{G}[n]R_r[n]\mathbf{G}^T[n] \\ &= \mathbf{A}[n](\mathbf{P}[n|n-1])\mathbf{A}^T[n] + \mathbf{B}[n]\mathbf{R}_u[n]\mathbf{B}^T[n] - \mathbf{G}[n]R_r[n]\mathbf{G}^T[n]. \end{aligned} \quad (2.91)$$

Así, fijando unas condiciones iniciales de funcionamiento ($\mathbf{\Sigma}[0] = 0$, $\hat{\mathbf{w}}[0| -1] = 0$ y $\mathbf{P}[0| -1] = \mathbf{\Pi}[0]$), y conocidos $d[n]$, $\mathbf{A}[n]$, $\mathbf{B}[n]$, $R_r[n]$, $R_u[n]$ y $R_{ur}[n]$, el algoritmo de *Kalman* para resolver el filtro óptimo de un sistema adaptativo quedaría:

- $R_e[n] = \mathbf{x}^T[n]\mathbf{P}[n|n-1]\mathbf{x}[n] + R_r[n]$
- $\mathbf{G}[n] = (\mathbf{A}[n]\mathbf{P}[n|n-1]\mathbf{x}^T[n] + \mathbf{B}[n]R_{ur}[n])R_e^{-1}[n]$
- $e[n] = d[n] - \mathbf{x}^T[n]\hat{\mathbf{w}}[n|n-1]$
- $\hat{\mathbf{w}}[n+1|n] = \mathbf{K}_p[n]\hat{\mathbf{w}}[n|n-1] + \mathbf{G}[n]e[n]$
- $\mathbf{K}_p[n] = \mathbf{A}[n] - \mathbf{G}[n]\mathbf{x}^T[n]$
- $\mathbf{P}[n+1|n] = \mathbf{A}[n](\mathbf{P}[n|n-1])\mathbf{A}^T[n] + \mathbf{B}[n]\mathbf{R}_u[n]\mathbf{B}^T[n] - \mathbf{G}[n]R_r[n]\mathbf{G}^T[n]$

Notar que el algoritmo RLS puede ser considerado un caso particular del filtrado de *Kalman* cuando $\mathbf{A}[n] = \frac{1}{\sqrt{\lambda}}$ y $\mathbf{B}[n] = 0$, [33]. Aunque el coste computacional de este algoritmo es elevado, también existen extrategias que permiten reducirlo como la descrita en [34] donde se presenta una versión eficiente del filtrado de *Kalman* aplicada al CAR.

2.7. Sistemas adaptativos aplicados al CAR

Hasta el momento se han presentado algunos algoritmos usados en sistemas adaptativos, cuyo objetivo es tratar de minimizar una señal denominada señal de error, consistente en la diferencia entre otra señal ya existente (denominada señal deseada) y la generada por el filtro adaptativo.

Aplicar dichos sistemas adaptativos a un sistema de control activo de ruido resulta casi inmediato, sin más que considerar la señal de ruido como la señal a minimizar. Pero hay que tener en cuenta ciertas diferencias con los sistemas adaptativos estándar que condicionarán la convergencia de los algoritmos y motivarán algunas modificaciones sobre dichos algoritmos que controlan los sistemas adaptativos.

Un sistema monocanal de control activo de ruido, puede representarse tal y como aparece en la figura 2.7.

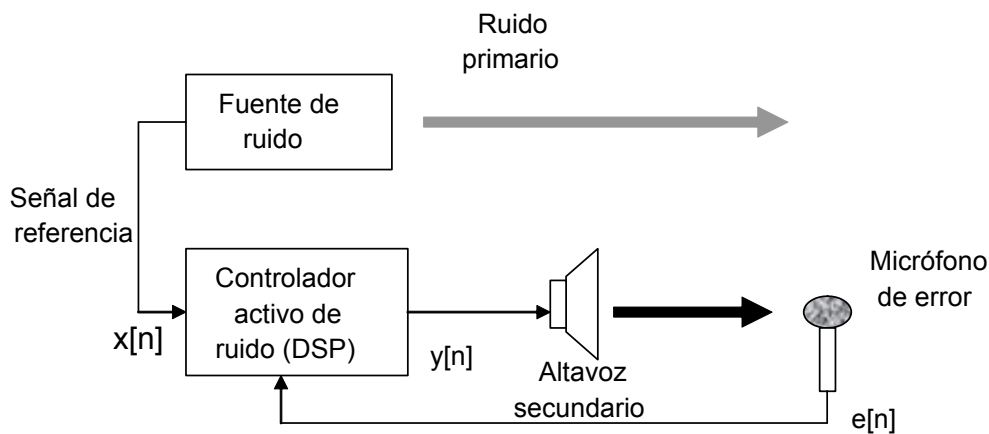


Figura 2.7: Sistema de control activo de ruido.

Si consideramos que la parte acústica provoca en la señal un retardo lineal y una variación en la ganancia de la misma, que depende del entorno donde se encuentre el sistema, y que la respuesta electroacústica tanto del altavoz

como del micrófono, puede considerarse lineal (o por lo menos se va a trabajar con señales que varían dentro del margen lineal de dichos componentes), entonces podemos representar el mismo esquema anterior mediante un diagrama de bloques como el representado en la figura 2.8, que comparándolo con el esquema general de un sistema adaptativo presentado en la figura 2.4, nos permite apreciar las diferencias y analogías existentes entre ambos sistemas.

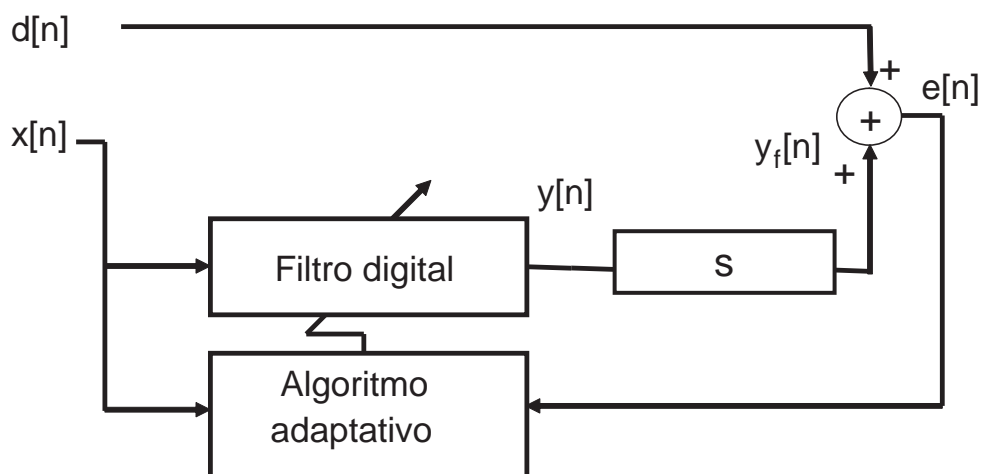


Figura 2.8: Diagrama de bloques del sistema de control activo de ruido.

Aunque ambos esquemas intentan minimizar una señal de error $e[n]$, en el primer caso (figura 2.4), dicha señal de error es la diferencia entre la señal deseada $d[n]$ y la generada por nuestro filtro adaptativo $y[n]$, mientras que en el segundo caso (figura 2.8) la señal de error sería la mezcla acústica de la señal deseada, y una versión de nuestra señal generada filtrada por el modelo de lo que se conoce como planta acústica ($S(z)$). Y es ésta la principal diferencia entre un sistema adaptativo estándar, y uno aplicado al CAR. El bloque s , que aparece en la figura 2.8, representa un filtro lineal (a menudo suele ser modelado por un filtro FIR) que contiene la respuesta conjunta del sistema electroacústico (esto es, respuesta de los conversores A/D y D/A, amplificadores, filtros de reconstrucción y antialiasing, micrófono, altavoz y camino acústico entre el altavoz y el micrófono). En la mayoría de los casos, dicho filtro se considera invariante en el tiempo (las condiciones del entorno y las posiciones de los transductores acústicos no varían), pero también puede considerarse variante por lo que vendría representada por $s[n]$. La aparición de este filtrado introduce una variación en la señal de error motivada por el retardo introducido en la señal generada por el filtro adaptativo (y en menor medida, por su variación en amplitud), tal y como se discute en [19]. Esto

provoca una degradación en las propiedades de convergencia de los algoritmos que en muchos de los casos conduce a la divergencia, por lo que tenemos que compensar de algún modo la introducción del filtro s en el sistema adaptativo. Por tanto, la principal diferencia que se puede encontrar entre un sistema de CAR y un sistema adaptativo estándar, es la existencia de los caminos acústicos que enlazan la fuente secundaria con el sensor de error (camino secundario s) y la fuente de ruido con el sensor de error (camino primario), ya que en el caso del sistema adaptativo no existen.

Otra diferencia que puede observarse de la comparación entre las figuras 2.4 y 2.8, es que en el primer caso, la señal de error se forma como la diferencia entre dos señales, mientras que en el segundo, dicha señal es la suma, puesto que el micrófono que suele actuar como sensor de error recoge la mezcla aditiva de los dos campos acústicos que se produce en el aire. Esta diferencia es mucho menos grave que la anterior, puesto que solo involucra un cambio de signo en una de las señales (que luego se verá reflejado en la ecuación de actualización de los coeficientes).

En un sistema de CAR (y en un sistema adaptativo en general) para que pueda existir cancelación, es necesario que la señal de entrada al filtro adaptativo esté correlada con la señal $d[n]$ (habitualmente llamada señal deseada y que en los sistemas CAR la podemos identificar como el ruido existente cuando no se ejerce control). La señal $x[n]$ debe estar correlada por tanto con $d[n]$, y normalmente es obtenida a partir de un micrófono que únicamente recoge la señal de ruido, o mediante otro tipo de sensores no acústicos como acelerómetros o tacómetros que nos pueden dar información del ruido acústico a partir de características mecánicas de la fuente de ruido.

Al intentar encontrar mediante cualquier algoritmo iterativo los coeficientes que realizan el filtrado óptimo, se obtiene una relación iterativa para encontrar dichos coeficientes que suele depender entre otros factores de la señal de error $e[n]$ y la señal de entrada al filtro adaptativo $x[n]$. Sin embargo, en un sistema de CAR, la señal $e[n]$ no es la diferencia entre una señal $d[n]$ correlada con $x[n]$ y la señal $y[n]$, salida del filtro adaptativo, como ya ha quedado claro, sino que la señal que llega al micrófono ha atravesado varios subsistemas que modifican sus características desde que sale del filtro adaptativo. De hecho, ni siquiera la señal de error que se recoge en el micrófono es la misma que llega al algoritmo. Por tanto, en los algoritmos adaptativos usados en aplicaciones de CAR, hay que introducir de alguna forma el efecto de los caminos secundarios y compensar de esta forma las variaciones producidas por las respuestas de los mismos.

La introducción del camino primario (modelado como un filtro, $P(z)$) no implica tantos inconvenientes, puesto que la señal $d[n]$ antes y después de pasar

por el camino acústico, aunque modifica sus propiedades, sigue siendo una señal correlada con $x[n]$, por lo que en vez de considerar en nuestro algoritmo la señal $d[n]$ antes de pasar por el camino primario, puede considerarse después de pasar por él sin tener ninguna repercusión en su funcionamiento. De hecho, suele considerarse que la relación que existe entre la señal de referencia $x[n]$, y la deseada $d[n]$, en el dominio transformado es $D(z) = X(z)P(z)$, o lo que es lo mismo, la señal deseada sería la de referencia filtrada por el camino primario. Por tanto, respecto del camino primario, no tenemos que hacer nada, pero sí en los caminos secundarios, puesto que la señal que genera nuestro filtro adaptativo no es, como ya se ha dicho, la que se mezcla con la señal de error en el sensor de error (está modificada por el camino secundario). Tenemos que tener en cuenta que la señal de referencia que aparece en la ecuación de actualización de los coeficientes y la señal $y_f[n]$ que se mezcla con la deseada $d[n]$ para obtener la señal de error $e[n]$, deben estar relacionadas mediante la función de transferencia del filtro adaptativo. Por lo tanto, para que dicha relación se mantenga, tenemos que filtrar también la señal de referencia por una estimación del camino secundario, antes de introducirla en la ecuación que gobierna la actualización de los coeficientes o deshacer el efecto de dicho camino secundario sobre la señal $y[n]$. Esto debe ser así porque la señal $y[n]$ que aparece en la expresión de la señal de error en los algoritmos adaptativos, no es más que la señal $x[n]$ filtrada por el filtro adaptativo, lo que no sucede cuando la aplicación es control activo de ruido debido a la existencia del camino secundario.

Para corregir estas diferencias entre un sistema adaptativo y dicho sistema aplicado al control activo de ruido se proponen varias estructuras que a continuación se detallan.

2.7.1. Estructura directa

Una posible solución simple e intuitiva que pudiera corregir los efectos del camino acústico secundario en la aplicación de los algoritmos adaptativos al control activo de ruido podría consistir en, a partir de una estimación del camino secundario, implementar el filtro inverso de dicho camino y filtrar la salida del filtro adaptativo por dicha estimación de forma que el conjunto formado por el camino secundario real y su filtro inverso se compensen como se puede apreciar en la figura 2.9, y la señal $y[n]$ que llega al micrófono coincida con la que sale del filtro adaptativo y se usa en el algoritmo.

En este caso, considerando que $D(z) = X(z)P(z)$, con esta estructura se alcanzaría la solución $W(z) = -P(z)$ para los coeficientes del filtro adaptativo en estado estacionario. Es decir, se trataría de un sistema de identificación de

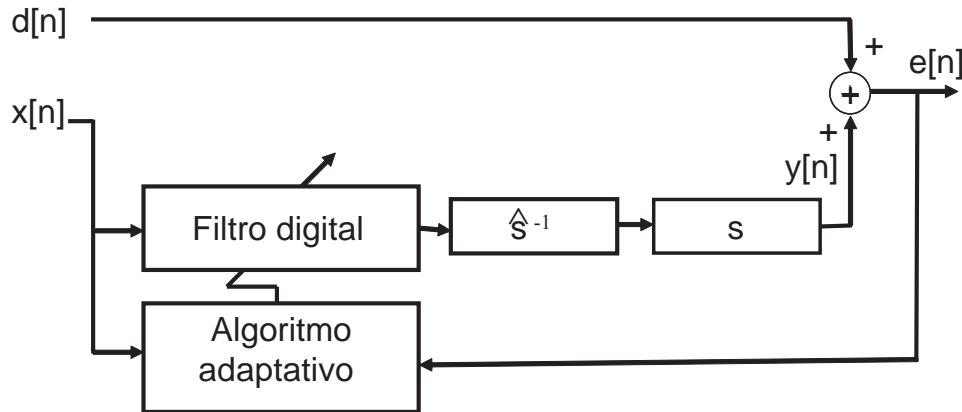


Figura 2.9: Post-ecualización por el camino inverso.

canal. Sin embargo, esta solución no siempre es posible, puesto que no podemos garantizar la estabilidad del filtro inverso del camino secundario. Además, el filtro inverso no puede ser anticausal si queremos que sea implementable, por lo que no solo no eliminaremos el retardo, sino que aumentaría. Pero esta estrategia puede considerarse como primera solución teórica del problema, y para señales de ruido periódicas puede obtener buenos resultados, siempre que se haya estimado bien el valor del filtro inverso en las frecuencias de interés. Dicha estructura también suele denominarse post-ecualización de la señal filtrada y admite una solución similar (llamada pre-ecualización) si filtramos por el modelo del filtro inverso antes del filtro adaptativo como se muestra en la figura (2.10), dando lugar a la denominada pre-ecualización de la señal filtrada, con idéntica solución en estado estacionario del filtro adaptativo.

2.7.2. Estructura clásica o de filtrado-x convencional

Hemos visto que el problema comentado con la existencia del camino secundario no está del todo resuelto con la estructura directa anteriormente presentada, y además, el empleo de un sistema adaptativo sin más puede degradar en exceso las propiedades de convergencia del sistema. Otras configuraciones y estructuras se han estudiado para tratar de corregir el retardo que introduce el camino acústico y que hay que tratar de compensar de alguna manera. Una solución a este problema fue propuesta en paralelo por Widrow [35] y Burgess [36] y más tarde por Morgan [37]. Su propuesta, inicialmente desarrollada

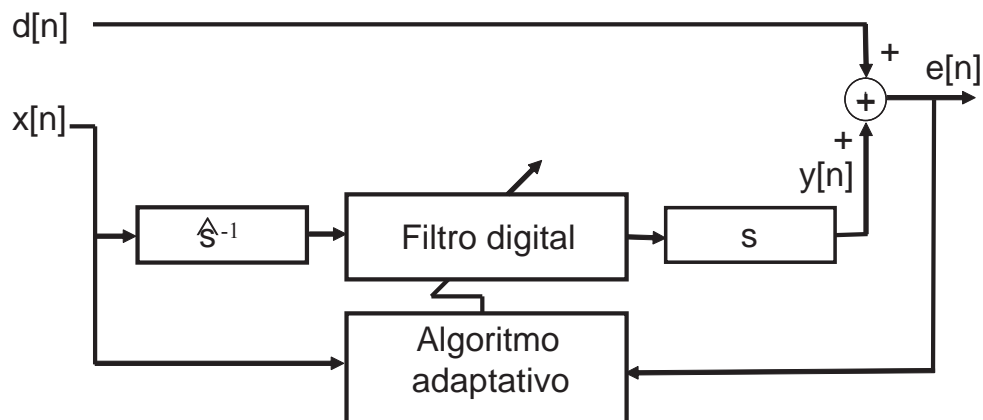


Figura 2.10: Pre-ecualización por el camino inverso.

considerando el algoritmo LMS como algoritmo de control, consistía en filtrar la señal $x[n]$ por la estimación del camino secundario antes de introducir los datos en el algoritmo adaptativo como se puede apreciar en la figura 2.11. De esta manera, la señal $x_f[n]$ que se utiliza en el algoritmo LMS y la señal $y_f[n]$ a partir de la que se obtiene la señal de error, sí que están relacionadas por el filtro adaptativo. Es decir, $y_f[n]$ es $x_f[n]$ tras pasar por el filtro adaptativo. Básicamente, lo que realiza esta estructura junto con el algoritmo LMS es aplicar dicho algoritmo pero a la señal $x_f[n]$ en vez de a $x[n]$. Es decir, la señal de referencia es ahora $x_f[n]$.

Otra forma de justificar la necesidad de este filtrado por el camino secundario es que podemos considerar los sistemas que intervienen en el modelo como lineales e invariantes (en el caso del filtro adaptativo esto es correcto cuando los pesos varían de manera suficientemente lenta como vamos a suponer que es el caso) y por tanto intercambiables. De esta forma, y según se aprecia en la figura 2.12, podríamos intercambiar los filtrados hasta obtener un sistema aproximado equivalente.

En esta ocasión, la solución estacionaria de los coeficientes del filtro adaptativo contiene también los efectos del camino secundario, siendo $W(z) = -P(z)S^{-1}(z)$.

Esta estructura aplicada al control activo de ruido y controlada por el algoritmo LMS es conocida como algoritmo de filtrado-x LMS [35].

Para la implementación de este nuevo algoritmo, necesitamos una estimación del camino secundario para realizar el filtrado que nos proporcione $x_f[n]$

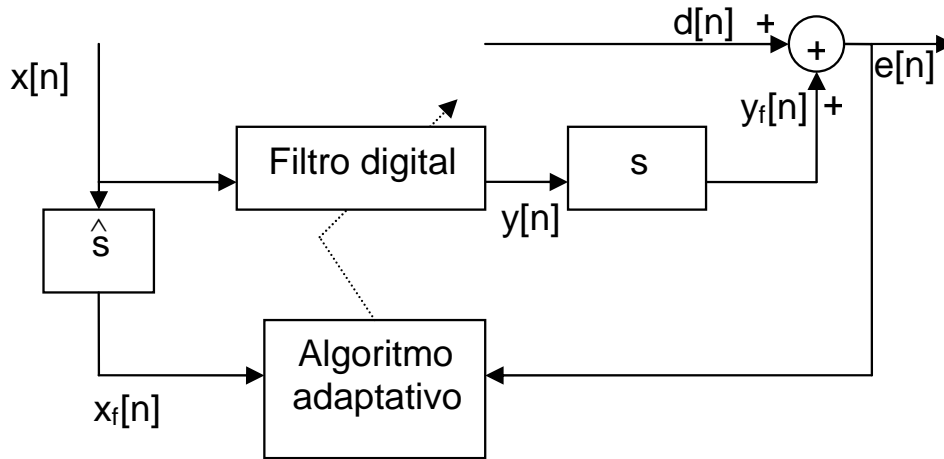


Figura 2.11: Solución empleada para contrarrestar el efecto del camino secundario haciendo uso de la estructura de filtrado-x convencional.

a partir de $x[n]$, lo que suele realizarse en una etapa previa de *setup* antes de ejecutar el algoritmo. Dicha etapa suele modelizar el camino acústico como un filtro FIR de M coeficientes.

La ventaja de este método frente al anterior, es que dicha estimación es más sencilla de obtener y con más precisión que la inversa del sistema acústico.

Esta estructura, puede ser extrapolada para cualquier algoritmo, siendo su diagrama de bloques más general el expresado en la figura 2.13.

Los pasos que se deberían realizar en cada iteración para implementar un algoritmo adaptativo aplicado al CAR con estructura de filtrado-x convencional podrían ser:

1. Obtención de las muestras de la señal primaria $x[n]$ y de la señal de error $e[n]$.
2. Actualizar al vector de entrada $\mathbf{x}[n]$.
3. Filtrar $x[n]$ por la estimación del camino secundario: $x_f[n] = \hat{s} * x[n]$.
4. Actualizar el vector $\mathbf{x}_f[n]$.
5. Filtrar $\mathbf{x}[n]$ por el filtro adaptativo: $y[n] = \mathbf{w}[n]^T \mathbf{x}[n]$.

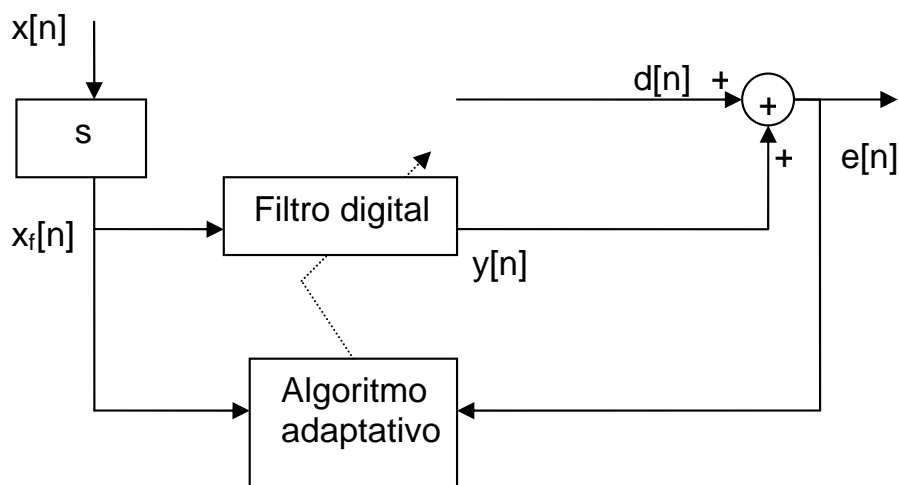


Figura 2.12: Aproximación intercambiando sistemas lineales invariantes.

6. Actualización de pesos según la ecuación de actualización del algoritmo usado:

$$\mathbf{w}[n + 1] = \mathbf{w}[n] + \mu e[n] \mathbf{x}_f[n],$$
 si el algoritmo es el LMS.

Como podemos observar, el número de operaciones a realizar crece con respecto al LMS estándar ($2L + M + 1$, frente a $2L + 1$ del LMS), ya que se introduce el filtrado adicional por la estimación del camino acústico.

Aunque la introducción de esta estructura (o alguna equivalente) es necesaria para el buen funcionamiento del algoritmo adaptativo aplicado al CAR, también se introducen ciertas desventajas como que el coste computacional es mayor que aplicar únicamente el algoritmo adaptativo (necesidad de un filtrado adicional para generar la nueva señal), y la ralentización de la convergencia debido a una cota menor para la constante de convergencia μ . La disminución de esta cota es debida a dos factores: el retardo inherente a los sistemas de control activo de ruido entre la señal generada y el punto de cancelación, y la más que posible imprecisión a la hora de estimar el camino secundario. En [38]-[41], se discuten estrategias para estimar la cota óptima del paso de convergencia para el algoritmo de filtrado-x LMS. En [42], se discute cómo mejorar aun más la eficiencia de este algoritmo.

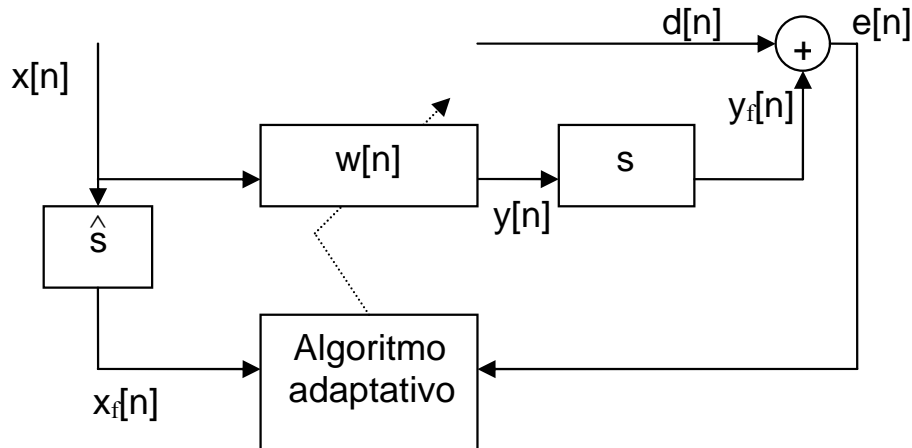


Figura 2.13: Estructura de filtrado-x convencional para control activo de ruido.

2.7.3. Estructura de filtrado-x modificada

Para tratar de compensar la disminución de la velocidad de convergencia introducida por la estructura de filtrado-x convencional descrita en el apartado anterior, surge la estructura de filtrado-x modificada. El esquema simplificado de la estructura modificada puede contemplarse en la figura 2.14.

Esta estructura, aunque es más compleja computacionalmente como se puede comprobar en el diagrama de bloques (realiza dos filtrados más que la estructura de filtrado-x), intenta mitigar los efectos negativos que introduce cualquier otra estructura cuando aplicamos un algoritmo adaptativo al CAR, en cuanto a las propiedades de convergencia. Según se ha comentado anteriormente, estas propiedades quedan degradadas porque la señal que genera el algoritmo adaptativo después de filtrar la señal de referencia por el filtro adaptativo, no es la misma que se mezcla con la señal deseada, sino que ha de pasar por un sistema acústico equivalente a un filtrado. Una de las premisas que justificaban el uso de la estructura de filtrado-x era que dicho filtrado acústico podía ser intercambiado por el filtro adaptativo (suposición de linealidad, válida cuando los coeficientes del filtro adaptativo varían lentamente). La estructura modificada, realiza implícitamente dicho intercambio de forma que la señal generada es la que se mezcla con la señal deseada para obtener una señal de error que controle el algoritmo adaptativo. Si nos fijamos en la

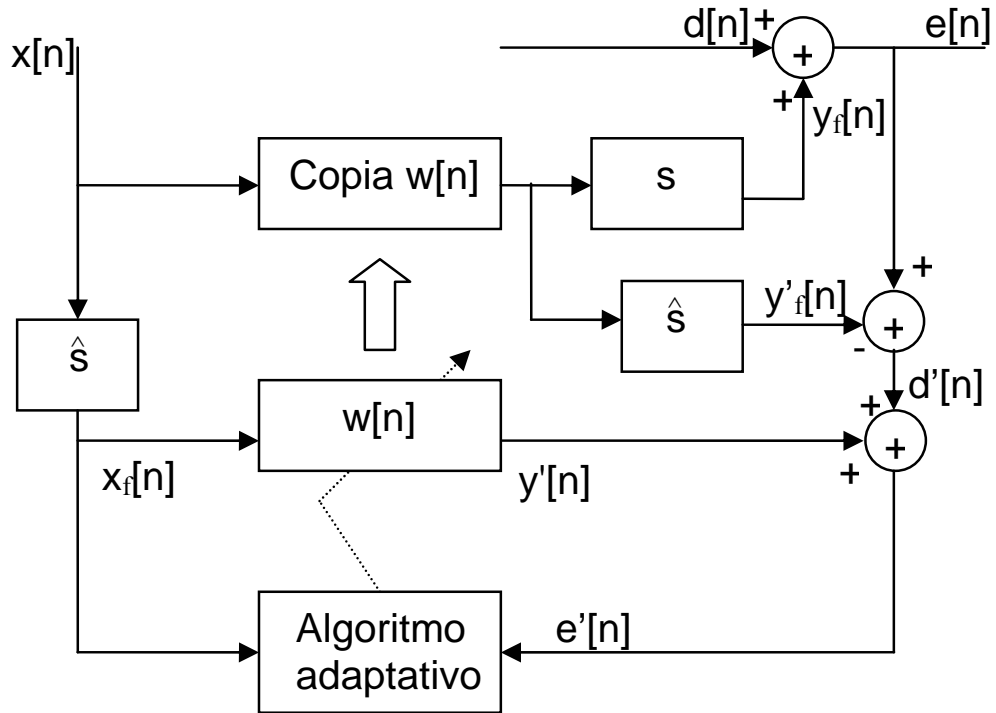


Figura 2.14: Esquema de la estructura de filtrado-x modificada para aplicaciones de CAR.

rama inferior del diagrama de bloques presentado, su estructura es idéntica a la de cualquier sistema adaptativo. Es por ello que las prestaciones obtenidas por esta estructura son muy similares a las obtenidas por el sistema adaptativo sin aplicarlo al CAR (serían idénticas si la estimación del camino secundario fuera exacta). La solución en estado estacionario de los coeficientes del filtro adaptativo es $W(z) = -P(z)\hat{S}^{-1}(z)$, que en el caso de estimación exacta del camino secundario, coincidiría con la esperada y obtenida en la estructura del filtrado-x convencional.

El problema que nos encontramos cuando planteamos esta estructura es que en los sistemas de control activo de ruido no disponemos de la señal deseada ($d[n]$) sino que directamente se obtiene la señal de error. En este caso, se usa el modelo de la planta acústica para extraer de la señal de error, la contribución del actuador y así poder estimar el campo primario $d[n]$. A esta estimación se le suele llamar $d'[n]$, y se calcula de la siguiente forma:

$$d'[n] = e[n] - \hat{\mathbf{s}}^T \mathbf{y}[n] \quad (2.92)$$

siendo $\hat{\mathbf{s}}$ el vector con los coeficientes de la estimación de la planta acústica.

La estructura modificada puede estudiarse con más detalle en [43] y [44].

Suponiendo que el algoritmo adaptativo usado sea el LMS, las operaciones que definirían este sistema adaptativo aplicado al control activo de ruido y con la estructura modificada serían:

1. Obtención de las muestras de la señal primaria $x[n]$ y de la señal de error $e[n]$.
2. Actualizar el vector de entrada $\mathbf{x}[n]$.
3. Filtrar $\mathbf{x}[n]$ por la estimación del camino secundario: $x_f[n] = \hat{\mathbf{s}} * \mathbf{x}[n]$.
4. Actualizar el vector $\mathbf{x}_f[n] = [x_f[n], x_f[n-1], \dots, x_f[n-L+1]]$.
5. Filtrar $\mathbf{x}_f[n]$ por el filtro adaptativo: $y'_f[n] = \mathbf{x}_f[n] * \mathbf{w}[n]$.
6. Actualizar el vector $\mathbf{y}_M[n] = [y[n], y[n-1], \dots, y[n-M+1]]^T$.
7. Filtrar $\mathbf{y}_M[n]$ por la estimación del camino secundario: $y'_M[n] = \hat{\mathbf{s}} * \mathbf{y}_M[n]$.
8. Calcular la estimación de la señal deseada: $d'[n] = e[n] - y'_M[n]$.
9. Calcular el pseudoerror (señal de error que controla el algoritmo): $e'[n] = y'_f[n] + d'[n]$.
10. Filtrar $\mathbf{x}[n]$ por el filtro adaptativo: $y[n] = \mathbf{w}[n]^T \mathbf{x}[n]$.
11. Actualización de pesos: $\mathbf{w}[n+1] = \mathbf{w}[n] + \mu e'[n] \mathbf{x}_f[n]$.

En este caso, el número de multiplicaciones que se necesita por iteración es de $3L+2M+1$, $L+M$ multiplicaciones más por iteración que en la estructura de filtrado-x convencional, lo que no es de extrañar, puesto que se realizan dos filtrados más, uno de tamaño L y otro de tamaño M . En los sistemas multicanal, este incremento del número de operaciones puede llegar a ser prohibitivo, puesto que los dos filtrados adicionales han de realizarse por cada canal, por lo que el aumento en la velocidad de convergencia obtenido con esta estructura viene penalizado por un incremento en el coste computacional.

Existen versiones que optimizan el número de operaciones en la estructura modificada como la presentada por Douglas en [45], donde mediante manipulaciones matriciales podemos pasar de las $3L+2M+1$ multiplicaciones a $2L+5M+1$, que proporciona ahorro computacional para sistemas donde $L > 3M$, aunque dicha condición raramente se da en sistemas de control activo de ruido. Pueden resultar interesantes adaptar las conclusiones para la estructura modificada, a las que se llegan en [46] para sistema de control activo

de ruido multicanal que usan el algoritmo LMS con la estructura de filtrado-x convencional, puesto que operando para la actualización de coeficientes de forma similar a como se describirá más adelante cuando se hable de los coeficientes auxiliares en los algoritmos de proyección afín (3.5.2), se obtendría ahorro computacional sin pérdida en las propiedades de convergencia de dicho algoritmo.

Esta estructura juega un papel importante en los algoritmos de proyección afín aplicados al control activo de ruido, como se describirá más adelante.

2.7.4. Estructura adjunta

Fue introducida por Wan en 1996 [47] como una alternativa eficiente a la ya conocida de filtrado-x convencional y de menor coste computacional para sistemas multicanal donde existen varias señales de referencia. En este caso, no se usan las estimaciones de las plantas acústicas como en las estructuras anteriores, sino el modelo adjunto o reverso de dichas estimaciones. Dicho modelo sería el dado por:

$$\mathbf{h}' = (h_M, h_{M-1}, h_{M-2}, \dots, h_1)^T. \quad (2.93)$$

Este modelo reverso se utilizaría para filtrar la señal de error que posteriormente intervendrá en la actualización de los coeficientes del filtro adaptativo. El esquema de dicha estructura puede apreciarse en la figura 2.15.

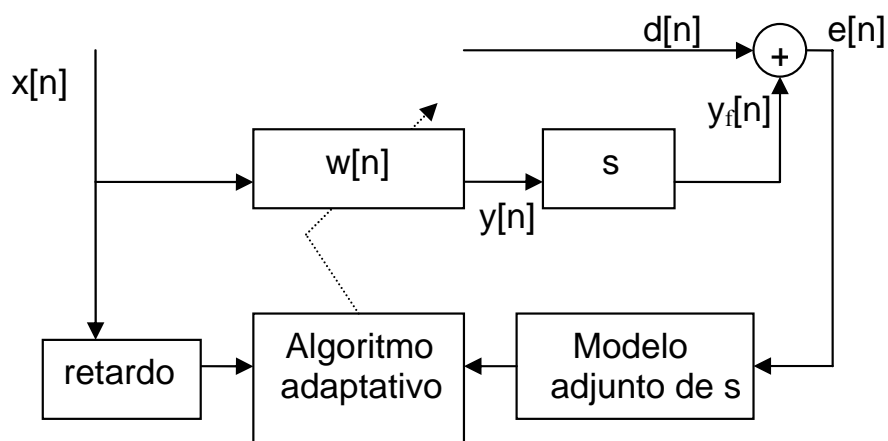


Figura 2.15: Esquema de la estructura adjunta para aplicaciones de CAR.

La nueva señal de error que se usaría en el algoritmo vendrá dada por la expresión:

$$\mathbf{e}'[n - M + 1] = \mathbf{h}'^T \mathbf{e}[n], \quad (2.94)$$

siendo $\mathbf{e}[n]$:

$$\mathbf{e}[n] = [e[n], e[n - 1], e[n - 2], \dots, e[n - M + 1]]^T. \quad (2.95)$$

La ecuación de la actualización de los pesos cuando usamos el algoritmo LMS sería ahora:

$$\mathbf{w}[n + 1] = \mathbf{w}[n] + \mu \mathbf{x}[n - M + 1] \mathbf{e}'[n - M + 1], \quad (2.96)$$

por lo que las operaciones que definirían el algoritmo LMS con la estructura adjunta podrían expresarse del siguiente modo:

1. Obtención de las muestras de la señal primaria $x[n]$ y de la señal $e[n]$.
2. Actualizar el vector de error $\mathbf{e}[n]$ con M muestras de la señal de error.
3. Filtrar $\mathbf{e}[n]$ por la estimación del modelo adjunto del camino secundario: $\mathbf{e}'[n - M + 1] = \mathbf{h}'^T \mathbf{e}[n]$.
4. Filtrar $\mathbf{x}[n]$ por el filtro adaptativo: $y[n] = \mathbf{w}[n]^T \mathbf{x}[n]$.
5. Actualización de pesos: $\mathbf{w}[n + 1] = \mathbf{w}[n] + \mu e'[n - M + 1] \mathbf{x}[n - M + 1]$.

Hay que hacer notar que al filtrar por el reverso del modelo de la planta acústica se está introduciendo un retardo con respecto a la señal de referencia. Por eso es necesario retardar dicha señal antes de introducirla en la ecuación de actualización de los coeficientes del filtro adaptativo el mismo número de muestras que el retardo teórico introducido por la estimación del modelo acústico del reverso del canal. Los resultados que se obtienen con esta estructura, son un poco más deficientes en cuanto a la velocidad de convergencia que el resto de estructuras (sobre todo si no se ajusta bien el retardo). En sistemas monocanal, el número de operaciones a realizar por iteración es el mismo que usando la estructura de filtrado-x ($2L + M + 1$ multiplicaciones), pero cuando tenemos un sistema multicanal con varias señales de referencia, conseguimos un importante ahorro computacional con la estructura adjunta puesto que no tendremos que realizar los filtrados de las señales de referencia por la estimación de los caminos acústicos, sino solo los filtrados de la señal de error por los modelos adjuntos de las estimaciones de los caminos acústicos.

La estructura adjunta, puede considerarse un caso particular de la estructura de filtrado-x con ecualización de la señal de error. Esta estrategia es útil cuando no deseamos cancelar la señal de error, sino eliminar solo una parte de su espectro o ecualizarla. En estos casos, conformamos espectralmente la señal de error residual con una función conformadora (c) que tendría una respuesta en frecuencia inversa a la deseada para el ruido residual y que filtraría la señal de error antes de ser usada en el algoritmo adaptativo [48]. Puesto que dicha función conformadora está entre la señal de error original y el sistema adaptativo, deberíamos de introducirla también en el camino de la señal de referencia hasta el algoritmo adaptativo tal y como aparece en la figura 2.16.

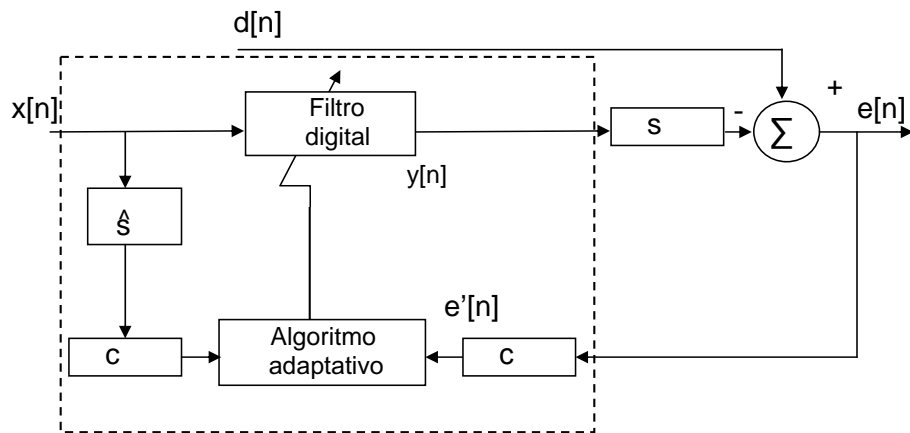


Figura 2.16: Esquema de la estructura de filtrado-x con ecualización para CAR.

En este caso, la señal a minimizar será $e'[n]$ quedando las componentes espectrales de $e[n]$ eliminadas por la respuesta de c sin minimizar.

La estructura adjunta, resultaría del caso particular en el que el filtro $c[n] = \hat{s}[-n]$. Así, la rama procedente de la señal de error estaría formado por el adjunto de la estimación del camino secundario ($\hat{s}[-n]$), pero retardado un número de muestras igual a su tamaño para que sea causal ($\hat{s}[M-1-n]$) y en la rama de la señal de referencia tendríamos la convolución de los sistemas ($\hat{s}[n] * \hat{s}[M-1-n]$), que se traduce en un retardo de $M-1$ muestras.

2.7.5. Elección de la estructura

Al enfrentarnos con un problema de CAR, no resulta indiferente la estructura usada para el mismo. Muchas veces, dicha estructura viene impuesta por el sistema CAR (por ejemplo, imposibilidad de usar filtros inversos del modelo del camino secundario, por lo que se ha de descartar la estructura directa) o

por el algoritmo (necesidad de conocer ciertas señales como la señal deseada $d[n]$ en los algoritmos de proyección afín, para los cuales se hace necesaria inicialmente la estructura modificada) o por la naturaleza de la señal deseada (periódica, aleatoria de carácter estacionario o no estacionario).

Generalmente, y aunque parezca la más intuitiva, la estructura directa es la menos usada para CAR, mientras que si lo que se necesita es un algoritmo con altas prestaciones en cuanto a velocidad de convergencia, la estructura elegida sería la modificada. Si por el contrario, se prefiere un coste computacional bajo sin importar la velocidad de convergencia, la estructura adjunta sería la ideal. La estructura de filtrado-x convencional proporciona una velocidad de convergencia mayor que la adjunta con un coste computacional menor que el de la estructura modificada, por lo que la mayoría de las aplicaciones CAR usan la estructura de filtrado-x convencional.

Capítulo 3

Algoritmos de proyección afín y sus versiones eficientes

3.1. Introducción

El algoritmo de proyección afín se basa en el uso de proyecciones ortogonales en subespacios afines de filtros adaptativos [49] y fue propuesto por Douglas [50] basándose en los trabajos previos de Nagumo y Noda [51], como una extensión del algoritmo LMS normalizado (NLMS) [19] pudiendo encontrar una amplia descripción del mismo en [2].

Como ya se ha comentado anteriormente al hablar del algoritmo LMS (sección 2.4.3), una de las variantes de dicho algoritmo que consigue mejorar sus propiedades de convergencia, estabilidad y funcionamiento en régimen permanente es el LMS normalizado (NLMS). Esta variante pretende independizar dichas propiedades del tamaño del filtro adaptativo usado y de la potencia de la señal de entrada (propiedades que guardan dependencia con estos parámetros en el LMS) y tiene gran utilidad para el control adaptativo de señales con grandes fluctuaciones en sus niveles de potencia.

El algoritmo de proyección afín es, pues, una extensión del NLMS (o el NLMS puede considerarse como un caso particular de este algoritmo) que se constituye en otra variante del LMS mejorada respecto al NLMS en cuanto a propiedades de convergencia, a costa de introducir mayor complejidad de cálculo.

Antes de describir dicho algoritmo, se expondrá brevemente el algoritmo LMS normalizado para posteriormente desarrollar el algoritmo de proyección afín, como una extensión natural de este último.

3.2. Algoritmo LMS normalizado

Antes de nada, convendría recordar el esquema típico de un sistema adaptativo que puede contemplarse en la figura 2.4 en el capítulo 2, junto con la nomenclatura clásica de las señales que intervienen en dichos sistemas.

El algoritmo LMS normalizado debe su nombre a que a la hora de realizar el ajuste de los pesos del filtro adaptativo para la siguiente iteración no se usa la ecuación estándar dada por (3.1), sino que dicha expresión está normalizada por la norma euclídea al cuadrado (estimación de la potencia) de la señal de referencia $\mathbf{x}_L[n]$ tal y como se expresa en (3.2).

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu e[n] \mathbf{x}_L[n] \quad (3.1)$$

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu \frac{e[n] \mathbf{x}_L[n]}{\mathbf{x}_L[n]^T \mathbf{x}_L[n]} = \mathbf{w}_L[n-1] + \mu \frac{e[n] \mathbf{x}_L[n]}{\|\mathbf{x}_L[n]\|^2} \quad (3.2)$$

donde se ha usado el cuadrado de la norma vectorial, siendo $\|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{x}$, y μ representa una constante real positiva denominada constante de convergencia o paso de la iteración, que controla entre otras cosas la velocidad de la convergencia y la estabilidad de la solución alcanzada.

Normalizando según se describe en (3.2) vencemos la problemática que sufre el algoritmo adaptativo LMS de amplificar el error que se comete al estimar el gradiente cuando el número de coeficientes del filtro L (tamaño del vector de la señal de referencia $\mathbf{x}_L[n]$) es grande. Otra forma de ver esta normalización es considerando que el paso de la iteración μ no es constante, sino que depende de la potencia de la señal de referencia. De esta forma, aumentaría la velocidad de convergencia para niveles pequeños de dicha señal y reduciría dicha velocidad para niveles mayores, lo que hace a este algoritmo especialmente adecuado para señales no demasiado estacionarias en su nivel de potencia. Visto de este modo, podríamos usar la misma expresión para la adaptación de los coeficientes en el NLMS que la empleada en el LMS sin más que redefinir la constante de convergencia μ tal y como se describe en (3.3) y (3.4),

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \tilde{\mu}[n] e[n] \mathbf{x}_L[n] \quad (3.3)$$

con

$$\tilde{\mu}[n] = \frac{\mu}{\|\mathbf{x}_L[n]\|^2}. \quad (3.4)$$

Sin embargo, podemos llegar a estas mismas conclusiones definiendo otras condiciones que deba satisfacer el problema y sin necesidad de partir del algoritmo LMS. Más concretamente, podemos obligar al algoritmo a que cumpla el *principio de mínima perturbación* que para el caso que nos ocupa puede

formularse del siguiente modo: “para una señal de referencia $x(n)$ y una señal deseada $d(n)$, el vector de pesos del filtro adaptativo debería variar lo mínimo posible de una iteración a la siguiente imponiéndole que cumpla la restricción de que pueda generar la señal de salida deseada a partir de la de referencia”. Matemáticamente, esto puede expresarse como sigue. Si definimos la variación de los pesos de una iteración a otra tal y como se describe en (3.5)

$$\Delta \mathbf{w}_L[n] = \mathbf{w}_L[n] - \mathbf{w}_L[n-1], \quad (3.5)$$

el problema consistiría en intentar minimizar la expresión (3.6) sujeta a la restricción dada en (3.7)

$$\|\Delta \mathbf{w}_L[n]\|^2 = \Delta \mathbf{w}_L^T[n] \Delta \mathbf{w}_L[n] \quad (3.6)$$

$$\mathbf{w}_L^T[n] \mathbf{x}_L[n] = d[n]. \quad (3.7)$$

Este tipo de problemas es lo que se conoce como un problema de optimización con restricciones y se puede resolver haciendo uso del *método de los multiplicadores de Lagrange* [2], mediante el cual encontrar la solución al problema planteado (para el caso donde solo intervienen señales reales) es equivalente a minimizar la siguiente función de coste:

$$\xi[n] = \|\Delta \mathbf{w}_L[n]\|^2 + \lambda(d[n] - \mathbf{w}_L^T[n] \mathbf{x}_L[n]) \quad (3.8)$$

donde λ es una constante denominada multiplicador de *Lagrange* [2].

Para encontrar el valor óptimo de los pesos actualizados $\mathbf{w}_L[n]$ que minimicen esta nueva función de coste $\xi[n]$ derivamos con respecto a $\mathbf{w}_L[n]$:

$$\frac{\partial \xi[n]}{\partial \mathbf{w}_L[n]} = 2(\mathbf{w}_L[n] - \mathbf{w}_L[n-1]) - \lambda \mathbf{x}_L[n]. \quad (3.9)$$

Igualando a cero la expresión (3.9) llegamos a:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \frac{1}{2} \lambda \mathbf{x}_L[n]. \quad (3.10)$$

Podemos obtener el valor del multiplicador de *Lagrange* sustituyendo la expresión obtenida en (3.10) en la ecuación que expresaba la restricción dada por (3.7) obteniendo:

$$d[n] = (\mathbf{w}_L[n-1] + \frac{1}{2} \lambda \mathbf{x}_L[n])^T \mathbf{x}_L[n] \quad (3.11)$$

$$d[n] = \mathbf{w}_L^T[n-1]\mathbf{x}_L[n] + \frac{1}{2}\lambda\|\mathbf{x}_L[n]\|^2 \quad (3.12)$$

resultando finalmente para λ

$$\lambda = \frac{2e[n]}{\|\mathbf{x}_L[n]\|^2} \quad (3.13)$$

donde

$$e[n] = d[n] - \mathbf{w}_L^T[n-1]\mathbf{x}_L[n] \quad (3.14)$$

es la señal que hemos denominado de error.

Sustituyendo la expresión obtenida para λ en la ecuación (3.10) obtenemos para la actualización de los pesos la siguiente expresión:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \frac{e[n]\mathbf{x}_L[n]}{\|\mathbf{x}_L[n]\|^2}. \quad (3.15)$$

Si introducimos una constante positiva en la ecuación de actualización de los pesos encontrada, con el objeto de ejercer algún tipo de control sobre la velocidad de dicha actualización, resultaría

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu \frac{e[n]\mathbf{x}_L[n]}{\|\mathbf{x}_L[n]\|^2} \quad (3.16)$$

que coincide con la expresión hallada anteriormente en (3.2).

Aunque la expresión propuesta en (3.2) y (3.16) es las que matemáticamente define la actualización de los pesos del filtro adaptativo en el algoritmo NLMS, a veces, sobre todo por cuestiones de implementación práctica, suele modificarse ligeramente dicha expresión introduciendo una pequeña constante positiva en el divisor a sumar con la estimación de la potencia de la señal de referencia:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu \frac{e[n]\mathbf{x}_L[n]}{(\delta + \|\mathbf{x}_L[n]\|^2)}. \quad (3.17)$$

El objetivo de dicha constante δ es evitar los problemas que puedan producirse cuando la señal de referencia $\mathbf{x}_L[n]$ es pequeña y pueda dar lugar a inestabilidades numéricas y errores de precisión. De esta forma acotamos el denominador y evitamos problemas relacionados con la precisión numérica en implementaciones reales de dicho algoritmo.

3.3. Extensión del NLMS hasta el algoritmo de proyección afín.

Como ya hemos comentado anteriormente, el algoritmo de proyección afín (AP) es una extensión del problema de optimización con restricciones propuesto para obtener la expresión de la actualización de los coeficientes en el algoritmo LMS normalizado, donde el problema se reformularía como se describe a continuación.

Hay que intentar minimizar la expresión (3.18) sujeta a las N restricciones dadas en (3.19)

$$\|\Delta \mathbf{w}_L[n]\|^2 = \Delta \mathbf{w}_L^T[n] \Delta \mathbf{w}_L[n] \quad (3.18)$$

$$\mathbf{w}_L^T[n] \mathbf{x}_L[n-k] = d[n-k] \quad \text{para } k = 0, 1, \dots, N-1 \quad (3.19)$$

donde N (orden de la proyección afín) ha de ser menor que L (tamaño del filtro adaptativo y del vector de muestras de entrada $\mathbf{x}_L[n]$). Como resulta fácil observar, el algoritmo NLMS es un caso particular de algoritmo de proyección afín para $N = 1$.

Exactamente igual que para el caso del NLMS, resolver este problema de optimización con las restricciones expuestas en (3.19), puede hacerse mediante el uso del método de los multiplicadores de *Lagrange* y es equivalente a tratar de minimizar la siguiente función de coste:

$$\xi[n] = \|\Delta \mathbf{w}_L[n]\|^2 + \sum_{k=0}^{N-1} \lambda_k (d[n-k] - \mathbf{w}_L^T[n] \mathbf{x}_L[n-k]). \quad (3.20)$$

En dicha expresión, cada λ_k representa el multiplicador de *Lagrange* asociado a cada una de las N restricciones.

La expresión (3.20) puede reescribirse de forma matricial según:

$$\xi[n] = \|\Delta \mathbf{w}_L[n]\|^2 + (\mathbf{d}_N[n] - \mathbf{A}[n] \mathbf{w}_L[n])^T \underline{\lambda} \quad (3.21)$$

donde $\mathbf{A}[n]$ es una matriz de dimensión $N \times L$ definida como:

$$\mathbf{A}^T[n] = (\mathbf{x}_L[n], \mathbf{x}_L[n-1], \dots, \mathbf{x}_L[n-N+1]). \quad (3.22)$$

$\mathbf{d}_N[n]$ representa el vector de muestras de la señal deseada de tamaño $N \times 1$

$$\mathbf{d}_N^T[n] = (d[n], d[n-1], \dots, d[n-N+1]), \quad (3.23)$$

y $\underline{\lambda}$ el vector de multiplicadores de *Lagrange* asociados a cada restricción del problema:

$$\underline{\lambda}^T = (\lambda_0, \lambda_1, \dots, \lambda_{N-1}). \quad (3.24)$$

Derivando la expresión (3.21) respecto a $\mathbf{w}_L(n)$ obtenemos:

$$\frac{\partial \xi[n]}{\partial \mathbf{w}_L[n]} = 2(\mathbf{w}_L[n] - \mathbf{w}_L[n-1]) - \mathbf{A}^T[n]\underline{\lambda} \quad (3.25)$$

e igualando a cero

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \frac{1}{2}\mathbf{A}^T[n]\underline{\lambda}. \quad (3.26)$$

Una vez llegado a este punto, lo único que falta para obtener una expresión compacta de la actualización de los pesos es eliminar el vector $\underline{\lambda}$ de los multiplicadores de *Lagrange*. Si reescribimos la ecuación (3.19) de forma matricial, obtenemos:

$$\mathbf{d}_N[n] = \mathbf{A}[n]\mathbf{w}_L[n] \quad (3.27)$$

y haciendo uso de (3.26) llegamos a la expresión:

$$\mathbf{d}_N[n] = \mathbf{A}[n]\mathbf{w}_L[n-1] + \frac{1}{2}\mathbf{A}[n]\mathbf{A}^T[n]\underline{\lambda} \quad (3.28)$$

de donde finalmente se obtiene:

$$\underline{\lambda} = 2(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N[n] \quad (3.29)$$

siendo $\mathbf{e}_N(n)$ un vector de tamaño $N \times 1$ con las N últimas diferencias entre las muestras de la señal deseada y la generada

$$\mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n-1]. \quad (3.30)$$

Sustituyendo el valor de $\underline{\lambda}$ obtenido en (3.29) en la ecuación (3.26) se obtiene finalmente para la actualización de los pesos en el algoritmo de proyección afín la expresión:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N[n]. \quad (3.31)$$

Al igual que en el caso del NLMS, suele introducirse una constante μ para tener control sobre la actualización de los pesos, quedando:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N[n]. \quad (3.32)$$

Por último, debemos mencionar que en algunas ocasiones, la inversión de la matriz que aparece en (3.32) $(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}$, puede provocar imprecisiones numéricas debido a que puede estar mal condicionada (solo depende de

la señal de referencia). Es algo parecido a lo que podía pasar en el NLMS cuando la señal de referencia tenía valores pequeños. Para intentar solucionar este problema se procede de forma análoga a lo realizado en el caso del LMS normalizado [52], sumando a la diagonal de la matriz a invertir una constante positiva δ que mejore las propiedades de dicha matriz a la hora de la inversión [2]. Esto es lo que se conoce como regularización (proceso del cual se han realizado estudios recientes para optimizar el valor de la constante δ como en [53] o en [54] o incluso para realizar este proceso de forma adaptativa [55]). Existen procesos alternativos a la regularización [56] pero que no serán tratados en este trabajo. La expresión final de la actualización de los pesos con regularización sería:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1} \mathbf{e}_N[n] \quad (3.33)$$

donde \mathbf{I} es la matriz identidad de orden $N \times N$.

La ecuación (3.33) será la ecuación de actualización de los coeficientes del algoritmo adaptativo bajo estudio en esta tesis doctoral, y que como se ha mencionado, denominaremos algoritmo de proyección afín (AP). No obstante, en algunas ocasiones suele definirse una ecuación de actualización de los pesos más genérica que define a toda la familia de algoritmos de proyección afín, tal y como se expresa en (3.34)

$$\mathbf{w}_L[n] = \mathbf{w}_{L\alpha(N-1)}[n-1] + \mu \mathbf{A}_\tau^T[n](\mathbf{A}_\tau[n]\mathbf{A}_\tau^T[n] + \delta I)^{-1} \mathbf{e}_{N\tau}[n] \quad (3.34)$$

con $\mathbf{e}_{N\tau}[n] = \mathbf{d}_{N\tau}[n] - \mathbf{A}_\tau \mathbf{w}_L[n-1 - \alpha(N-1)]$,

$$\mathbf{A}_\tau^T[n] = (\mathbf{x}_L[n], \mathbf{x}_L[n-\tau], \dots, \mathbf{x}_L[n-(N-1)\tau]) \quad (3.35)$$

y

$$\mathbf{d}_{N\tau}^T[n] = (d[n], d[n-\tau], \dots, d[n-(N-1)\tau]). \quad (3.36)$$

Debe considerarse que los algoritmos de proyección afín se diferencian del algoritmo LMS en que la actualización de los pesos del filtro adaptativo se basa en N vectores de datos previos de la señal de referencia (sin tener que ser justo los N vectores de datos anteriores) en vez de solo en el actual vector de datos de la señal de referencia. Dependiendo de cómo elegir dichos vectores y cómo usarlos en la ecuación de actualización definida en (3.34), se pueden considerar diversas variantes de esta familia de algoritmos como los algoritmos NLMS con factores de corrección ortogonal (NLMS-OCF) [57], los algoritmos de proyección afín de rango parcial (PRA) [58], o los empleados con mayor frecuencia, llamados algoritmos de proyección afín estándar (APA), que se obtienen cuando $\delta = 0$, $\alpha = 0$ y $\tau = 1$. Por último, también podemos encontrar los algoritmos de proyección afín con regularización (R-APA) [59], definidos cuando $\delta = 1$, $\alpha = 0$ y $\tau = 1$ y que serán los estudiados en este trabajo, y que se denominarán a partir de ahora simplemente algoritmos AP.

3.4. Optimización computacional de los AP

Al igual que los filtros de mínimos cuadrados admiten versiones eficientes desde el punto de vista computacional (apendice B) que sobre todo tratan de minimizar el impacto que provoca el cálculo de la inversión matricial, los algoritmos de proyección afín también pueden implementarse mediante versiones que tratan de optimizar el coste computacional, especialmente para el cálculo de la matriz inversa que aparece en la actualización de los coeficientes de dichos algoritmos (ecuaciones (3.32) ó (3.33) mostradas anteriormente). Estos algoritmos más eficientes suelen denominarse algoritmos FAP (Fast Affine Projection) [60] y se valen de aproximaciones y de la estructura particular que tienen las matrices de datos para reducir el número de operaciones en la ejecución del algoritmo.

3.4.1. Cálculo eficiente de la matriz inversa

Lo primero que se nos puede ocurrir para optimizar el coste computacional es intentar reducir el número de operaciones necesarias en el cálculo de la inversión matricial que aparece en la actualización de los coeficientes del algoritmo de proyección afín. Esto puede hacerse usando el lema de inversión matricial que ya se presentó en el capítulo 2 al hablar del algoritmo RLS.

Así, si llamamos $\mathbf{M}[n]$, a la matriz resultado del producto de la matriz de datos $\mathbf{A}[n]$ por su traspuesta ($\mathbf{M}[n] = \mathbf{A}[n]\mathbf{A}^T[n]$), podemos darnos cuenta que se cumple la siguiente relación:

$$\mathbf{M}[n] = \mathbf{M}[n-1] + \mathbf{x}_N[n]\mathbf{x}_N^T[n] - \mathbf{x}_N[n-L]\mathbf{x}_N^T[n-L] \quad (3.37)$$

siendo

$$\mathbf{x}_N[n] = (x[n], x[n-1], \dots, x[n-N+1])^T. \quad (3.38)$$

Esta expresión recursiva para calcular $\mathbf{M}[n]$ nos ahorraría operaciones antes de realizar la inversión, pero no evitaría tener que calcular $\mathbf{M}^{-1}[n]$, que es lo que necesitamos en la actualización de los pesos en los algoritmos de proyección afín.

Pero si recordamos que el lema de inversión de matrices afirma que dadas dos matrices \mathbf{A} y \mathbf{B} , definidas positivamente, de dimensiones $M \times M$ que cumplen la siguiente relación:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T, \quad (3.39)$$

entonces podemos calcular \mathbf{A}^{-1} como:

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^T\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{B} \quad (3.40)$$

donde \mathbf{D} y \mathbf{C} son otras dos matrices ($N \times N$) y ($M \times N$), respectivamente.

Volviendo a la expresión recursiva para el cálculo de $\mathbf{M}[n]$ (ecuación (3.37)), y manipulando adecuadamente las expresiones, podemos escribir:

$$\mathbf{M}[n] = \mathbf{M}[n-1] + \mathbf{F}[n]\mathbf{D}\mathbf{F}^T[n], \quad (3.41)$$

siendo $\mathbf{F}[n] = (\mathbf{x}_N[n], \mathbf{x}_N[n-L])$ de dimensiones $N \times 2$, y $\mathbf{D} = \mathbf{D}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

De esta forma podemos identificar:

$$\mathbf{A} = \mathbf{M}[n] \quad (3.42)$$

$$\mathbf{B}^{-1} = \mathbf{M}[n-1] \quad (3.43)$$

$$\mathbf{C} = \mathbf{F}[n] \quad (3.44)$$

$$\mathbf{D}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (3.45)$$

con lo que podemos usar el lema de inversión de matrices para calcular $\mathbf{M}^{-1}[n]$, quedando:

$$\begin{aligned} \mathbf{M}^{-1}[n] &= \mathbf{M}^{-1}[n-1] \\ &\quad - \mathbf{M}^{-1}[n-1]\mathbf{F}[n](\mathbf{D} + \mathbf{F}^T[n]\mathbf{M}^{-1}[n-1]\mathbf{F}[n])^{-1}\mathbf{F}^T[n]\mathbf{M}^{-1}[n-1]. \end{aligned} \quad (3.46)$$

Usando (3.46) podemos calcular la inversa de la matriz $\mathbf{M}[n]$ a partir de la inversa de dicha matriz en la iteración anterior más unas operaciones matriciales que involucran otro cálculo de una inversa, pero de una matriz 2×2 , por lo que prácticamente se reduce a la evaluación de un determinante. Evidentemente, el ahorro computacional usando esta estrategia será mayor conforme el orden de la matriz original a invertir sea más elevado (orden de proyección más grande). Mientras que realizando la inversión matricial, la ejecución del algoritmo de proyección afín para órdenes elevados resultaba impracticable ($O(N^3/2)$ multiplicaciones), usando el cálculo recursivo, dicho coste computacional no sería tan elevado ($6(N^2 + 2N)$ multiplicaciones).

En la ecuación (3.46), podemos darnos cuenta de que el cálculo iterativo de la inversa involucra un conjunto de operaciones matriciales que para órdenes pequeños del algoritmo de proyección afín y pocos coeficientes en los filtros adaptativos, no produce ninguna mejoría en el coste computacional respecto a realizarla de forma directa. Además, el coste computacional del cálculo iterativo de la inversa, solo depende del orden de proyección, manteniéndose fijo

con la variación del número de coeficientes L , lo que resulta muy interesante si queremos ejecutar este algoritmo para filtros con muchos coeficientes.

Es cierto, que puesto que es un método iterativo, necesitamos partir de un valor conocido de la inversa de la matriz a calcular. Esto podría realizarse en una primera etapa de inicialización (*setup*) del algoritmo, de forma que el cálculo iterativo de la inversa resulte idéntico al exacto. Sin embargo, se ha podido comprobar de forma práctica, que eligiendo una buena estimación de dicha inversa como aproximación inicial, el algoritmo sigue funcionando correctamente (claro está que funciona mejor conforme el valor de la estimación elegida sea más parecido a su valor real). En la práctica se suele usar como valor inicial de la matriz inversa a calcular una constante multiplicada por la matriz identidad de tamaño $N \times N$, consiguiendo resultados bastante buenos, comparables con los obtenidos calculando la inversa matricial de forma directa. La descripción y análisis de esta estrategia para el algoritmo AP aplicado al control activo multicanal puede encontrarse en [61].

3.5. Algoritmo eficiente de proyección afín (FAP)

El algoritmo eficiente de proyección afín fue desarrollado inicialmente para aplicaciones de cancelación de eco por Gay ([60], [62]) y posteriormente para aplicaciones de control activo de ruido por Douglas [63] y Bouchard [64]. Una descripción detallada del mismo podemos encontrarla en [59].

Basándonos en lo expuesto en el capítulo 3, el algoritmo de proyección afín introduciendo relajación (constante de convergencia μ) y regularización ($\delta \neq 0$) podría describirse según las siguientes ecuaciones:

$$\epsilon_N[n] = (\mathbf{A}[n]\mathbf{A}^T[n] + \delta\mathbf{I})^{-1}\mathbf{e}_N[n] \quad (3.47)$$

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu\mathbf{A}^T[n]\epsilon_N[n] \quad (3.48)$$

donde $\epsilon_N[n]$ es lo que se conoce como vector de error normalizado y $\mathbf{e}_N[n]$ es el vector de error ya introducido en (3.30).

Aunque existen algunas versiones más complejas [65], la implementación del algoritmo FAP aprovecha básicamente tres vías de optimización para mejorar la eficiencia computacional, que son, el cálculo eficiente de la actualización de los coeficientes, el cálculo eficiente del vector de error y el cálculo eficiente del vector de error normalizado. Veremos uno por uno cómo se implementan dichos cálculos.

3.5.1. Cálculo eficiente del vector de error

En muchas ocasiones, el vector de error suele ser un dato del problema (como sucede en la aplicación del algoritmo de proyección afín al control activo de ruido con la estructura convencional de filtrado-x, que se verá posteriormente). Sin embargo, también puede considerarse como un vector a calcular tal y como se describe en (3.49), pudiendo escribir:

$$\begin{aligned} \mathbf{e}_N[n] &= \mathbf{d}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n-1] = \begin{pmatrix} d[n] - \mathbf{x}_L^T[n]\mathbf{w}_L[n-1] \\ \tilde{\mathbf{d}}[n] - \tilde{\mathbf{A}}[n]\mathbf{w}_L[n-1] \end{pmatrix} \\ &= \begin{pmatrix} d[n] - \mathbf{x}_L^T[n]\mathbf{w}_L[n-1] \\ \bar{\mathbf{d}}[n-1] - \bar{\mathbf{A}}[n-1]\mathbf{w}_L[n-1] \end{pmatrix} \end{aligned} \quad (3.49)$$

donde $\tilde{\mathbf{d}}[n]$ es el vector de tamaño $(N-1) \times 1$ que contiene los $N-1$ elementos inferiores (más antiguos) del vector $\mathbf{d}_N[n]$ y $\tilde{\mathbf{A}}[n]$ es una matriz de tamaño $(N-1) \times L$ que contiene las últimas $N-1$ filas de la matriz $\mathbf{A}[n]$, mientras que $\bar{\mathbf{d}}_N[n-1]$ y $\bar{\mathbf{A}}[n-1]$ hacen referencia a las primeras $N-1$ filas de $\mathbf{d}_L[n-1]$ y $\mathbf{A}[n-1]$.

Si definimos un nuevo vector error llamado $\mathbf{e}'_N[n-1]$ de la siguiente manera:

$$\mathbf{e}'_N[n-1] = \mathbf{d}_N[n-1] - \mathbf{A}[n-1]\mathbf{w}_L[n-1] \quad (3.50)$$

entonces, introduciendo el valor recursivo para la actualización de los coeficientes calculada para el algoritmo de proyección afín dada en (3.33) tendremos que:

$$\begin{aligned} &\mathbf{e}'_N[n-1] \\ &= \mathbf{d}_N[n-1] - \mathbf{A}[n-1]\{\mathbf{w}_L[n-2] + \mu\mathbf{A}^T[n-1](\mathbf{A}[n-1]\mathbf{A}^T[n-1] \\ &\quad + \delta I)^{-1}\mathbf{e}_N[n-1]\} \\ &= \mathbf{e}_N[n-1] - \mu\mathbf{A}[n-1]\mathbf{A}^T[n-1](\mathbf{A}[n-1]\mathbf{A}^T[n-1] + \delta I)^{-1}\mathbf{e}_N[n-1]. \end{aligned} \quad (3.51)$$

Si consideramos la aproximación $\mathbf{A}[n-1]\mathbf{A}^T[n-1] + \delta I \approx \mathbf{A}[n-1]\mathbf{A}^T[n-1]$, lo que suele estar justificado cuando el valor de la constante de regularización δ es significativamente más pequeño que los autovalores de $\mathbf{A}[n-1]\mathbf{A}^T[n-1]$, tendremos que:

$$\mathbf{e}'_N[n-1] \approx \mathbf{e}_N[n-1] - \mu\mathbf{e}_N[n-1] = (1 - \mu)\mathbf{e}_N[n-1]. \quad (3.52)$$

Así, el cálculo del vector de error $\mathbf{e}_N[n]$ quedaría:

$$\begin{aligned} \mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n-1] &= \begin{pmatrix} d[n] - \mathbf{x}_L^T[n]\mathbf{w}_L[n-1] \\ (1-\mu)\bar{\mathbf{e}}_N[n-1] \end{pmatrix} \\ &= \begin{pmatrix} e[n] \\ (1-\mu)\bar{\mathbf{e}}_N[n-1] \end{pmatrix}, \end{aligned} \quad (3.53)$$

donde $\bar{\mathbf{e}}_N[n-1]$ representa los $N-1$ elementos superiores (más recientes) del vector $\mathbf{e}_N[n-1]$. Usando esta forma de calcular el vector de error, reducimos el número de multiplicaciones de $L \times N$ hasta $(N-1)$.

De la expresión obtenida en (3.53) podríamos concluir que para el caso en el que la constante de paso es igual a la unidad ($\mu = 1$), tendríamos que

$$\mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n-1] = \begin{pmatrix} e[n] \\ \mathbf{0} \end{pmatrix} \quad (3.54)$$

siendo $\mathbf{0}$ un vector columna de $N-1$ ceros.

Esto simplificaría mucho la ecuación de actualización de los coeficientes del algoritmo de proyección afín (3.33).

Además, se puede demostrar que

$$\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1} = (\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{A}^T[n] \quad (3.55)$$

por lo que haciendo uso de la ecuación (3.54), podríamos escribir para la actualización de los coeficientes:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{x}_L[n]e[n], \quad (3.56)$$

ecuación que es similar a la encontrada en (2.39) al resolver el problema de mínimos cuadrados de forma recursiva. Así pues, estas aproximaciones conducen a algo parecido a un algoritmo RLS (en el caso de usar un orden de proyección igual al índice temporal n y $\delta = 0$, las ecuaciones (3.56) y (2.39) serían idénticas).

El cálculo eficiente del vector de error es usado por Bouchard en sus algoritmos FAP [64] y FAP-GS [66] reduciendo el coste computacional al usar o calcular solamente los elementos de la matriz $(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}$ que luego se usan al multiplicar por el vector de error dado en (3.54). O lo que es lo mismo, solo se usa o calcula una fila de dicha matriz.

3.5.2. Cálculo eficiente de la actualización de los coeficientes

La técnica del cálculo eficiente de los coeficientes, permite usar y actualizar otros coeficientes diferentes, llamados coeficientes auxiliares ($\hat{\mathbf{w}}_L[n]$) [59] en

vez de los habituales ($\mathbf{w}_L[n]$), pero sin que las señales generadas ($y[n]$) se vean afectadas. Así que en el algoritmo trabajaremos con los coeficientes auxiliares mientras que para generar las señales usaremos las relaciones existentes entre los coeficientes auxiliares y los habituales, pero nunca llegaremos a calcular ni a disponer de los coeficientes normales. El proceso de actualización de los coeficientes auxiliares tiene un coste computacional equivalente al del LMS normalizado.

Para calcular la nueva recursión, partimos de la expresión original de actualización de los coeficientes del algoritmo AP dada en (3.48). Si expresamos los coeficientes de la iteración n -ésima en función de los datos y de los coeficientes desde el estado inicial, tendremos:

$$\mathbf{w}_L[n] = \mathbf{w}_L[0] + \mu \sum_{i=0}^{n-1} \mathbf{A}^T[n-i] \epsilon_N[n-i]. \quad (3.57)$$

La ecuación (3.57) puede reescribirse desglosando el sumatorio:

$$\mathbf{w}_L[n] = \mathbf{w}_L[0] + \mu \sum_{i=0}^{n-1} \sum_{j=0}^{N-1} \mathbf{x}_L[n-i-j] \epsilon_{N_j}[n-i] \quad (3.58)$$

donde $\mathbf{x}_L[n-i-j]$ coincide con la j -ésima columna de la matriz $\mathbf{A}^T[n-i]$ y $\epsilon_{N_j}[n-i]$ representa al j -ésimo elemento de $\epsilon_N[n-i]$. Considerando la hipótesis de enventanado ($\mathbf{x}_L[k]$ es distinto de cero solo para $1 \leq k \leq n$) entonces, podemos multiplicar el segundo término (3.58) por una ventana rectangular Π_1 sin que se altere el resultado:

$$\mathbf{w}_L[n] = \mathbf{w}_L[0] + \mu \sum_{i=0}^{n-1} \sum_{j=0}^{N-1} \mathbf{x}_L[n-i-j] \epsilon_{N_j}[n-i] \Pi_1 \quad (3.59)$$

con

$$\Pi_1 = \begin{cases} 1, & 0 \leq j+i \leq n-1 \\ 0, & \text{en otro caso} \end{cases} \quad (3.60)$$

Si aplicamos el cambio de variables $i = k - j$ e intercambiamos el orden de los sumatorios, podemos escribir:

$$\mathbf{w}_L[n] = \mathbf{w}_L[0] + \mu \sum_{j=0}^{N-1} \sum_{k=j}^{n-1+j} \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j] \Pi'_1 \quad (3.61)$$

siendo ahora

$$\Pi'_1 = \begin{cases} 1, & 0 \leq k \leq n-1 \\ 0, & \text{en otro caso} \end{cases} \quad (3.62)$$

por lo que la ecuación (3.61) puede reformularse como

$$\mathbf{w}_L[n] = \mathbf{w}_L[0] + \mu \sum_{j=0}^{N-1} \sum_{k=j}^{n-1} \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j]. \quad (3.63)$$

A continuación, dividimos el segundo sumatorio de (3.63) en dos partes, una para $j \leq k \leq N-1$ y otra para $N \leq k \leq n-1$, quedando:

$$\begin{aligned} \mathbf{w}_L[n] = \mathbf{w}_L[0] + \mu \sum_{j=0}^{N-1} \sum_{k=j}^{N-1} \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j] \\ + \mu \sum_{j=0}^{N-1} \sum_{k=N}^{n-1} \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j]. \end{aligned} \quad (3.64)$$

Si operamos solamente con el primer doble sumatorio de (3.64) y definimos otra ventana rectangular de la forma:

$$\Pi_2 = \left\{ \begin{array}{ll} 1, & 0 \leq k-j \\ 0, & \text{en otro caso,} \end{array} \right\} \quad (3.65)$$

podemos introducir esta segunda ventana en el primer doble sumatorio de (3.64) ampliando el índice inferior desde $k = j$ hasta $k = 0$ sin alterar el resultado,

$$\sum_{j=0}^{N-1} \sum_{k=j}^{N-1} \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j] = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j] \Pi_2. \quad (3.66)$$

Teniendo en cuenta el valor de la ventana definida en (3.65), podemos intercambiar el orden de los sumatorios y limitar el índice superior del primer sumatorio en (3.66) a $j = k$ quedando:

$$\begin{aligned} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j] \Pi_2 &= \sum_{k=0}^{N-1} \sum_{j=0}^k \mathbf{x}_L[n-k] \epsilon_{N_j}[n-k+j] \\ &= \sum_{k=0}^{N-1} \mathbf{x}_L[n-k] \sum_{j=0}^k \epsilon_{N_j}[n-k+j], \end{aligned} \quad (3.67)$$

llegando finalmente a la expresión que nos va a servir de punto de partida para obtener la recursión para el cálculo de los coeficientes auxiliares. Llevando

(3.67) a (3.64):

$$\begin{aligned} \mathbf{w}_L[n] = & \mathbf{w}_L[0] + \mu \sum_{k=0}^{N-1} \mathbf{x}_L[n-k] \sum_{j=0}^k \epsilon_{N_j}[n-k+j] \\ & + \mu \sum_{k=N}^{n-1} \mathbf{x}_L[n-k] \sum_{j=0}^{N-1} \epsilon_{N_j}[n-k+j], \end{aligned} \quad (3.68)$$

donde se ha intercambiado el orden del segundo doble sumatorio en (3.64) permitiendo sacar fuera del segundo sumatorio el término que es independiente de la variable j .

Así podemos definir los coeficientes auxiliares $\widehat{\mathbf{w}}_L[n-1]$ agrupando el primer y tercer término de la derecha en (3.68):

$$\widehat{\mathbf{w}}_L[n-1] = \mathbf{w}_L[0] + \mu \sum_{k=N}^{n-1} \mathbf{x}_L[n-k] \sum_{j=0}^{N-1} \epsilon_{N_j}[n-k+j], \quad (3.69)$$

mientras que el término restante de la ecuación (3.68) puede considerarse como una multiplicación de una matriz por un vector:

$$\sum_{k=0}^{N-1} \mathbf{x}_L[n-k] \sum_{j=0}^k \epsilon_{N_j}[n-k+j] = \mathbf{A}^T[n] \mathbf{e}_{aux}[n] \quad (3.70)$$

donde

$$\mathbf{e}_{aux}[n] = \begin{pmatrix} \epsilon_{N_0}[n] \\ \epsilon_{N_1}[n] + \epsilon_{N_0}[n-1] \\ \epsilon_{N_2}[n] + \epsilon_{N_1}[n-1] + \epsilon_{N_0}[n-2] \\ \vdots \\ \epsilon_{N_{N-1}}[n] + \epsilon_{N_{N-2}}[n-1] + \cdots + \epsilon_{N_1}[n-N+2] + \epsilon_{N_0}[n-N+1] \end{pmatrix}. \quad (3.71)$$

Además, el vector $\mathbf{e}_{aux}[n]$ (al que llamaremos *vector de error auxiliar*) puede calcularse recursivamente de una forma sencilla ya que:

$$\mathbf{e}_{aux}[n] = \epsilon_N[n] + \begin{pmatrix} 0 \\ \bar{\mathbf{e}}_{aux}[n-1] \end{pmatrix}, \quad (3.72)$$

siendo $\bar{\mathbf{e}}_{aux}[n]$ el vector de tamaño $N-1$ que contiene los primeros $N-1$ elementos (elementos más recientes) de \mathbf{e}_{aux} .

De la ecuación (3.69) se puede deducir que:

$$\begin{aligned}\widehat{\mathbf{w}}_L[n] &= \widehat{\mathbf{w}}_L[n-1] + \mu \mathbf{x}_L[n-N+1] \sum_{j=0}^{N-1} \epsilon_{N_j}[n-N+1+j] \\ &= \widehat{\mathbf{w}}_L[n-1] + \mu \mathbf{x}_L[n-N+1] E_{N-1}[n]\end{aligned}\quad (3.73)$$

donde $E_{N-1}[n]$ es el último elemento del vector $\mathbf{e}_{aux}[n]$, con lo cual tendríamos la ecuación recursiva buscada para los coeficientes auxiliares. Sin embargo, estos coeficientes no nos servirían para generar las señales deseadas y tenemos que buscar la relación existente entre los coeficientes auxiliares ($\widehat{\mathbf{w}}_L[n]$) y los verdaderos coeficientes de los filtros adaptativos ($\mathbf{w}_L[n]$).

Volviendo ahora a la expresión (3.68) e introduciendo en la misma las relaciones calculadas en (3.69) y (3.70), podemos escribir:

$$\mathbf{w}_L[n] = \widehat{\mathbf{w}}_L[n-1] + \mu \mathbf{A}^T[n] \mathbf{e}_{aux}[n]. \quad (3.74)$$

Despejando el valor de $\widehat{\mathbf{w}}_L[n-1]$ en la ecuación (3.73) e introduciéndolo en (3.74) tendremos:

$$\begin{aligned}\mathbf{w}_L[n] &= \widehat{\mathbf{w}}_L[n] - \mu \mathbf{x}_L[n-N+1] E_{N-1}[n] + \mu \mathbf{A}^T[n] \mathbf{e}_{aux}[n] = \\ &\quad \widehat{\mathbf{w}}_L[n] + \mu \overline{\mathbf{A}}^T[n] \overline{\mathbf{e}}_{aux}[n]\end{aligned}\quad (3.75)$$

donde $\overline{\mathbf{A}}[n]$ es una matriz $(N-1) \times L$ que contiene la parte superior de la matriz $\mathbf{A}[n]$, y $\overline{\mathbf{e}}_{aux}$ contiene los $N-1$ elementos más recientes (superiores) del vector \mathbf{e}_{aux} .

De esta forma, la señal a generar se calcularía:

$$y[n] = \mathbf{x}_L^T[n] \mathbf{w}_L[n] = \mathbf{x}_L^T[n] \widehat{\mathbf{w}}_L[n] + \mu \mathbf{x}_L^T[n] \overline{\mathbf{A}}^T[n] \overline{\mathbf{e}}_{aux}[n], \quad (3.76)$$

por lo tanto, la señal $y[n]$ sería la generada por el filtro formado por los coeficientes auxiliares ($\widehat{y}[n] = \mathbf{x}_L^T[n] \widehat{\mathbf{w}}_L[n]$) más un término que contiene información de la señal de error:

$$y[n] = \widehat{y}[n] + \mu \overline{\mathbf{r}}[n] \overline{\mathbf{e}}_{aux}[n]. \quad (3.77)$$

A primera vista parece que las operaciones que nos ahorramos en el cálculo recursivo de los coeficientes auxiliares, las necesitaremos después para generar las señales deseadas. Sin embargo, el vector $\overline{\mathbf{r}}[n] = \mathbf{x}_L^T[n] \overline{\mathbf{A}}^T[n]$ de dimensiones $1 \times (N-1)$, puede calcularse también de forma recursiva como a continuación se demuestra, considerando que:

$$\bar{\mathbf{r}}[n] = \mathbf{x}_L^T[n] \bar{\mathbf{A}}^T[n] = (\mathbf{x}_L^T[n] \mathbf{x}_L[n], \mathbf{x}_L^T[n] \mathbf{x}_L[n-1], \dots, \mathbf{x}_L^T[n] \mathbf{x}_L[n-N+2]) \quad (3.78)$$

y

$$\begin{aligned} \bar{\mathbf{r}}[n-1] &= \mathbf{x}_L^T[n-1] \bar{\mathbf{A}}^T[n-1] \\ &= (\mathbf{x}_L^T[n-1] \mathbf{x}_L[n-1], \mathbf{x}_L^T[n-1] \mathbf{x}_L[n-2], \dots, \mathbf{x}_L^T[n-1] \mathbf{x}_L[n-N+1]) \end{aligned} \quad (3.79)$$

por lo que por simple inspección de (3.78) y (3.79), podemos decir que:

$$\bar{\mathbf{r}}[n] = \bar{\mathbf{r}}[n-1] + x[n] \bar{\mathbf{x}}_N^T[n] - x[n-L] \bar{\mathbf{x}}_N^T[n-L] \quad (3.80)$$

donde $\bar{\mathbf{x}}_N^T[n]$ es el vector de dimensiones $(N-1) \times 1$ con las $N-1$ muestras superiores (más recientes) de $\mathbf{x}_N^T[n]$ definido en (3.38).

Así, un sencillo algoritmo FAP que use la estrategia de los coeficientes auxiliares quedaría de la siguiente forma:

1. Obtención de las muestras de la señal de entrada $x[n]$ y de la señal de error $e[n]$.
2. Actualizar los vectores de datos de entrada $\mathbf{x}_L[n]$, $\mathbf{x}_N[n]$ y la matriz de datos de entrada $\mathbf{A}[n]$.
3. Calcular el vector $\bar{\mathbf{r}}[n] = \bar{\mathbf{r}}[n-1] + x[n] \bar{\mathbf{x}}_N^T[n] - x[n-L] \bar{\mathbf{x}}_N^T[n-L]$.
4. Calcular el vector de error auxiliar $\mathbf{e}_{aux}[n] = \epsilon_N[n] + \begin{pmatrix} 0 \\ \bar{\mathbf{e}}_{aux}[n-1] \end{pmatrix}$.
5. Actualizar los coeficientes auxiliares $\hat{\mathbf{w}}_L[n] = \hat{\mathbf{w}}_L[n-1] + \mu \mathbf{x}_L[n-N+1] E_{N-1}[n]$.
6. Calcular la salida $y[n] = \hat{\mathbf{w}}_L^T[n] \mathbf{x}_L[n] + \mu \bar{\mathbf{r}}[n] \bar{\mathbf{e}}_{aux}[n]$.

El ahorro computacional que se introduce con el uso de los coeficientes auxiliares es de $(L-3)N+2$ multiplicaciones, como puede comprobarse en [67].

Con las ecuaciones definidas, podemos generar las señales deseadas usando los coeficientes auxiliares siempre que tengamos un método eficiente para calcular el valor del vector de error normalizado $\epsilon_N[n]$ que aparece en la ecuación de actualización del vector de error auxiliar (3.72). Para ello, también existen métodos eficientes como se expondrán a continuación.

3.5.3. Cálculo eficiente del vector de error normalizado

El cálculo del vector de error normalizado involucra la siempre molesta inversión matricial que aparece en (3.47). En la mayoría de los casos (órdenes de proyección mayores que 3), es el cálculo de esta inversión matricial la que consume la mayor parte del coste computacional del algoritmo, puesto que dicho coste suele ser del orden de N^3 multiplicaciones. Para realizar dicha inversión matricial, podemos optar por cualquiera de las estrategias propuestas en el caso del problema LS con enventanado deslizante que se describe en el anexo B (que proporciona matrices $\mathbf{R}[n]$ a invertir similares a las que surgen en los algoritmos de proyección afín). Otra forma de realizar esta inversión matricial de forma eficiente ya se presentó cuando se habló del cálculo eficiente de la matriz inversa haciendo uso del lema de inversión de matrices que es una estrategia similar a la resolución del problema mediante RLS con enventanado deslizante. Precisamente, la estrategia RLS con enventanado deslizante es usada en [64] para implementar un algoritmo de proyección afín rápido multicanal aplicado al control activo de ruido.

A continuación se mostrará otra estrategia eficiente basada en la metodología usada para resolver el problema de mínimos cuadrados con ventana deslizante (anexo B). Esta estrategia surge dado que la matriz $\mathbf{R}_N[n]$ de los algoritmos FAP con orden de proyección N , y L coeficientes en los filtros adaptativos, es idéntica a la matriz $\mathbf{R}_N[n]$ que aparece en los algoritmos de mínimos cuadrados con enventanado deslizante de tamaño L y filtros adaptativos de N coeficientes.

Así, llamando $\mathbf{R}_N[n]$ a la matriz a invertir:

$$\mathbf{R}_N[n] = (\mathbf{A}[n]\mathbf{A}[n]^T + \delta\mathbf{I}), \quad (3.81)$$

podemos darnos cuenta que se cumplen las siguientes relaciones de forma parecida a lo que sucedía en los algoritmos FTF descritos en el anexo B,

$$\mathbf{R}_N[n] = \begin{pmatrix} r'_{oL}[n] & \mathbf{r}'_{N-1}[n]^T \\ \mathbf{r}'_{N-1}[n] & \mathbf{R}_{N-1}[n-1] \end{pmatrix} = \begin{pmatrix} r'_{oL}[n] & \mathbf{r}'_{N-1}[n]^T \\ \mathbf{r}'_{N-1}[n] & \tilde{\mathbf{R}}_N[n] \end{pmatrix} \quad (3.82)$$

$$\mathbf{R}_N[n] = \begin{pmatrix} \mathbf{R}_{N-1}[n] & \mathbf{r}'_{N-1}[n] \\ \mathbf{r}'_{N-1}[n]^T & r'_{o(N-1)}[n] \end{pmatrix} \quad (3.83)$$

donde $\mathbf{R}_{N-1}[n]$ sería la matriz a invertir en el algoritmo de proyección afín en caso de usar un orden de proyección de $N - 1$, y que coincide con los $(N - 1) \times (N - 1)$ elementos inferiores-izquierdos de la matriz $\mathbf{R}_N[n + 1]$, por

lo que $\mathbf{R}_{N-1}[n-1] = \tilde{\mathbf{R}}_N[n]$. Los valores $\mathbf{r}'_{N-1}[n]$ y $\mathbf{r}'_{N-1}[n]$ se calculan de forma similar a los que se definen para los filtros predictores con enventanado deslizante en las ecuaciones (B.98) y (B.103) respectivamente (de igual forma a (B.110) y (B.111)), siendo:

$$\begin{aligned} \mathbf{r}'_{N-1}[n] &= \sum_{i=0}^L \mathbf{x}_{N-1}[n-i-1]x[n-i] \\ &= \mathbf{r}'_{N-1}[n-1] + \mathbf{x}_{N-1}[n-1]x[n] - \mathbf{x}_{N-1}[n-L-1]x[n-L], \end{aligned} \quad (3.84)$$

$$\begin{aligned} \mathbf{r}'_{N-1}[n] &= \sum_{i=0}^L \mathbf{x}_{N-1}[n-i]x[n-i-L] \\ &= \mathbf{r}'_{N-1}[n-1] + \mathbf{x}_{N-1}[n]x[n-L] - \mathbf{x}_{N-1}[n-L]x[n-2L] \end{aligned} \quad (3.85)$$

y de forma similar a (B.112) y (B.113):

$$r'_{oL}[n] = \mathbf{x}_L^T[n]\mathbf{x}_L[n] + \delta = \delta + \sum_{i=0}^L x^2[n-i] = r'_{oL}[n-1] + x^2[n] - x^2[n-L], \quad (3.86)$$

$$\begin{aligned} r'_{oL}[n] &= \mathbf{x}_L^T[n-L]\mathbf{x}_L[n-L] + \delta \\ &= \delta + \sum_{i=0}^L x^2[n-i-L] = r'_{oL}[n-1] + x^2[n-L] - x^2[n-2L]. \end{aligned} \quad (3.87)$$

Calculando la inversa de la matriz (3.82) haciendo uso de (B.17) según se describe en el anexo B, tendríamos que:

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} 1/\alpha'^f[n] & -\mathbf{a}'_{N-1}[n]/\alpha'^f[n] \\ -\mathbf{a}'_{N-1}[n]/\alpha'^f[n] & \tilde{\mathbf{R}}_N^{-1}[n] + \mathbf{a}'_{N-1}[n]\mathbf{a}'_{N-1}[n]/\alpha'^f[n] \end{pmatrix} \quad (3.88)$$

donde:

$$\alpha'^f[n] = r'_{oL}[n] - \mathbf{r}'_{N-1}[n]^T \mathbf{R}_{N-1}^{-1}[n-1] \mathbf{r}'_{N-1}[n] = r'_{oL}[n] - \mathbf{a}'_{N-1}[n] \mathbf{r}'_{N-1}[n]. \quad (3.89)$$

La matriz (3.88), puede escribirse de forma más compacta si definimos unos vectores extendidos de los coeficientes adaptativos que incluyen la unidad. Así:

$$e^f[n] = x[n] - \hat{x}[n] = x[n] - \mathbf{a}'_{N-1}[n]\mathbf{x}_{N-1}[n-1] = \hat{\mathbf{a}}_{N-1}^T[n]\mathbf{x}_N[n] \quad (3.90)$$

siendo:

$$\mathbf{x}_N[n-1] = (x[n], x[n-1], \dots, x[n-N+1])^T \quad (3.91)$$

y

$$\hat{\mathbf{a}}_{N-1}[n] = (1, -\mathbf{a}_{N-1}^T[n])^T \quad (3.92)$$

por lo que ahora, podemos poner la ecuación (3.88) como:

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \tilde{\mathbf{R}}_N^{-1}[n] \end{pmatrix} + \hat{\mathbf{a}}_{N-1}[n] \hat{\mathbf{a}}_{N-1}^T[n] / \alpha'^f[n] \quad (3.93)$$

donde $\mathbf{0}_N$ es un vector columna de N ceros.

De igual manera, podemos escribir

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} \mathbf{R}_{N-1}^{-1}[n] + \mathbf{b}'_{N-1}[n] \mathbf{b}'_{N-1}^T[n] / \alpha^b[n] & -\mathbf{b}'_{N-1}[n] / \alpha^b[n] \\ -\mathbf{b}_{N-1}^T[n] / \alpha^{b'}[n] & 1 / \alpha^{b'}[n] \end{pmatrix} \quad (3.94)$$

donde:

$$\alpha^b[n] = r_{oL}^b[n] - \mathbf{r}_N^b[n]^T \mathbf{R}_N^{-1}[n] \mathbf{r}_N^b[n] = r_{oL}^b[n] - \mathbf{b}_N^T[n] \mathbf{r}_N^b[n] \quad (3.95)$$

y definiendo

$$\begin{aligned} e^b[n] &= x[n-N] - \hat{x}[n-N] = x[n-N] - \mathbf{b}'_{N-1}^T[n] \mathbf{x}_{N-1}[n-N-1] \\ &= \hat{\mathbf{b}}_N^T[n] \mathbf{x}_N[n-N] \end{aligned} \quad (3.96)$$

con

$$\hat{\mathbf{b}}_N[n] = (1, -\mathbf{b}'_{N-1}^T[n])^T \quad (3.97)$$

podemos sintetizar la ecuación (3.94) de forma que:

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} \mathbf{R}_{N-1}^{-1}[n] & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{pmatrix} + \hat{\mathbf{b}}_N[n] \hat{\mathbf{b}}_N^T[n] / \alpha'^b[n]. \quad (3.98)$$

Volviendo al cálculo eficiente del vector de error normalizado ($\epsilon[n]$) podemos definir los siguientes vectores basados en él:

$$\tilde{\epsilon}[n] = \tilde{\mathbf{R}}_N^{-1}[n] \tilde{\mathbf{e}}_N[n] = \mathbf{R}_{N-1}^{-1}[n] \tilde{\mathbf{e}}_N[n] \quad (3.99)$$

$$\bar{\epsilon}[n] = \mathbf{R}_{N-1}^{-1}[n] \bar{\mathbf{e}}_N[n] \quad (3.100)$$

donde $\bar{\mathbf{e}}_N[n]$ y $\tilde{\mathbf{e}}_N[n]$ son los $N-1$ valores superiores (más recientes) e inferiores (más antiguos), respectivamente, de $\mathbf{e}_N[n]$.

Además, podemos comprobar que $\widetilde{\mathbf{R}}_N^{-1}[n]$ coincide con $\mathbf{R}_{N-1}^{-1}[n-1]$, y que $\widetilde{\mathbf{e}}_N[n] = \bar{\mathbf{e}}_N[n-1]$, con lo que podemos reescribir (3.99) de la siguiente forma:

$$\widetilde{\epsilon}[n] = \mathbf{R}_{N-1}^{-1}[n-1]\bar{\mathbf{e}}_N[n-1] = \bar{\epsilon}[n-1]. \quad (3.101)$$

Si ahora multiplicamos (3.82) por la derecha, por el vector $\mathbf{e}_N[n]$ obtendríamos que:

$$\begin{aligned} \epsilon[n] &= \begin{pmatrix} 0 \\ \widetilde{\epsilon}[n] \end{pmatrix} + \frac{\widehat{\mathbf{a}}'_{N-1}[n]\widehat{\mathbf{a}}'_{N-1}[n]^T \mathbf{e}_N[n]}{\alpha'^f[n]} \\ &= \begin{pmatrix} 0 \\ \bar{\epsilon}[n-1] \end{pmatrix} + \frac{\widehat{\mathbf{a}}'_{N-1}[n]\widehat{\mathbf{a}}'_{N-1}[n]^T \mathbf{e}_N[n]}{\alpha'^f[n]}, \end{aligned} \quad (3.102)$$

y multiplicando (3.83) por la derecha, por el vector $\mathbf{e}_N[n]$, obtenemos:

$$\epsilon[n] = \begin{pmatrix} 0 \\ \bar{\epsilon}[n] \end{pmatrix} + \frac{\widehat{\mathbf{b}}'_{N-1}[n]\widehat{\mathbf{b}}'_{N-1}[n]^T \mathbf{e}_N[n]}{\alpha'^b[n]} \quad (3.103)$$

de donde se puede despejar $\epsilon[n]$ para la siguiente iteración.

Para calcular los coeficientes de los filtros predictores ($\mathbf{a}_{N-1}[n]$ y $\mathbf{b}_{N-1}[n]$), así como los respectivos errores mínimos de predicción ($\alpha'^f[n]$ y $\alpha'^b[n]$) podemos proceder de forma similar a como se describe en el anexo B para resolver el algoritmo de mínimos cuadrados de ventana deslizante *fast* RLS o implementar una versión del *fast* RLS usando el error a posteriori.

Capítulo 4

Aplicación del algoritmo de proyección afín y sus versiones eficientes al CAR

4.1. Introducción

Como ya se ha comentado anteriormente, los algoritmos de proyección afín [49] han sido propuestos en los últimos años como controladores de los sistemas adaptativos puesto que mejoran la velocidad de convergencia de los algoritmos LMS (y variantes) manteniendo un coste computacional moderado. Estos algoritmos son pues, una alternativa eficiente a los algoritmos clásicos. Aun cuando los algoritmos de proyección afín han sido estudiados con gran interés desde que fueron propuestos para solucionar problemas de optimización adaptativa, la mejora en busca de estrategias eficientes ha estado siempre enfocada a resolver el problema de cancelación de eco [60]. El control activo de ruido [68] es un problema similar que admite una solución adaptativa por lo que los algoritmos de proyección afín también han sido aplicados como estrategia de control en dichos sistemas [63], [64]. Hay que recordar que el problema de la cancelación de eco [69] resulta análogo al de identificación de canal, mientras que para el control activo de ruido, no solo hay que identificar un canal (camino primario), sino que hay que invertir el efecto que otro canal existente (camino secundario) ejerce sobre la señal generada, puesto que ésta, a diferencia de lo que sucede en las aplicaciones de cancelación de eco, no es inyectada directamente en la señal deseada o señal a cancelar. Es por esto por lo que aparecen diferentes arquitecturas, como se ha comentado en el capítulo 2, que tratan de corregir el impacto de este camino secundario. No obstante, la preocupación por optimizar la eficiencia computacional de estos

algoritmos cuando son aplicados en sistemas de control de ruido, nos ha llevado a reflexionar sobre estrategias alternativas a las ya propuestas, que consiguen reducir el número de operaciones por iteración sin degradar las prestaciones de los algoritmos.

En este capítulo se discute cómo aplicar los algoritmos de proyección afín al control activo de ruido haciendo uso de la estructura de filtrado-x convencional [70],[71], y no de la habitualmente usada con estos algoritmos (estructura de filtrado-x modificada o de retardo compensado) [43], como se propone en la bibliografía existente [63], [64], [72]-[75]. Si bien los algoritmos propuestos podrían aplicarse a cualquier controlador multicanal de un sistema acústico (incluida la aplicación de CAR), debe hacerse notar que la estructura modificada (representada en la figura 2.14) consigue una velocidad de convergencia mayor que la obtenida por la estructura de filtrado-x convencional (figura 2.13), ya que compensa el retardo que la estructura de filtrado-x convencional introduce entre la actualización de los pesos de los algoritmos adaptativos y el efecto que se provoca en la señal a cancelar, ofreciendo la posibilidad de adaptar los algoritmos de proyección afín al CAR de una forma casi inmediata, puesto que con dicha estructura se tienen todas las señales que dichos algoritmos necesitan para su óptimo funcionamiento (incluida la señal $d[n]$, que en los algoritmos de CAR no suele estar presente). Sin embargo, la estructura de filtrado-x convencional proporciona un importante ahorro computacional [76], sobre todo para el caso multicanal, puesto que realizamos la mitad de filtrados por canal que con la estructura modificada sin introducir un gran perjuicio en las propiedades de convergencia de dichos algoritmos si lo comparamos con algunas estrategias que pretenden minimizar el coste computacional de los algoritmos de proyección afín usando la estructura de filtrado-x modificada [77], como el cálculo eficiente del vector de error [59]. Además, otras técnicas que se aplican a los algoritmos de proyección afín con la estructura de filtrado-x modificada para ahorrar coste computacional, pueden también adaptarse a la estructura de filtrado-x convencional. Es importante tener en cuenta que la estructura de filtrado-x convencional tiene principalmente dos inconvenientes: que posee una velocidad de convergencia menor que la de filtrado-x modificada y que necesitamos introducir ciertas aproximaciones para aplicar el algoritmo de proyección afín. Es por ello que hasta el momento sólo se ha aplicado la estructura de filtrado-x modificada en los algoritmos de proyección afín aplicados al control activo de ruido, aunque como veremos a continuación, puede lograrse un alto grado de compromiso entre las prestaciones del algoritmo y su coste computacional cuando usamos la estructura de filtrado-x convencional. Así que como novedad, se introducirá el algoritmo de proyección afín aplicado al control activo de ruido multicanal usando la estructura de filtrado-x convencional, prestando especial atención al coste computacional que necesita y

comparándolo con la carga computacional de otras alternativas eficientes ya propuestas en la bibliografía.

4.2. Algoritmos de proyección afín aplicados al CAR

Adaptar el algoritmo de proyección afín y sus versiones eficientes descritas en el capítulo 3 al problema del control activo de ruido, requiere algunos cambios en el mismo, como se puede suponer tras lo comentado en el punto 2.7 donde se describen las estrategias válidas para aplicar cualquier algoritmo adaptativo a la aplicación de CAR. Hay que tener en cuenta que ahora la señal de error no es la diferencia entre una señal generada y una deseada sino que es la suma acústica entre la señal deseada y la señal generada tras atravesar un camino acústico o planta acústica (por lo cual, la actualización de los pesos incorporará un cambio de signo y, por lo general, involucrará a unas señales que no son las de referencia, sino estas señales filtradas por una estimación del filtro acústico que ha de atravesar la señal generada, dependiendo de la estructura de control usada). Otra particularidad de las implementaciones reales es que la señal de error en el instante n -ésimo depende de los datos generados en el instante anterior (si suponemos un sistema DSP con muestreo periódico, en un instante cualquiera obtendrá las muestras de la señal de referencia en ese instante $x[n]$ y la señal de error $e[n]$ que es la suma acústica de la señal deseada y la generada por nuestro sistema en el instante anterior). Por lo tanto, si intentamos aplicar la ecuación de actualización de los coeficientes tal cual se define, por ejemplo, en (3.33), tendremos un desajuste entre las señales de referencia y la señal de error (provienen de la señal de referencia en diferentes instantes de muestreo). Para que los algoritmos funcionen óptimamente, debe existir la mayor correlación posible entre ambas señales, por lo que se ha de retrasar una muestra las señales que involucran a la señal de referencia. Es por ello que en las ecuaciones de actualización de los coeficientes de los algoritmos descritos aplicados al CAR, aparecerán los vectores y matrices relacionados con los datos de la señal de referencia retrasados una unidad temporal.

Por otra parte, el algoritmo de proyección afín se aplica al CAR haciendo uso de la estructura de filtrado-x modificada [63],[64]. La razón por la que se usa esta estructura es doble: por un lado, dicha estructura proporciona una estimación de la señal $d[n]$ necesaria para el cálculo del vector de error $\mathbf{e}_N[n]$ según se puede deducir de (3.30), y por otro lado, la estructura modificada es la que mejores propiedades muestra en cuanto a velocidad de convergencia para el problema del control activo de ruido [78]. Sin embargo, según se comenta en [19], la forma más sencilla de compensar los efectos que las plantas acústicas tienen sobre la convergencia del algoritmo es mediante la estructu-

ra de filtrado-x (representada en la figura 2.13). No obstante, el uso de esta estructura introduce ciertas restricciones en las propiedades de convergencia debidas al retardo que se introduce en la señal de referencia al ser filtrada por la estimación de la planta acústica. La estructura modificada, aun cuando es más compleja que la de filtrado-x, compensa dicho retardo y ofrece mejores propiedades de convergencia cuando la estimación del camino acústico es suficientemente buena, pero su coste computacional es mayor. Lo que se tratará de analizar es si la aplicación de las estrategias eficientes al algoritmo de proyección afín con la estructura de filtrado-x modificada proporcionan resultados que mejoran las prestaciones de los obtenidos mediante el uso de la estructura de filtrado-x convencional.

4.2.1. Algoritmo de proyección afín multicanal usando la estructura de filtrado-x modificada

Para definir el algoritmo de proyección afín multicanal con estructura de filtrado-x modificada nos ayudaremos de la siguiente nomenclatura (que será usada en lo que resta de capítulo):

- I : número de señales de referencia.
- J : número de señales secundarias.
- K : número de sensores de error.
- L : longitud de los filtros adaptativo.
- N : orden de proyección afín.
- M : longitud de los filtros $\mathbf{h}_{j,k}$ que modelan la planta acústica entre el sensor de error k -ésimo y el actuador j -ésimo.
- $x_i[n]$: valor en el instante n de la señal de referencia i -ésima.
- $y_j[n]$: valor en el instante n de la señal generada en el actuador j -ésimo.
- $e_k[n]$: valor en el instante n de la señal en el sensor de error k -ésimo.
- $w_{i,j,l}[n]$: valor en el instante n del l -ésimo coeficiente del filtro adaptativo que alimenta al actuador j a partir de la señal de referencia i -ésima.
- $\mathbf{w}_{i,j}[n] = (w_{i,j,1}[n], w_{i,j,2}[n], \dots, w_{i,j,L}[n])^T$.
- $h_{j,k,m}$: valor del m -ésimo coeficiente en el filtro FIR que modela la planta acústica entre el actuador j -ésimo y el sensor de error k -ésimo.

- $\mathbf{h}_{j,k} = (h_{j,k,1}, h_{j,k,2}, \dots, h_{j,k,M})^T$.
- $v_{i,j,k}[n]$: valor en el instante n de la señal de referencia $x_i[n]$ filtrada por el modelo de la planta $\mathbf{h}_{j,k}$ ($x_f[n]$ en las figuras 2.13) y 2.14.
- $\mathbf{v}_{i,j,k}[n] = (v_{i,j,k}[n], v_{i,j,k}[n-1], \dots, v_{i,j,k}[n-L+1])^T$.
- $\mathbf{V}_{i,j,k}[n] = (\mathbf{v}_{i,j,k}[n] \dots \mathbf{v}_{i,j,k}[n-N+1])^T$.
- $\mathbf{x}_i[n] = (x_i[n], x_i[n-1], \dots, x_i[n-L+1])^T$.
- $\mathbf{x}_i'[n] = (x_i[n], x_i[n-1], \dots, x_i[n-M+1])^T$.
- $\mathbf{y}_j[n] = (y_j[n], y_j[n-1], \dots, y_j[n-M+1])^T$.
- $\hat{\mathbf{d}}_k[n] = (\hat{d}_k[n], \hat{d}_k[n-1], \dots, \hat{d}_k[n-N+1])^T$.

Los pasos para ejecutar el algoritmo (que denominaremos de ahora en adelante MFXAP) y el número de multiplicaciones que implica para un sistema multicanal con I señales de referencia, J fuentes secundarias y K sensores de error (como medida del coste computacional), se reflejan a continuación:

- Filtramos la señal de referencia por los caminos secundarios
 $v_{i,j,k}[n] = \mathbf{h}_{j,k}^T \mathbf{x}_i'[n]$
 Multiplicaciones: $MIJK$.
- Calculamos la estimación de la señal deseada
 $\hat{d}_k[n] = e_k[n] - \sum_{j=1}^J \mathbf{h}_{j,k}^T \mathbf{y}_j[n]$
 Multiplicaciones: $MIJK$.
- Calculamos la señal de error
 $\hat{\mathbf{e}}_k[n] = \hat{\mathbf{d}}_k[n] + \sum_{i=1}^I \sum_{j=1}^J \mathbf{w}_{i,j}^T[n] \mathbf{V}_{i,j,k}[n]$
 Multiplicaciones: $LNIJK$.
- Calculamos el vector de error normalizado
 $\epsilon_{i,j,k}[n] = (\mathbf{V}_{i,j,k}^T[n-1] \mathbf{V}_{i,j,k}[n-1] + \delta I)^{-1} \hat{\mathbf{e}}_k[n]$
 Multiplicaciones: $KIJ[LN^2 + O(\frac{N^3}{2} + N^2)]$.
- Actualizamos los coeficientes
 $\mathbf{w}_{i,j}[n] = \mathbf{w}_{i,j}[n-1] - \mu \sum_{k=1}^K \mathbf{V}_{i,j,k}[n-1] \epsilon_{i,j,k}[n]$
 Multiplicaciones: $K(NL + L)IJ$.

- Realizamos el filtrado adaptativo

$$y_j[n] = \sum_{i=1}^I \mathbf{w}_{i,j}^T[n] \mathbf{x}_i[n]$$

Multiplicaciones: LIJ .

Este número de operaciones puede reducirse si usamos alguna o todas las estrategias eficientes vistas en el capítulo 3 (adaptándolas para el CAR).

Usando la actualización eficiente del vector de error, tendríamos que el vector que contiene la estimación de la señal de error $\hat{\mathbf{e}}_k[n]$ podría calcularse (bajo la hipótesis de μ pequeña), según (3.53) siguiendo los pasos:

- $$\hat{\mathbf{e}}_k[n] = \hat{d}_k[n] + \sum_{i=1}^I \sum_{j=1}^J \mathbf{w}_{i,j}^T[n] \mathbf{v}_{i,j,k}[n]$$

Multiplicaciones: LJK

- $$\mathbf{e}'_k[n] = \begin{pmatrix} \hat{\mathbf{e}}_k[n] \\ (1 - \mu) \bar{\mathbf{e}}'_k[n - 1] \end{pmatrix}$$

Multiplicaciones: $(N - 1)K$.

siendo $\bar{\mathbf{e}}'_k[n - 1]$ los $N - 1$ elementos superiores de $\mathbf{e}'_k[n - 1]$. De esta forma introduciríamos un ahorro desde $LJJKN$ hasta $LJJK + (N - 1)K$ multiplicaciones, si bien es cierto que estamos introduciendo aproximaciones en los datos que degradan la velocidad de convergencia. A esta variante la denominaremos *fast* MFXAP-FE.

Si empleamos, en cambio, la estrategia de actualización eficiente de los coeficientes (uso de los coeficientes auxiliares), teniendo en cuenta que:

- $\bar{\mathbf{r}}_{i,j,k}[n]$ es el vector auxiliar de tamaño $(N - 1) \times 1$ que nos sirve para el cálculo iterativo del producto $\mathbf{v}_{i,j,k}^T[n] \bar{\mathbf{V}}_{i,j,k}^T[n]$.
- $\bar{\mathbf{V}}_{i,j,k}^T[n]$ es la matriz de tamaño $(N - 1) \times L$ que contiene las $N - 1$ filas superiores de la matriz $\mathbf{V}_{i,j,k}[n]$.
- $\bar{\mathbf{v}}_{i,j,k}[n]$ es el vector de tamaño $(N - 1) \times 1$ con los $N - 1$ elementos más recientes de la señal $v_{i,j,k}[n]$.
- $e_{i,j,k(N-1)}[n]$ es el último elemento del vector $\mathbf{e}'_{i,j,k}[n]$.
- $\bar{\mathbf{e}}'_k[n - 1]$ contiene los primeros $N - 1$ elementos de $\mathbf{e}'_{i,j,k}[n - 1]$.

Las operaciones a realizar serían:

- Calculamos el vector de error normalizado

$$\epsilon_{i,j,k}[n] = (\mathbf{V}_{i,j,k}^T[n-1]\mathbf{V}_{i,j,k}[n-1] + \delta I)^{-1}\widehat{\mathbf{e}}_k[n]$$
 Multiplicaciones: $KIJ(LN^2 + O(\frac{N^3}{2}) + N^2)$.
- Calculamos el vector de error auxiliar

$$\mathbf{e}'_{i,j,k}[n] = \epsilon_{i,j,k}[n] + \begin{pmatrix} 0 \\ \bar{\mathbf{e}}'_{i,j,k}[n-1] \end{pmatrix}.$$
- Actualizamos los coeficientes auxiliares

$$\widehat{\mathbf{w}}_{i,j}[n] = \widehat{\mathbf{w}}_{i,j}[n-1] + \mu \sum_{k=1}^K \mathbf{v}_{i,j,k}[n-N]e_{i,j,k(N-1)}[n]$$
 Multiplicaciones: $(LK + 1)IJ$.
- Filtramos por los caminos estimados

$$v_{i,j,k}[n] = \mathbf{h}_{j,k}^T \mathbf{x}'_i[n]$$
 Multiplicaciones: $MIJK$.
- Calculamos el vector auxiliar $\bar{\mathbf{r}}_{i,j,k}[n]$

$$\bar{\mathbf{r}}_{i,j,k}[n] = \bar{\mathbf{r}}_{i,j,k}[n-1] + v_{i,j,k}[n]\bar{\mathbf{v}}_{i,j,k}^T[n] - v[n-L]\bar{\mathbf{v}}_{i,j,k}^T[n-L]$$
 Multiplicaciones: $2(N-1)IJK$.
- Realizamos el filtrado adaptativo

$$y_j[n] = \sum_{i=1}^I \left(\widehat{\mathbf{w}}_{i,j}^T[n] \mathbf{x}_i[n] - \sum_{k=1}^K \mu \bar{\mathbf{r}}_{i,j,k}^T[n] \bar{\mathbf{e}}'_{i,j,k}[n-1] \right)$$
 Multiplicaciones: $IJ(L + NK)$.

A este algoritmo que usa la estrategia eficiente de actualización de los coeficientes lo denominaremos *fast* MFXAP-CA.

El número total de multiplicaciones por iteración que necesitaría una variante del algoritmo de proyección afín que incorporara tanto la actualización eficiente de los coeficientes como la actualización eficiente del vector de error sería $KIJ[M + 3N - 2 + 2L + N^2(L + 1) + O(\frac{N^3}{2})] + IJ(L + 1) + JKM + (N - 1)K$, introduciendo un ahorro debido al uso de los coeficientes auxiliares (actualización eficiente de los coeficientes) de $IJK(LN + 2 - 3N) - IJ$ multiplicaciones.

Por último, como se especificó en el capítulo 3, el cálculo del vector de error normalizado también puede hacerse de forma eficiente usando solo $K(5N - 2)$ multiplicaciones, más las necesarias para calcular los coeficientes de los filtros predictores y los errores de predicción (que serían $K(10N + 22)$ si usamos un algoritmo *fast* RLS enventanado de orden N). La desventaja de esta estrategia es que el cálculo de dichos coeficientes puede diverger por la propia inestabilidad del algoritmo RLS, así que podemos usar estrategias alternativas que aunque no son tan eficientes sí que garantizan la estabilidad, como

la propuesta también en el capítulo 3 (publicada en [61]) donde se describe cómo reducir el coste de la inversión matricial desde $K(LN^2 + O(\frac{N^3}{2}))$ hasta $6K(N^2 + 2N)$. Así, el coste computacional del algoritmo de proyección afín con estructura modificada usando estrategias eficientes y sin ellas, queda resumido en la tabla 4.1 (ejemplo típico para un sistema 1:2:2 con $M = 250$, $L = 50$ y $N = 5$), donde el algoritmos *fast* MFXAP- ϵ es el que usa la estrategia eficiente comentada para el cálculo de la inversión matricial, y el *fast* MFXAP usa conjuntamente todas las estrategias eficientes (la actualización eficiente de los coeficientes, del vector de error y del vector de error normalizado).

Algoritmo	Multiplicaciones por iteración	Ejemplo
MFXAP	$KIJ[M + 2LN + L + LN^2 + N^2 + O(\frac{N^3}{2})] + LIJ + JKM$	10150
<i>fast</i> MFXAP-FE	$KIJ[M + LN + 2L + LN^2 + N^2] + O(\frac{N^3}{2}) + K(N - 1) + LIJ + JKM$	9358
<i>fast</i> MFXAP-CA	$KIJ[M + 3N - 2 + LN + LN^2 + N^2 + O(\frac{N^3}{2})] + IJ(L + LK + 1) + JKM$	9204
<i>fast</i> MFXAP- ϵ	$KIJ(M + 2LN + L + 7N^2 + 12N) + LIJ + JKM$	5240
<i>fast</i> MFXAP	$KIJ(M - 2 + 7N^2 + 15N) + IJ(L + LK + 1) + K(N - 1) + JKM$	3302

Tabla 4.1: Número de multiplicaciones por iteración para diferentes algoritmos de proyección afín con estructura modificada que emplean estrategias eficientes. Ejemplo para $L = 50$, $M = 250$, $N = 5$, $I = 1$, $J = 2$ y $K = 2$.

4.2.2. Algoritmo de proyección afín multicanal usando la estructura de filtrado-x convencional

Como ya se ha comentado anteriormente, la estructura de filtrado-x puede ser usada conjuntamente con el algoritmo de proyección afín considerando que habría que aproximar, de algún modo, el vector de error $\mathbf{e}[n]$. Con esta estructura, decrecería el número de operaciones por iteración (puesto que son necesarios menos filtrados), si bien, la velocidad de convergencia es algo más lenta. Como se puede apreciar en la figura 2.13, la señal de error calculada según (3.30), pasaría ahora a calcularse según (4.1):

$$\mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{V}[n]\mathbf{w}_L[n]. \quad (4.1)$$

El motivo por el cual debemos aproximar o estimar dicho vector, es que con esta estructura no disponemos de la señal $d[n]$. La forma más sencilla de aproximar dicho vector sería mediante los datos de la señal de error (tanto en el instante actual como en los $N - 1$ anteriores) ($\mathbf{e}_N[n] = (e[n], e[n - 1], \dots, e[n - N + 1])^T$). Esto es justificable, si tenemos en cuenta que la variación de los pesos es suficientemente lenta como para considerarse constante en una ventana temporal de duración N . Para un sistema monocanal se podría describir esta hipótesis según:

$$\begin{aligned} \mathbf{e}_N[n] &= \mathbf{d}_N[n] - \mathbf{V}[n]\mathbf{w}_L[n] = \\ &\begin{pmatrix} d[n] - \mathbf{v}_L^T[n]\mathbf{w}_L[n] \\ d[n - 1] - \mathbf{v}_L^T[n - 1]\mathbf{w}_L[n] \\ \vdots \\ d[n - N + 1] - \mathbf{v}_L^T[n - N + 1]\mathbf{w}_L[n] \end{pmatrix} \approx \\ &\approx \begin{pmatrix} d[n] - \mathbf{v}_L^T[n]\mathbf{w}_L[n] \\ d[n - 1] - \mathbf{v}_L^T[n - 1]\mathbf{w}_L[n - 1] \\ \vdots \\ d[n - N + 1] - \mathbf{v}_L^T[n - N + 1]\mathbf{w}_L[n - N + 1] \end{pmatrix}, \end{aligned} \quad (4.2)$$

siendo $\mathbf{v}_L[n]$ un vector columna que contiene las últimas L muestras de la señal de referencia filtrada por la estimación del camino secundario, (resultado de filtrar la señal de referencia $x'(n)$ por \mathbf{h}) y $\mathbf{V}[n]$ una matriz de tamaño $N \times L$ formada por los últimos N vectores $\mathbf{v}_L[n]$.

$$\begin{aligned} \mathbf{v}_L[n] &= (v[n], v[n - 1], \dots, v[n - L + 1])^T. \\ v[n] &= \mathbf{h}^T \mathbf{x}'[n]. \end{aligned}$$

Las suposiciones realizadas son bastante realistas considerando que el sistema está en el régimen permanente (mínima variación de los pesos) o que la constante de convergencia μ es pequeña.

Los pasos que definirían este algoritmo (llamado a partir de ahora FXAP), serían:

- Calculamos el vector de error normalizado:
 $\epsilon_{i,j,k}[n] = (\mathbf{V}_{i,j,k}[n - 1]\mathbf{V}_{i,j,k}^T[n - 1] + \delta I)^{-1} \mathbf{e}_k[n]$
 Multiplicaciones: $KIJ(LN^2 + N^2 + O(N^3/2))$.
- Actualizamos los coeficientes:
 $\mathbf{w}_{i,j}[n] = \mathbf{w}_{i,j}[n - 1] - \mu \sum_{k=1}^K \mathbf{V}_{i,j,k}^T[n - 1] \epsilon_{i,j,k}[n]$
 Multiplicaciones: $K(LN + L)IJ$.

- Filtramos por los caminos estimados:

$$v_{i,j,k}[n] = \mathbf{h}_{j,k}^T \mathbf{x}'_i[n]$$

Multiplicaciones: $MIJK$.

- Realizamos el filtrado adaptativo:

$$y_j[n] = \sum_{i=1}^I \mathbf{w}_{i,j}^T[n] \mathbf{x}'_i[n]$$

Multiplicaciones: LIJ .

Este algoritmo necesitaría un total de $KIJ[LN + L + M + LN^2 + N^2 + O(\frac{N^3}{2})] + LIJ$ multiplicaciones por iteración, ahorrando con respecto a la estructura modificada $MIJK + LNIJK$ multiplicaciones. Además, hay que tener en cuenta que sobre esta estructura también pueden aplicarse las estrategias eficientes descritas en el capítulo 3, lo que resulta casi idéntico que las modificaciones realizadas al algoritmo MFXAP para introducirle dichas estrategias eficientes. La salvedad es, que en este caso no es necesario aplicar la estrategia del cálculo eficiente del vector de error puesto que como se dedujo en (4.2), con la estructura de filtrado-x, el vector de error se consigue con muestras de los datos de la señal de error. Pero las otras dos estrategias si que pueden ser abordadas.

Por ejemplo, el algoritmo FXAP con el cálculo eficiente de la actualización de los coeficientes (al que llamaremos *fast* FXAP-CA) quedaría:

- Calculamos el vector de error normalizado:

$$\epsilon_{i,j,k}[n] = (\mathbf{V}_{i,j,k}[n-1] \mathbf{V}_{i,j,k}^T[n-1] + \delta I)^{-1} \mathbf{e}_k[n]$$

Multiplicaciones: $KIJ(LN^2 + N^2 + O(\frac{N^2}{2}))$.

- Calculamos el vector de error auxiliar:

$$\mathbf{e}'_{i,j,k}[n] = \epsilon_{i,j,k}[n] + \begin{pmatrix} 0 \\ \hat{\mathbf{e}}_k[n-1] \end{pmatrix}$$

- Actualizamos los coeficientes auxiliares:

$$\hat{\mathbf{w}}_{i,j}[n] = \hat{\mathbf{w}}_{i,j}[n-1] + \mu \sum_{k=1}^K \mathbf{v}_{i,j,k}[n-N] e_{i,j,k(N-1)}[n]$$

Multiplicaciones: $(LK + 1)IJ$.

- Filtramos por los caminos estimados:

$$v_{i,j,k}[n] = \mathbf{h}_{j,k}^T \mathbf{x}'_i[n]$$

Multiplicaciones: $MIJK$.

- Calculamos el vector auxiliar $\bar{\mathbf{r}}_{i,j,k}[n]$

$$\bar{\mathbf{r}}_{i,j,k}[n] = \bar{\mathbf{r}}_{i,j,k}[n-1] + v_{i,j,k}[n] \bar{\mathbf{v}}_{i,j,k}^T[n] - v[n-L] \bar{\mathbf{v}}_{i,j,k}^T[n-L]$$

Multiplicaciones: $2(N-1)IJK$.

- Realizamos el filtrado adaptativo:

$$y_j[n] = \sum_{i=1}^I (\widehat{\mathbf{w}}_{i,j}^T[n] \mathbf{x}_i[n] - \sum_{k=1}^K \mu \bar{\mathbf{r}}_{i,j,k}^T[n] \mathbf{e}'_{i,j,k}[n-1])$$

Multiplicaciones: $IJ(L + NK)$.

Algoritmo	Multiplicaciones por iteración	Ejemplo
<i>fast</i> MFXAP	$KIJ(M - 2 + 7N^2 + 15N) +$ $+IJ(L + LK + 1) + K(N - 1) + JKM$	3202
FXAP	$KIJ(LN + L + M + LN^2 + N^2 +$ $+O(\frac{N^3}{2})) + LIJ$	8150
<i>fast</i> FXAP-CA	$KIJ(M + 3N - 2) + LN^2 + N^2 +$ $+O(\frac{N^3}{2}) + IJ(L + LK + 1)$	7204
<i>fast</i> FXAP- ϵ	$KIJ(M + L + LN + 12N + 7N^2) +$ $+IJL$	3240
<i>fast</i> FXAP	$KIJ(M + 15N - 2 + 7N^2) +$ $+IJ(L + LK + 1)$	2294

Tabla 4.2: Número de multiplicaciones por iteración para diferentes algoritmos de proyección afín con estructura de filtrado-x convencional que emplean estrategias eficientes. Ejemplo para $L = 50$, $M = 250$, $N = 5$, $I = 1$, $J = 2$ y $K = 2$.

Hay que destacar que este algoritmo resulta idéntico al equivalente para la estructura modificada con la salvedad de que ahora el vector de error usado es $\mathbf{e}_k[n]$ y no $\widehat{\mathbf{e}}_k[n]$ como en la modificada.

Por supuesto, las inversiones matriciales que aparecen en cálculo del vector de error normalizado también puede obtenerse igual que se propuso para el caso de la estructura modificada, reduciendo su coste desde $K(LN^2 + O(\frac{N^3}{2}))$ hasta $6K(N^2 + 2N)$.

En la tabla 4.2 queda reflejado el coste computacional de este algoritmo y sus variantes comparadas con el *fast*-MFXAP (ejemplo típico para un sistema 1:2:2 con $M = 250$, $L = 50$ y $N = 5$). El algoritmo *fast* MFXAP- ϵ usa el cálculo eficiente de la inversión matricial y el *fast* FXAP combina tanto esta estrategia como la de actualización eficiente de los coeficientes.

4.3. Estudio de las prestaciones de los algoritmos propuestos

Aunque como se ha visto anteriormente, el algoritmo de proyección afín con estructura de filtrado-x es una alternativa válida desde el punto de vista computacional, faltaría comprobar la eficiencia de este algoritmo (velocidad de convergencia y atenuación final) comparado con el algoritmo de proyección afín que usa la estructura modificada y basado en las otras estrategias eficientes comentadas. Para ello se han realizado diferentes simulaciones de un sistema de control activo de ruido donde se disponía de una fuente de ruido, dos sensores de error y dos fuentes canceladoras, según la disposición que se aprecia en la figura 4.1, donde los sensores de error estarían colocados en las cercanías de los oídos del hipotético oyente y las fuentes secundarias sería las dos fuentes laterales que se aprecian en dicha figura, siendo la fuente central la que fue usada para inyectar el ruido.

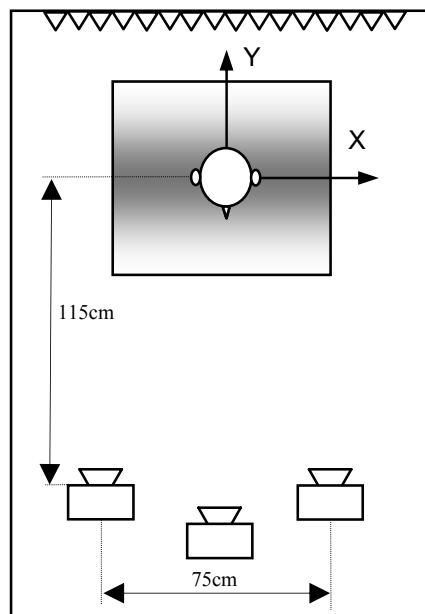
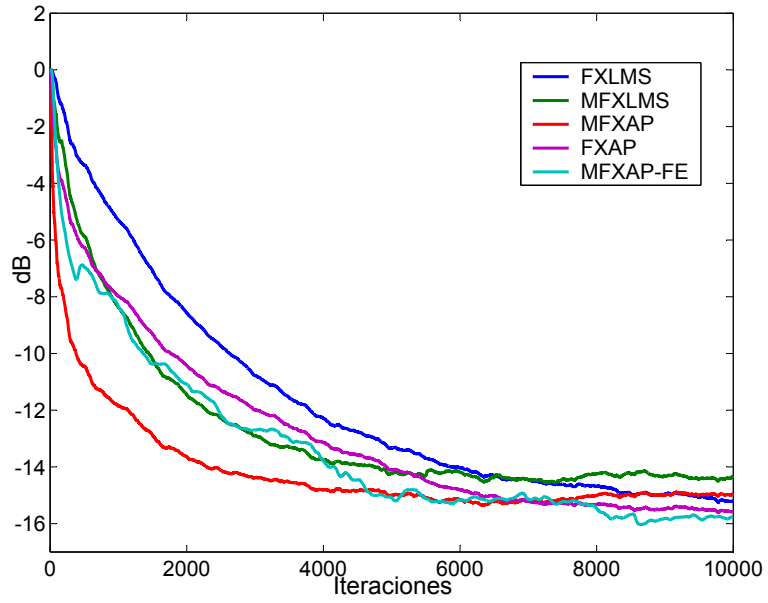


Figura 4.1: Prototipo experimental para un sistema CAR multicanal

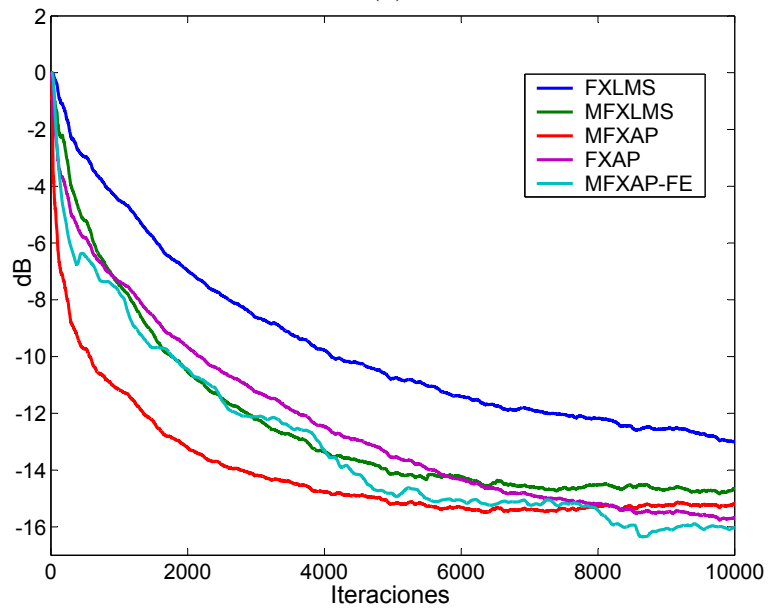
Las simulaciones han sido realizadas con el simulador acústico que se describe en el anexo D. Los caminos acústicos primarios y secundarios fueron medidos previamente (a una frecuencia de muestreo de 500 Hz) en un sistema real como el representado en la figura 4.1 e introducidos en las simulaciones,

siendo modelados con filtros FIR de 250 coeficientes. El tipo de ruido usado en las simulaciones fue ruido aleatorio de media cero. Los algoritmos estudiados en las simulaciones han sido aquellos que poseen alguna diferencia entre sí en sus prestaciones. De entre los que usan la estructura modificada, solo se presentan los resultados del *fast* MFXAP-FE y MFXAP (ya que las variantes donde se realiza el cálculo eficiente de los coeficientes auxiliares (*fast* MFXAP-CA) o el cálculo eficiente del vector normalizado de error (*fast* MFXAP- ϵ), ofrecen el mismo rendimiento en cuanto a velocidad de convergencia que el algoritmo MFXAP (sin aplicar estas estrategias), y el *fast* MFXAP obtendría resultados similares al *fast* MFXAP-FE, puesto que es esta estrategia la única que degrada los resultados en cuanto a la convergencia). Respecto a los basados en la estructura de filtrado-x convencional, solo se ha considerado el algoritmo FXAP, puesto que el resto (tanto el *fast* FXAP- ϵ como el *fast* FXAP-CA) ofrecen prestaciones idénticas al mismo (si hubiésemos definido una variante que usara el cálculo eficiente del vector de error, proporcionaría resultados algo peores (véase [77]), y tampoco mejoraría la eficiencia computacional, por lo que no ha sido considerado en estas simulaciones). También se han realizado simulaciones con los algoritmos LMS tanto con estructura de filtrado-x (que se ha llamado FXLMS) como con estructura modificada (al que se ha denominado MFXLMS), que de alguna forma sirven de referencia para evaluar las prestaciones del resto.

Las curvas obtenidas muestran la atenuación lograda por cada algoritmo en función de la iteración en cada uno de los sensores de error, y son el resultado de promediar 10 simulaciones diferentes. Los parámetros usados para configurar los algoritmos han sido los siguientes: para todos ellos se ha usado 50 coeficientes en los filtros adaptativos, y en los algoritmos de proyección afín se han usado órdenes de proyección $N = 5$ y 10 y un factor de regularización igual a 0.001 . La constante de convergencia ha sido ajustada en cada caso hasta obtener la máxima velocidad de convergencia sin producir inestabilidades en los algoritmos.

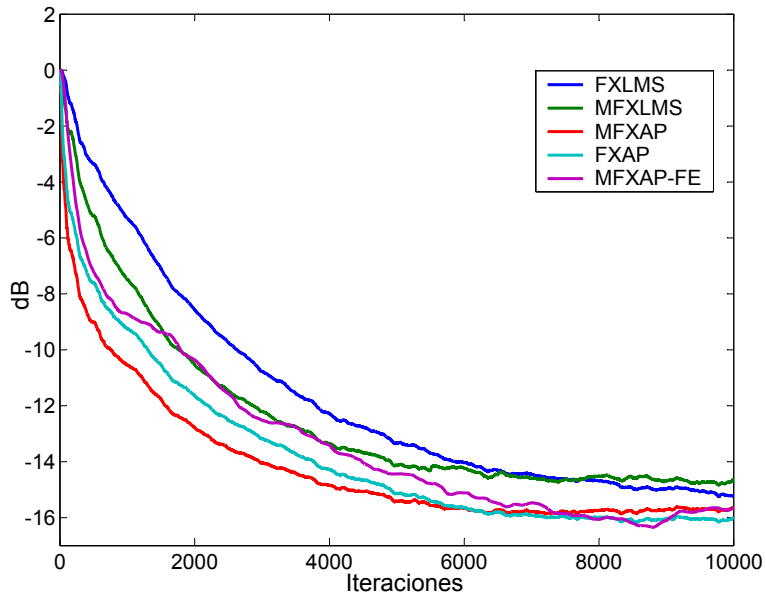


(a)

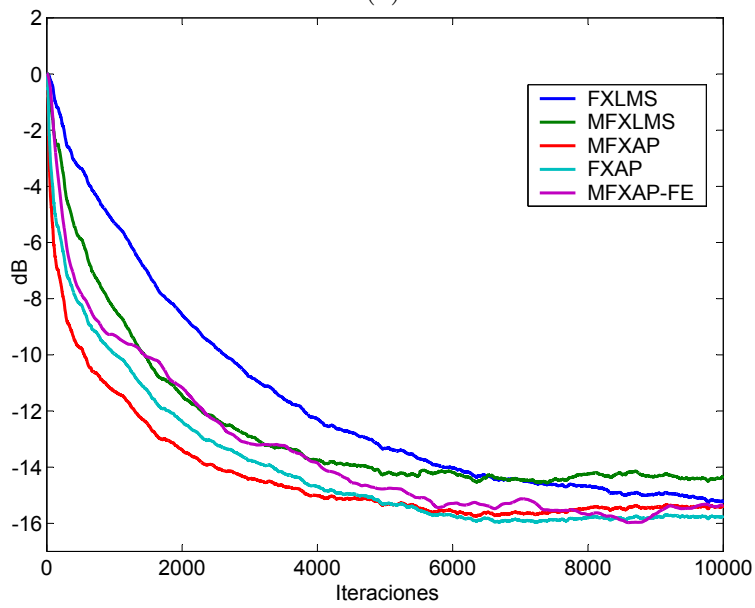


(b)

Figura 4.2: Curvas de convergencia para el algoritmo de proyección afín de orden $N = 5$, con ruido aleatorio. Sensor de error 1 (a) y 2 (b).



(a)



(b)

Figura 4.3: Curvas de convergencia para el algoritmo de proyección afín de orden $N = 10$, con ruido aleatorio. Sensor de error 1 (a) y 2 (b).

En las figuras 4.2 y 4.3 se aprecian dichas curvas en cada uno de los sensores de error, pudiéndose comprobar como la estructura modificada siempre ofrece los mejores resultados (tanto con el algoritmo de proyección afín como con el LMS). También se puede observar que cualquiera de las estrategias usada con los algoritmos de proyección afín supera las prestaciones del algoritmo de filtrado-x LMS (FXLMS). En cuanto al algoritmo bajo estudio (FXAP), podemos decir que sus prestaciones son al menos tan buenas como el algoritmo de proyección afín que usa las estrategias *fast* incluyendo el cálculo eficiente del vector de error y la estructura modificada (*fast* MFXAP, que es equivalente en cuanto a prestaciones de convergencia al *fast* FXAP-CA). Esto es debido a que en ambos se usa un vector de error que es una aproximación del real, por lo que ninguno de ellos llega al rendimiento que se observa en el MFXAP.

En el anexo A, se desarrollan modelos matemáticos que describen el comportamiento en regimen transitorio de los algoritmos de proyección afín y en el capítulo 6 se presentan más resultados de simulación, así como resultados de medidas reales en el capítulo 7.

Capítulo 5

Estudio del error cuadrático medio de los algoritmos de proyección afín.

5.1. Introducción

En este capítulo se pretende obtener un modelo matemático que nos ofrezca información sobre el comportamiento en estado estacionario de los algoritmos desarrollados permitiendo predecir su comportamiento y facilitar el ajuste de los parámetros que definen su comportamiento en un sistema práctico. Aunque los resultados alcanzados para el modelo del MSE del algoritmo de proyección afín, ya han sido propuestos en [79], para mayor claridad de la metodología usada en el análisis del MSE de dicho algoritmo aplicado al CAR, comenzaremos detallando este mismo proceso para posteriormente extrapolarlo a los algoritmos aplicados al CAR descritos en el capítulo anterior.

Para el estudio de las propiedades de convergencia (y del rendimiento) de los algoritmos de proyección afín, partiremos (al igual que en [79]) de la expresión de actualización de los coeficientes dada en (3.33), que contemplaba el caso de regularización.

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mu \mathbf{A}^T[n] (\mathbf{A}[n] \mathbf{A}^T[n] + \delta I)^{-1} \mathbf{e}_N[n] \quad (5.1)$$

donde

- $\mathbf{w}_L[n]$ es un vector columna que contiene los L coeficientes del filtro adaptativo en el instante n -ésimo.

- $\mathbf{A}[n]$ es una matriz de tamaño $N \times L$ que contiene los últimos N vectores fila (desde $\mathbf{x}_L^T[n]$ hasta $\mathbf{x}_L^T[n - N + 1]$) de L muestras con los correspondientes valores de la señal de referencia, $x[n]$.
- $\mathbf{e}_N[n]$ o vector de error, definido como $\mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n - 1]$, siendo $\mathbf{d}_N[n]$ un vector que contiene las últimas N muestras de la señal deseada $d[n]$.
- δ , valor de la constante de regularización.
- μ , parámetro de convergencia.

Supondremos (facilitando el análisis) que nuestro algoritmo converge a una determinada solución \mathbf{w}_L^o que en un caso ideal cancelaría en su totalidad la señal de error ($\mathbf{d}_N[n] = \mathbf{A}[n]\mathbf{w}_L^o$), pero que, considerando hipótesis más realistas, dejaría un residuo de la señal sin cancelar (por imprecisiones en las medidas, desviaciones en la convergencia, etc), siendo entonces la ecuación que relaciona esos coeficientes óptimos con la señal deseada:

$$\mathbf{d}_N[n] = \mathbf{A}[n]\mathbf{w}_L^o + \mathbf{r}_N[n], \quad (5.2)$$

considerando que $\mathbf{r}_N[n]$ contendría N muestras de un proceso aleatorio $r[n]$ con distribución uniforme, independiente, varianza σ_r^2 y media cero, que representaría el error cometido en la estimación. Esta variable también se considera independiente de $x[n]$. Para simplificar los cálculos, supondremos también que $x[n]$ y $d[n]$ tienen media nula.

5.2. Estudio del error cuadrático medio del algoritmo de proyección afín

La manera más común de evaluar las prestaciones en estado estacionario de los algoritmos es determinar el error cuadrático medio (*Mean Square Error* o MSE) cuando el algoritmo ha convergido. Matemáticamente, esto es:

$$\text{MSE} = \lim_{n \rightarrow \infty} E\{|e[n]|^2\}, \quad (5.3)$$

donde $e[n] = d[n] - \mathbf{x}_L[n]^T \mathbf{w}_L[n - 1]$ es la estimación del error en el instante n -ésimo. Además, si definimos un nuevo vector de error de los pesos del filtro adaptativo como

$$\tilde{\mathbf{w}}_L[n] = \mathbf{w}_L^o - \mathbf{w}_L[n] \quad (5.4)$$

podemos redefinir la ecuación (5.1) según

$$\tilde{\mathbf{w}}_L[n] = \tilde{\mathbf{w}}_L[n - 1] - \mu \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1} \mathbf{e}_N[n] \quad (5.5)$$

y multiplicando por la izquierda ambos lados de la igualdad en (5.5) por $\mathbf{A}[n]$ nos quedaría

$$\mathbf{e}_N^p[n] = \mathbf{e}_N^a[n] - \mu \mathbf{A}[n] \mathbf{A}^T[n] (\mathbf{A}[n] \mathbf{A}^T[n] + \delta I)^{-1} \mathbf{e}_N[n], \quad (5.6)$$

donde hemos introducido dos nuevos vectores de error que llamaremos error a priori ($\mathbf{e}_N^a[n]$) y error a posteriori ($\mathbf{e}_N^p[n]$) y se definen como

$$\mathbf{e}_N^a[n] = \mathbf{A}[n] \tilde{\mathbf{w}}_L[n-1] \quad (5.7)$$

$$\mathbf{e}_N^p[n] = \mathbf{A}[n] \tilde{\mathbf{w}}_L[n]. \quad (5.8)$$

De (5.6) podemos escribir

$$(\mathbf{A}[n] \mathbf{A}^T[n] + \delta I)^{-1} \mathbf{e}_N[n] = \frac{1}{\mu} (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} (\mathbf{e}_N^a[n] - \mathbf{e}_N^p[n]) \quad (5.9)$$

y sustituyendo en (5.5), obtenemos:

$$\tilde{\mathbf{w}}_L[n] = \tilde{\mathbf{w}}_L[n-1] - \mu \mathbf{A}^T[n] \frac{1}{\mu} (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} (\mathbf{e}_N^a[n] - \mathbf{e}_N^p[n]), \quad (5.10)$$

que puede ser reordenada tal y como se expresa a continuación:

$$\tilde{\mathbf{w}}_L[n] + \mathbf{A}^T[n] (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n] = \tilde{\mathbf{w}}_L[n-1] + \mathbf{A}^T[n] (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n]. \quad (5.11)$$

Si calculamos la energía, definida como el cuadrado de la norma euclídea, de cada uno de los términos de la igualdad, llegamos (como se describe en 5.6.1) a:

$$\begin{aligned} & \|\tilde{\mathbf{w}}_L[n]\|^2 + (\mathbf{e}_N^a[n])^T (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n] \\ &= \|\tilde{\mathbf{w}}_L[n-1]\|^2 + (\mathbf{e}_N^p[n])^T (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n]. \end{aligned} \quad (5.12)$$

La ecuación (5.12) es también conocida como la relación de la conservación de la energía [80], pero adaptada para este algoritmo [79]. Partiremos de dicha ecuación intentando encontrar alguna expresión para el error cuadrático medio en estado estacionario. Así, tomando esperanzas matemáticas:

$$\begin{aligned} & E\{\|\tilde{\mathbf{w}}_L[n]\|^2\} + E\{(\mathbf{e}_N^a[n])^T (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n]\} \\ &= E\{\|\tilde{\mathbf{w}}_L[n-1]\|^2\} + E\{(\mathbf{e}_N^p[n])^T (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n]\}. \end{aligned} \quad (5.13)$$

Si hacemos tender $n \rightarrow \infty$ (régimen estacionario) resulta que $E\{\|\tilde{\mathbf{w}}_L[n]\|^2\} = E\{\|\tilde{\mathbf{w}}_L[n-1]\|^2\}$ con lo que la relación expresada en la ecuación (5.13) podría reescribirse como

$$E\{(\mathbf{e}_N^a[n])^T (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n]\} = E\{(\mathbf{e}_N^p[n])^T (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n]\}. \quad (5.14)$$

Expresando el error a posteriori en función del error a priori según (5.6) en la parte derecha de la ecuación (5.14), y considerando las identidades descritas en (5.16) y (5.17), introducidas para compactar las expresiones resultantes, llegaríamos (tal y como se describe en 5.6.2) a la siguiente expresión:

$$\mu E\{\mathbf{e}_N^T[n]\Psi[n]\mathbf{e}_N[n]\} = E\{\mathbf{e}_N^T[n]\Phi[n]\mathbf{e}_N^a[n]\} + E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N[n]\}, \quad (5.15)$$

$$\Phi[n] = (\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1} \quad (5.16)$$

$$\begin{aligned} \Psi[n] &= (\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}(\mathbf{A}[n]\mathbf{A}^T[n])(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1} \\ &= \Phi[n](\mathbf{A}[n]\mathbf{A}^T[n])\Phi[n]. \end{aligned} \quad (5.17)$$

Puesto que

$$\mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n-1] \quad (5.18)$$

introduciendo (5.2) en (5.18) nos quedaría:

$$\mathbf{e}_N[n] = \mathbf{A}[n]\mathbf{w}_L^o + \mathbf{r}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n-1] = \mathbf{A}[n](\mathbf{w}_L^o - \mathbf{w}_L[n-1]) + \mathbf{r}_N[n], \quad (5.19)$$

que usando las definiciones descritas en (5.4) y (5.7) nos conduciría a la expresión:

$$\mathbf{e}_N[n] = \mathbf{e}_N^a[n] + \mathbf{r}_N[n]. \quad (5.20)$$

Sustituyendo este valor de $\mathbf{e}_N[n]$ en cada una de las expresiones que aparecen en (5.15), y operando tal y como se describe en 5.6.3, obtendríamos:

$$\begin{aligned} \mu E\{(\mathbf{e}_N^a[n])^T\Psi[n]\mathbf{e}_N^a[n]\} + \mu E\{\mathbf{r}_N^T[n]\Psi[n]\mathbf{r}_N[n]\} \\ = 2E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N^a[n]\}. \end{aligned} \quad (5.21)$$

Hay que considerar que tanto la señal $r[n]$ como $e_N^a[n]$ son estadísticamente independientes de la señal $x[n]$ (o sea, de las matrices $\Phi[n]$ y $\Psi[n]$ formadas por valores de $x[n]$). También podría haberse considerado que $\mu \approx 1$ y $\delta \approx 0$ (caso frecuente), llegándose a la misma conclusión.

Para continuar manipulando la expresión (5.21) hacemos uso de la propiedad algebraica que nos permite, dados dos vectores columna (\mathbf{a} y \mathbf{b}) de longitud cualquiera N , expresar la siguiente igualdad: $\mathbf{a}^T\mathbf{b} = Tr\{\mathbf{a}\mathbf{b}^T\}$. Este resultado puede aplicarse a cada uno de los términos de la ecuación (5.21), y teniendo en cuenta que tanto $\Phi[n]$ como $\Psi[n]$ son matrices simétricas, nos quedaría:

$$\begin{aligned} \mu Tr(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\Psi[n]\}) + \mu Tr(E\{\mathbf{r}_N[n]\mathbf{r}_N^T[n]\Psi[n]\}) \\ = 2Tr(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\Phi[n]\}). \end{aligned} \quad (5.22)$$

Considerando que tanto los datos de $\mathbf{r}_N[n]$ como los de $\mathbf{e}_N^a[n]$ son estadísticamente independientes de los de las matrices $\mathbf{\Phi}[n]$ y $\mathbf{\Psi}[n]$ (formadas por valores de $x[n]$), podemos reescribir la ecuación (5.22) como:

$$\begin{aligned} \mu \text{Tr}(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}E\{\mathbf{\Psi}[n]\}) + \mu \text{Tr}(E\{\mathbf{r}_N[n]\mathbf{r}_N^T[n]\}E\{\mathbf{\Psi}[n]\}) \\ = 2\text{Tr}(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}E\{\mathbf{\Phi}[n]\}). \end{aligned} \quad (5.23)$$

La expresión (5.23) parece todavía estar lejos de aportarnos información sobre el descriptor elegido para definir el comportamiento de los algoritmos en estado estacionario expresado en (5.3) y tampoco parece demasiado manejable. Sin embargo, bajos ciertas suposiciones, se puede simplificar aún más, puesto que podemos considerar que la matriz $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$ es diagonal (lo que se cumple si $e_N^a[n]$ se considera como un proceso aleatorio estadísticamente independiente y de media nula, o lo que es lo mismo, si consideramos los vectores $\mathbf{x}_L[n]$ y $\mathbf{x}_L[n-j]$ variables aleatorias independientes para $j \neq 0$) con lo que no sería necesario el cálculo de los elementos de las matrices $E\{\mathbf{\Psi}[n]\}$ y $E\{\mathbf{\Phi}[n]\}$ colocados fuera de la diagonal, para calcular el valor de la traza que aparecen en (5.23). Por tanto, los productos matriciales que aparecen dentro de las trazas los simplificaremos al producto de las diagonales de la matrices que se multiplican, tal y como se recoge en [79], ya que son los únicos valores que nos interesan para calcular posteriormente su traza. Tomando en cuenta esta consideración, operando sobre el primer término de la izquierda de la igualdad expresada en (5.23), el cálculo de la esperanza matemática $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$ se simplifica, puesto que no vamos a calcular los términos que queden fuera de la diagonal del producto de los dos vectores. Recordamos que el vector $\mathbf{e}_N^a[n]$ puede escribirse (desglosando la expresión (5.7)) como:

$$\mathbf{e}_N^a[n] = \begin{pmatrix} \mathbf{x}_L^T[n]\tilde{\mathbf{w}}_L[n-1] \\ \mathbf{x}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1] \\ \vdots \\ \mathbf{x}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1] \end{pmatrix}, \quad (5.24)$$

por lo que los valores de la diagonal de $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$ serán:

$$\begin{aligned} [E\{|\mathbf{x}_L^T[n]\tilde{\mathbf{w}}_L[n-1]|^2\}, E\{|\mathbf{x}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|^2\}, \\ \dots, E\{|\mathbf{x}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1]|^2\}]. \end{aligned}$$

Si consideramos el caso frecuente de que la constante de regularización es muy pequeña ($\delta \approx 0$), partiendo de (5.6) podemos llegar a la siguiente igualdad:

$$\mathbf{e}_N^p[n] \approx \mathbf{e}_N^a[n] - \mu \mathbf{e}_N[n] \quad (5.25)$$

que usando (5.20) y trasponiendo la igualdad, podemos reescribir:

$$(\mathbf{e}_N^p[n])^T = (\mathbf{e}_N^a[n])^T(1 - \mu) - \mu \mathbf{r}_N^T[n]. \quad (5.26)$$

Calculando ahora la esperanza matemática de la multiplicación de $\mathbf{e}_N^p[n]$ por $(\mathbf{e}_N^p[n])^T$, obtendremos:

$$E\{\mathbf{e}_N^p[n](\mathbf{e}_N^p[n])^T\} = (1 - \mu)^2 E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\} + \mu^2 \sigma_r^2 \mathbf{I} \quad (5.27)$$

donde \mathbf{I} representa la matriz identidad de tamaño $N \times N$, y se ha considerado la independencia estadística de $r[n]$ y $e^a[n]$.

Los elementos de la diagonal de la expresión (5.27) (que son los únicos que nos interesa calcular para poder evaluar los elementos de la diagonal en $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$), son:

$$\begin{aligned} & \begin{pmatrix} E\{|\mathbf{x}_L^T[n]\tilde{\mathbf{w}}_L[n]|^2\} \\ E\{|\mathbf{x}_L^T[n-1]\tilde{\mathbf{w}}_L[n]|^2\} \\ \vdots \\ E\{|\mathbf{x}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n]|^2\} \end{pmatrix} = \mu^2 \sigma_r^2 \\ & + (1 - \mu)^2 \begin{pmatrix} E\{|\mathbf{x}_L^T[n]\tilde{\mathbf{w}}_L[n-1]|^2\} \\ E\{|\mathbf{x}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|^2\} \\ \vdots \\ E\{|\mathbf{x}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1]|^2\} \end{pmatrix}. \end{aligned} \quad (5.28)$$

Puesto que el primer elemento del vector $\mathbf{e}_N^a[n]$, $e^a[n] = \mathbf{x}_L^T[n]\tilde{\mathbf{w}}_L[n-1]$, tendríamos que el primer elemento de la diagonal de la matriz buscada (primer elemento del vector definido en (5.24)) sería $E\{|e^a[n]|^2\}$ y que el resto de términos pueden ponerse en función de éste. Así, partiendo de la ecuación (5.28), si retrasamos una unidad los elementos de la primera fila obtenemos el valor del segundo término de la diagonal de $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$.

$$E\{|\mathbf{x}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|^2\} = (1 - \mu)^2 E\{|e^a[n-1]|^2\} + \mu^2 \sigma_r^2 \quad (5.29)$$

y usando el valor calculado en (5.29) en la segunda fila de (5.28), se llegaría a la siguiente igualdad:

$$\begin{aligned} E\{|\mathbf{x}_L^T[n-1]\tilde{\mathbf{w}}_L[n]|^2\} &= (1 - \mu)^2 E\{|\mathbf{x}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|^2\} + \mu^2 \sigma_r^2 \\ &= (1 - \mu)^2 ((1 - \mu)^2 E\{|e^a[n-1]|^2\} + \mu^2 \sigma_r^2) + \mu^2 \sigma_r^2 \\ &= (1 - \mu)^4 E\{|e^a[n-1]|^2\} + (1 - \mu)^2 \mu^2 \sigma_r^2 + \mu^2 \sigma_r^2 \end{aligned} \quad (5.30)$$

que, a su vez, si es retrasado una unidad, nos proporciona el tercer elemento de la diagonal de $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$.

$$E\{|\mathbf{x}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]|^2\} = (1 - \mu)^4 E\{|e^a[n-2]|^2\} + (1 - \mu)^2 \mu^2 \sigma_r^2 + \mu^2 \sigma_r^2. \quad (5.31)$$

De esta forma se obtendrían todos los términos de la diagonal de la matriz $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$. Considerando que en estado estacionario ($n \rightarrow \infty$),

$E\{|e^a[n]|^2\} = E\{|e^a[n-1]|^2\} = \dots = E\{|e^a[n-N+1]|^2\}$, podemos escribir de forma compacta el producto de los términos de la diagonal de $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$:

$$E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\} \approx E\{|e^a[n]|^2\}D_{11} + \mu^2\sigma_r^2D_{12} \quad (5.32)$$

Siendo D_{11} y D_{12} las siguientes matrices diagonales:

$$D_{11} = \begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & (1-\mu)^2 & 0 & \dots & \vdots \\ \vdots & 0 & (1-\mu)^4 & & \\ & & & \ddots & 0 \\ 0 & & & & (1-\mu)^{2(N-1)} \end{pmatrix} \quad (5.33)$$

y

$$D_{12} = \begin{pmatrix} 0 & \dots & & & 0 \\ 0 & 1 & 0 & \dots & \vdots \\ \vdots & 0 & 1+(1-\mu)^2 & 0 & \\ & & & \ddots & 0 \\ 0 & \dots & & 0 & 1+(1-\mu)^2+\dots+(1-\mu)^{2(N-2)} \end{pmatrix}. \quad (5.34)$$

Volviendo de nuevo a la ecuación (5.23), el primer término de la parte izquierda de la igualdad se corresponde con:

$$\begin{aligned} & Tr(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}E\{\Psi[n]\}) \\ & \approx E\{|e^a[n]|^2\}Tr(D_{11}E\{\Psi[n]\}) + Tr(D_{12}E\{\Psi[n]\})\mu^2\sigma_r^2. \end{aligned} \quad (5.35)$$

Análogamente, para el término de la derecha de (5.23) se puede escribir:

$$\begin{aligned} & Tr(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}E\{\Phi[n]\}) \\ & \approx E\{|e^a[n]|^2\}Tr(D_{11}E\{\Phi[n]\}) + Tr(D_{12}E\{\Phi[n]\})\mu^2\sigma_r^2. \end{aligned} \quad (5.36)$$

El segundo término de la izquierda de (5.23), considerando que $E\{\mathbf{r}_N[n]\mathbf{r}_N^T[n]\} \approx \sigma_r^2\mathbf{I}$, puede escribirse:

$$Tr(E\{\mathbf{r}_N[n]\mathbf{r}_N^T[n]\}E\{\Psi[n]\}) \approx \sigma_r^2Tr(E\{\Psi[n]\}) \quad (5.37)$$

y finalmente, la expresión (5.23), puede expresarse como:

$$\begin{aligned} & \mu E\{|e^a[n]|^2\}Tr(D_{11}E\{\Psi[n]\}) + \mu Tr(D_{12}E\{\Psi[n]\})\mu^2\sigma_r^2 + \mu\sigma_r^2Tr(E\{\Psi[n]\}) \\ & = 2E\{|e^a[n]|^2\}Tr(D_{11}E\{\Phi[n]\}) + 2Tr(D_{12}E\{\Phi[n]\})\mu^2\sigma_r^2. \end{aligned} \quad (5.38)$$

El límite en estado estacionario de $E\{|e^a[n]|^2\}$ se conoce como error cuadrático medio en exceso (EMSE, *Excess Mean Square Error*)

$$\text{EMSE} = \lim_{n \rightarrow \infty} E\{|e^a[n]|^2\} \quad (5.39)$$

y la relación que existe entre el EMSE y el MSE descrito en (5.3) es:

$$\text{MSE} = \text{EMSE} + \sigma_r^2, \quad (5.40)$$

lo que se puede demostrar a partir de (5.20), sin más que realizar el límite de la esperanza matemática de la multiplicación de cada término de la igualdad por su traspuesto, y quedarnos con el primer elemento de la igualdad matricial que aparece. Por lo tanto, a partir del conocimiento de uno de los errores se puede deducir el otro.

De la ecuación (5.38) podemos despejar $E\{|e^a[n]|^2\}$, resultando:

$$\begin{aligned} & E\{|e^a[n]|^2\} \\ &= \frac{\mu\sigma_r^2 \text{Tr}(E\{\Psi[n]\}) + (\mu\text{Tr}(D_{12}E\{\Psi[n]\}) - 2\text{Tr}(D_{12}E\{\Phi[n]\}))\mu^2\sigma_r^2}{2\text{Tr}(D_{11}E\{\Phi[n]\}) - \mu\text{Tr}(D_{11}E\{\Psi[n]\})}. \end{aligned} \quad (5.41)$$

Tomando límites en (5.41) se obtiene el valor del EMSE.

Finalmente, para obtener una expresión lo más compacta posible, consideramos que la varianza de la señal $r[n]$, (σ_r^2) es despreciable frente a $E\{|e^a[n]|^2\}$ por lo que podemos eliminar la aportación del segundo término de la derecha de la igualdad expresada en (5.32). De esta forma, el primer término de (5.23) quedaría:

$$\text{Tr}(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}E\{\Psi[n]\}) \approx E\{|e^a[n]|^2\}\text{Tr}(D_{11}E\{\Psi[n]\}) \quad (5.42)$$

y análogamente, para el término de la derecha de (5.23) se puede escribir:

$$\text{Tr}(E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}E\{\Phi[n]\}) \approx E\{|e^a[n]|^2\}\text{Tr}(D_{11}E\{\Phi[n]\}), \quad (5.43)$$

lo que permite alcanzar una expresión para $E\{|e^a[n]|^2\}$ más manejable:

$$E\{|e^a[n]|^2\} = \frac{\mu\sigma_r^2 \text{Tr}(E\{\Psi[n]\})}{2\text{Tr}(D_{11}E\{\Phi[n]\}) - \mu\text{Tr}(D_{11}E\{\Psi[n]\})}. \quad (5.44)$$

Dicha expresión nos permite obtener una estimación del EMSE en función de las estadísticas de los datos de entrada ($x[n]$) y del ruido residual ($r[n]$). Se pueden hacer algunas aproximaciones más si consideramos condiciones especiales de funcionamiento. Por ejemplo, para el caso habitual de un valor de

regularización pequeño ($\delta \approx 0$), resulta que $\Phi[n] = \Psi[n] = (\mathbf{A}[n]\mathbf{A}^T[n])^{-1}$ y el valor del EMSE puede expresarse como:

$$\text{EMSE} = \lim_{n \rightarrow \infty} \left(\frac{\mu\sigma_r^2}{2 - \mu} \right) \left(\frac{\text{Tr}(E\{\Psi[n]\})}{\text{Tr}(D_{11}E\{\Psi[n]\})} \right), \quad (5.45)$$

expresión que se simplifica más si sustituimos los valores estimados para D_{11} en función de μ , considerando los siguientes dos casos:

- Si $\mu \approx 0$ entonces D_{11} es simplemente la matriz identidad.
- Si $\mu \approx 1$ entonces D_{11} es una matriz con todos sus elementos a cero excepto el que ocupa la posición (1,1) que vale 1.

Así,

- Si $\mu \approx 0$

$$\text{EMSE} = \frac{\mu\sigma_r^2}{2 - \mu} \quad (5.46)$$

- Si $\mu \approx 1$

$$\text{EMSE} = \left(\frac{\mu\sigma_r^2}{2 - \mu} \right) \lim_{n \rightarrow \infty} \left(\frac{\text{Tr}(E\{\Psi[n]\})}{E\{\Psi[1, 1]\}} \right) \quad (5.47)$$

donde $\Psi[1, 1]$ denota el elemento (1,1) de $\Psi[n]$.

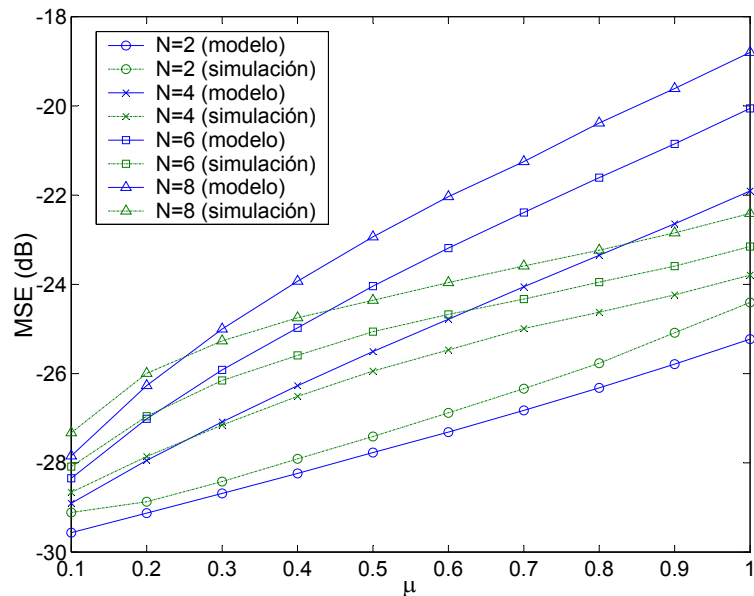
Aunque estos resultados obtenidos para el EMSE no son originales puestos que ya han sido publicados en [79] y [81] y admiten alguna aproximación más como la publicada recientemente en [82], se ha descrito con detalle la metodología seguida para alcanzarlos porque comparte gran parte del desarrollo seguido en el estudio del MSE de este algoritmo aplicado al CAR.

Las conclusiones principales que se pueden obtener de estas expresiones para el cálculo de una estimación del EMSE son que, ésta es creciente con el valor de μ y también con el orden de proyección N , puesto que las trazas que aparecen en los numeradores de las expresiones involucran tantos elementos como valor tenga el orden de proyección. Por lo tanto, el valor del error cuadrático medio en estado estacionario crecerá tanto con el orden de proyección del algoritmo como con el valor del paso de convergencia, por lo que estos parámetros (que controlan la velocidad de convergencia) deberán ser minimizados si lo que interesa es obtener un mínimo valor del error cuadrático medio.

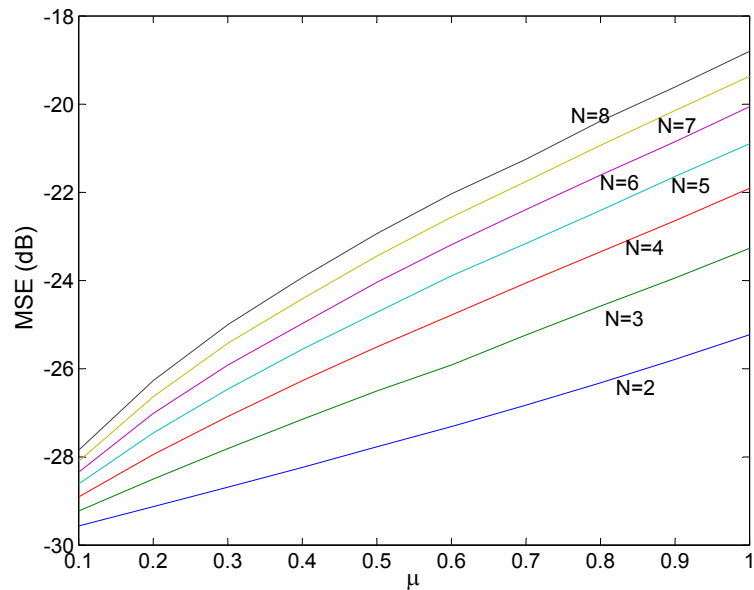
Para contrastar los resultados obtenidos del estimador del MSE en estado estacionario, se han realizado simulaciones con el algoritmo de proyección

afín (usando un filtro adaptativo de 16 coeficientes) para diferentes valores de la constante de convergencia μ y órdenes de proyección N . Como señal de referencia se ha usado ruido Gaussiano (de varianza unidad y media cero) coloreado tras ser filtrado por un filtro autorregresivo de primer orden con función de transferencia $\sqrt{1-a^2}/(1-az^{-1})$ con $a=0.9$. La señal deseada $\mathbf{d}[n]$ se ha generado siguiendo el modelo descrito en (5.2) usando ruido Gaussiano de varianza $\sigma_r^2=0.001$. Para obtener los valores de MSE experimentalmente se han realizado simulaciones con 200.000 iteraciones (y usando un factor de regularización de 0.0001).

Se han comparado los valores obtenidos experimentalmente con los proporcionados por el valor del MSE (5.40), usando, para el cálculo del EMSE, el modelo expresado en (5.47), ya que las simulaciones han sido realizadas bajo un régimen de funcionamiento donde el parámetro de convergencia μ era grande. En la figura 5.1 (a) se representan las curvas para el MSE, donde se refleja como el modelo teórico hallado se ajusta mejor a los valores obtenidos mediante simulación para órdenes de proyección pequeños. No obstante, dichas curvas revelan la validez del modelo. En la figura 5.1 (b) se puede observar la tendencia del MSE tanto con el orden de proyección como con el valor de la constante de convergencia. Como cabía esperar, dicho error es menor conforme la constante de convergencia disminuye, y crece conforme el orden de proyección es mayor.



(a)



(b)

Figura 5.1: Curvas del MSE empleando el modelo (5.47) (a) para distintos órdenes de proyección (línea continua) comparada con los valores experimentales (línea discontinua), (b) para diferentes órdenes de proyección comparadas entre sí.

5.3. Estudio del error cuadrático medio del algoritmo de proyección afín aplicado al CAR

Veamos ahora cómo afecta al valor del EMSE (y por consiguiente, al MSE) las modificaciones que se realizan en los algoritmos de proyección afín cuando éstos se aplican a un problema de control activo de ruido. En este caso habría que tener en cuenta una serie de cambios de signo debido a que la señal de error es ahora la mezcla acústica entre la señal deseada y la generada después de ser filtrada por un camino acústico (que enlazaría el actuador con el sensor de error). Los cambios de signos se contrarrestan debido a que en la actualización de los coeficientes también se invertiría el signo, llegando a las mismas ecuaciones finales que las descritas en el punto anterior. La existencia del camino acústico también introduce modificaciones en los datos de la señal de referencia $x(n)$ puesto que habría que cambiarlos por los valores de dicha señal filtrados por dicho camino. En las estructuras de control más comunes, la señal de referencia está filtrada por estimaciones de dicho camino acústico, pero para el siguiente análisis, supondremos que las estimaciones son suficientemente precisas y consideraremos caminos acústicos y estimados idénticos. Con estas condiciones, las modificaciones que habría que introducir en las ecuaciones descritas en el punto anterior se limitarían a cambiar la matriz de datos $\mathbf{A}[n]$ por su homóloga $\mathbf{V}[n]$ compuesta por datos de la señal $x(n)$ filtrada por el camino acústico tal y como se describe en (5.48).

$$\mathbf{V}^T[n] = [\mathbf{v}_L[n], \mathbf{v}_L[n-1], \dots, \mathbf{v}_L[n-N+1]] \quad (5.48)$$

siendo $\mathbf{v}_L[n]$ un vector columna que contiene L elementos de la señal $x(n)$ filtrada por el camino acústico (igual que el definido en la sección 4.2.2). Esta nueva matriz implica que habría que redefinir las matrices $\Phi[n]$ y $\Psi[n]$ descritas en (5.16) y (5.17). Así,

$$\Phi_v[n] = (\mathbf{V}[n]\mathbf{V}^T[n] + \delta I)^{-1} \quad (5.49)$$

y

$$\Psi_v[n] = (\mathbf{V}[n]\mathbf{V}^T[n] + \delta I)^{-1}(\mathbf{V}[n]\mathbf{V}^T[n])(\mathbf{V}[n]\mathbf{V}^T[n] + \delta I)^{-1}, \quad (5.50)$$

mientras que la ecuación de actualización de los coeficientes de los filtros sería:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] - \mu \mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta I)^{-1} \mathbf{e}_N[n], \quad (5.51)$$

siendo $\tilde{\mathbf{w}}_L[n] = \tilde{\mathbf{w}}_L[n-1] + \mu \mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta I)^{-1} \mathbf{e}_N[n]$.

De la misma forma a lo expuesto en el apartado anterior, podríamos llegar a una expresión para el valor del EMSE como la descrita en (5.41)

$$\begin{aligned}
 & E\{|e_v^a[n]|^2\} \\
 &= \frac{\mu\sigma_r^2 \text{Tr}(E\{\Psi_v[n]\}) + (\mu\text{Tr}(D_{12}E\{\Psi_v[n]\}) - 2\text{Tr}(D_{12}E\{\Phi_v[n]\}))\mu^2\sigma_r^2}{2\text{Tr}(D_{11}E\{\Phi_v[n]\}) - \mu\text{Tr}(D_{11}E\{\Psi_v[n]\})}.
 \end{aligned} \tag{5.52}$$

Hemos considerado el nuevo vector de error a priori definido como:

$$\mathbf{e}_{vN}^a[n] = \mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1] \tag{5.53}$$

cuyo primer elemento $\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]$ llamamos $e_v^a[n]$, y el vector de error a posterior (que más tarde se usará):

$$\mathbf{e}_{vN}^p[n] = \mathbf{V}[n]\tilde{\mathbf{w}}_L[n]. \tag{5.54}$$

También tendremos que tener en cuenta que, para aprovechar la nomenclatura definida anteriormente, la señal $\mathbf{d}_N[n]$ la definiremos ahora como $\mathbf{d}_N[n] = -\mathbf{V}[n]\mathbf{w}_L^0 + \mathbf{r}_N[n]$, donde el cambio de signo se debe a que en control activo de ruido se produce una suma acústica entre la señal deseada y la generada para proporcionar la señal de error. Por lo tanto, la señal de error $\mathbf{e}_N[n]$, en el caso más genérico, sería:

$$\begin{aligned}
 \mathbf{e}_N[n] &= \mathbf{d}_N[n] + \mathbf{V}[n]\mathbf{w}_L[n-1] = -\mathbf{V}[n]\mathbf{w}_L^0 + \mathbf{r}_N[n] + \mathbf{V}[n]\mathbf{w}_L[n-1] \\
 &= -\mathbf{V}[n](\mathbf{w}_L^0 - \mathbf{w}_L[n-1]) + \mathbf{r}_N[n] = -\mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1] + \mathbf{r}_N[n].
 \end{aligned} \tag{5.55}$$

La expresión (5.55), empleada en alcanzar el modelo del EMSE descrito en (5.52), sería válida cuando se usa la estructura de filtrado-x modificada, ya que las ecuaciones que describen al algoritmo usando dicha estructura son similares (con las salvedades antes comentadas) a las expuestas en el punto anterior para el caso de identificación de sistemas.

En cambio, si usamos estrategias eficientes para actualizar los coeficientes, hay que tener en cuenta que el cálculo eficiente del vector de error se basa en aproximaciones que modifican la expresión (5.55), alterando el modelo alcanzado para el MSE. Por tanto, el cálculo del MSE de las estrategias eficientes que redefinen el vector $\mathbf{e}_N[n]$ es susceptible de estudio, mientras que el resto de estrategias (cálculo eficiente de la inversión matricial o uso de los coeficientes auxiliares) en realidad solo optimizan el coste computacional sin introducir cambios en el resultado del MSE. También el uso de la estructura de filtrado-x modificaría la expresión (5.55). En los siguientes puntos extenderemos los resultados obtenidos en el punto 5.2 tanto para la estructura modificada como para la de filtrado-x, usando el cálculo eficiente del vector de error y también sin usarlo.

5.3.1. MSE usando estructura modificada

Como ya se ha comentado, la expresión (5.52) nos proporcionaría una estimación del EMSE en un algoritmo de control activo de ruido usando la estructura modificada y el algoritmo de proyección afín. Además, todas las aproximaciones aplicadas en el punto anterior, que nos conducían a expresiones particulares del EMSE, también podrían usarse en este caso, por lo que se obtendrían expresiones idénticas a las ya obtenidas sin más que sustituir las matrices de datos $\Psi[n]$ y $\Phi[n]$ por sus correspondientes matrices equivalentes $\Psi_v[n]$ y $\Phi_v[n]$, y que se resumen a continuación.

Primeramente, si consideramos que la varianza del ruido σ_r^2 es despreciable frente al valor de $E\{|e_v^a[n]|^2\}$ (al igual que lo realizado en (5.41) para obtener (5.44)), obtenemos:

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 \text{Tr}(E\{\Psi_v[n]\})}{2\text{Tr}(D_{11}E\{\Phi_v[n]\}) - \mu\text{Tr}(D_{11}E\{\Psi_v[n]\})}. \quad (5.56)$$

Usando un valor de regularización pequeño ($\delta \approx 0$), tendríamos, como en (5.45),

$$E\{|e_v^a[n]|^2\} = \left(\frac{\mu\sigma_r^2}{2-\mu}\right) \left(\frac{\text{Tr}(E\{\Psi_v[n]\})}{\text{Tr}(D_{11}E\{\Psi_v[n]\})}\right) \quad (5.57)$$

- Si $\mu \approx 0$

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2}{2-\mu}$$

- Si $\mu \approx 1$

$$E\{|e_v^a[n]|^2\} = \left(\frac{\mu\sigma_r^2}{2-\mu}\right) \left(\frac{\text{Tr}(E\{\Psi_v[n]\})}{E\{\Psi_v[1,1]\}}\right).$$

Los límites (cuando $n \rightarrow \infty$) de la expresiones expuestas para $E\{|e_v^a[n]|^2\}$ nos darían los valores del EMSE.

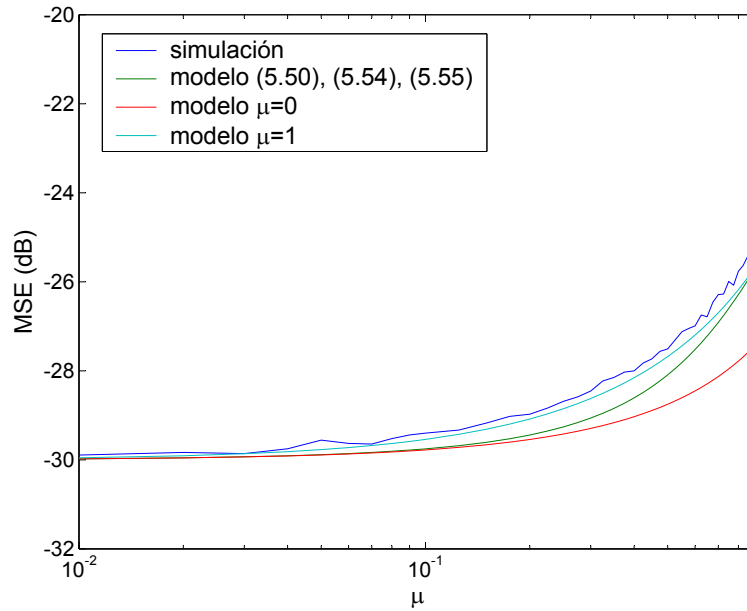


Figura 5.2: Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.

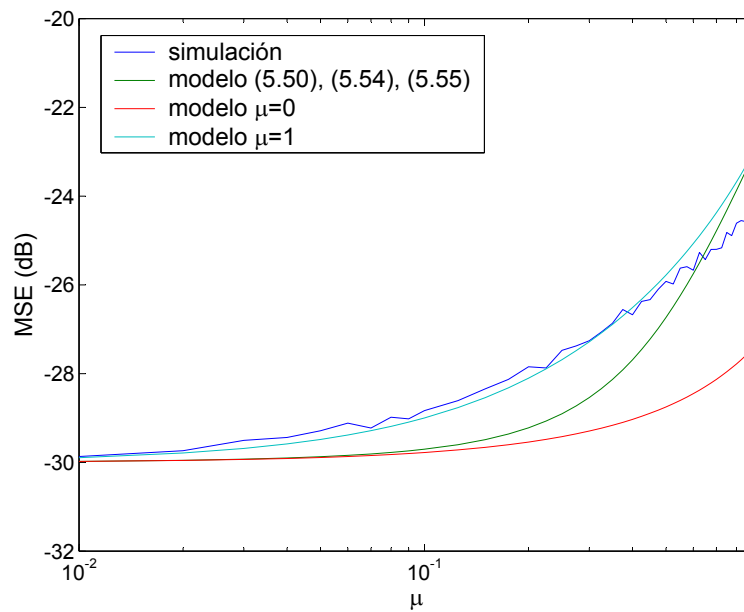


Figura 5.3: Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.

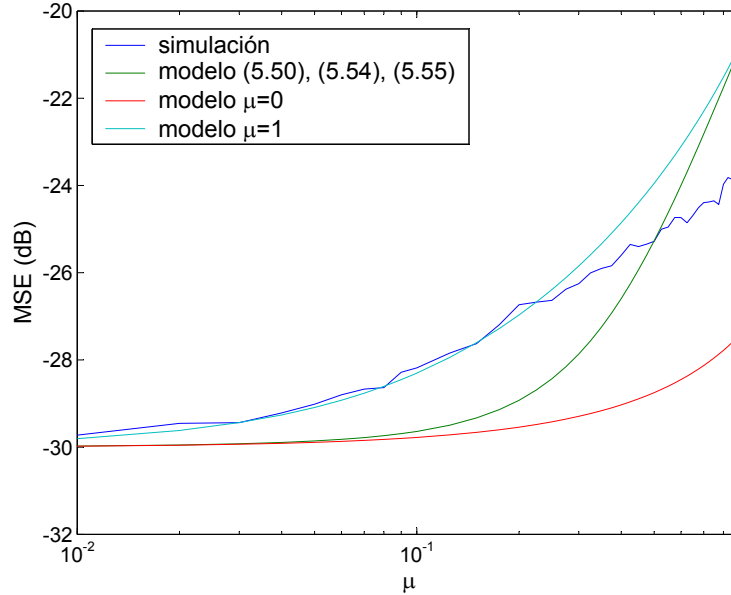


Figura 5.4: Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.

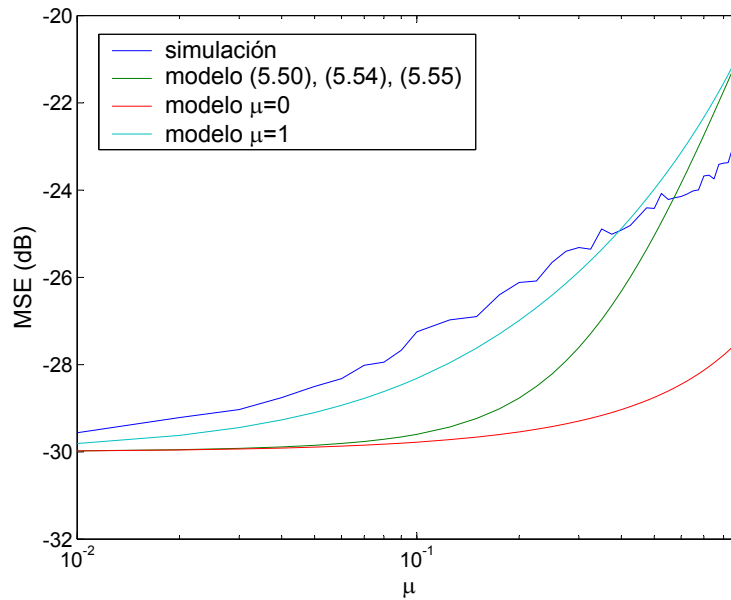


Figura 5.5: Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.

Realizando simulaciones similares a las descritas en el punto anterior (pero para un sistema de CAR monocanal, introduciendo un camino secundario modelado mediante un filtro FIR), obtenemos las figuras 5.2 - 5.5 donde se comparan los resultados analíticos proporcionados por los diferentes modelos y los obtenidos mediante simulación, para diferentes órdenes de proyección. En este caso, el eje de ordenadas está en escala logarítmica para que pueda apreciarse el comportamiento del modelo y los resultados de simulación en márgenes del parámetro de convergencia pequeños ($0.01 < \mu < 0.1$), para los que se han tomado tantos puntos como en el resto del intervalo ($0.1 < \mu < 0.9$).

En dichas figuras podemos apreciar como el modelo genérico (que para el factor de regularización usado, $\delta = 10^{-6}$, coincide con los modelos expresados en (5.56) y (5.57)) se aproxima bastante bien a los valores obtenidos mediante simulación, sobre todo para valores de μ pequeños, donde todos los modelos ofrecen buenos resultados. No obstante, el modelo que mejor aproxima los resultados obtenidos mediante simulación es el resultante de la aproximación $\mu \approx 1$, en (5.57). Por otro lado, la similitud de los datos obtenidos mediante simulación con la que los modelos aportan es bastante mejor para órdenes de proyección bajos, mientras que para órdenes altos algunos de los modelos distan mucho de los valores simulados (hasta casi 4 dB).

Si usamos la actualización eficiente del vector del error (como se describe en 3.5.1), el MSE cambia, puesto que según se desprende de (3.53), el vector de error tiene una expresión diferente. Así:

$$\begin{aligned} \mathbf{e}_N[n] &= \begin{pmatrix} e[n] \\ (1 - \mu)\bar{\mathbf{e}}_N[n - 1] \end{pmatrix} \\ &= \begin{pmatrix} d[n] + \mathbf{v}_L[n]^T \mathbf{w}_L[n - 1] \\ (1 - \mu)(d[n - 1] + \mathbf{v}_L[n - 1]^T \mathbf{w}_L[n - 2]) \\ \vdots \\ (1 - \mu)(d[n - N + 1] + \mathbf{v}_L[n - N + 1]^T \mathbf{w}_L[n - 2]) \end{pmatrix} \end{aligned} \quad (5.58)$$

y usando la $\mathbf{d}_N[n] = -\mathbf{V}[n]\mathbf{w}_L^0 + \mathbf{r}_N[n]$, nos permite reescribir

$$\mathbf{e}_N[n] = \begin{pmatrix} -\mathbf{v}_L[n]^T \tilde{\mathbf{w}}_L[n - 1] + r[n] \\ (1 - \mu)(-\mathbf{v}_L[n - 1]^T \tilde{\mathbf{w}}_L[n - 2]) + (1 - \mu)r[n - 1] \\ \vdots \\ (1 - \mu)(-\mathbf{v}_L[n - N + 1]^T \tilde{\mathbf{w}}_L[n - 2]) + (1 - \mu)r[n - N + 1] \end{pmatrix}. \quad (5.59)$$

Definiendo los siguientes vectores:

$$\mathbf{e}'_N[n] = \begin{pmatrix} \mathbf{v}_L[n]^T \tilde{\mathbf{w}}_L[n-1] \\ (1-\mu)(\mathbf{v}_L[n-1]^T \tilde{\mathbf{w}}_L[n-2]) \\ \vdots \\ (1-\mu)(\mathbf{v}_L[n-N+1]^T \tilde{\mathbf{w}}_L[n-2]) \end{pmatrix} \quad (5.60)$$

y

$$\mathbf{r}'_N[n] = \begin{pmatrix} r[n] \\ (1-\mu)r[n-1] \\ \vdots \\ (1-\mu)r[n-N+1] \end{pmatrix}, \quad (5.61)$$

el vector de error puede finalmente expresarse como $\mathbf{e}_N[n] = -\mathbf{e}'_N[n] + \mathbf{r}'_N[n]$.

Repitiendo los pasos realizados en (5.11) -(5.21), con los nuevos vectores y matrices definidos en (5.49), (5.50) y (5.59), y considerando los vectores (5.53) y (5.54) en vez de (5.7) y (5.8), llegamos a la siguiente expresión equivalente a (5.21)

$$\begin{aligned} \mu E\{(\mathbf{e}'_N[n])^T \Psi_v[n] \mathbf{e}'_N[n]\} + \mu E\{\mathbf{r}'_N[n]^T \Psi_v[n] \mathbf{r}'_N[n]\} \\ = 2E\{(\mathbf{e}'_N[n])^T \Phi_v[n] \mathbf{e}'_N[n]\} \end{aligned} \quad (5.62)$$

, que de igual forma que en (5.23), si suponemos que existe independencia estadística entre los datos de los vectores $\mathbf{r}'_N[n]$ y $\mathbf{e}'_N[n]$ con los de la matriz $\Psi_v[n]$ y de los datos de $\mathbf{e}'_N[n]$ y $\mathbf{e}'_N[n]$ con los de $\Phi_v[n]$, se puede escribir como:

$$\begin{aligned} \mu Tr(E\{\mathbf{e}'_N[n](\mathbf{e}'_N[n])^T\} E\{\Psi_v[n]\}) + \mu Tr(E\{\mathbf{r}'_N[n]\mathbf{r}'_N[n]^T\} E\{\Psi_v[n]\}) \\ = 2Tr(E\{\mathbf{e}'_N[n](\mathbf{e}'_N[n])^T\} E\{\Phi_v[n]\}). \end{aligned} \quad (5.63)$$

Operando tal y como se describe en la sección 5.6.4, podemos llegar a la siguiente estimación del EMSE:

$$\begin{aligned} E\{|e_v^a[n]|^2\} \\ = \frac{\mu \sigma_r^2 Tr(D_{25} E\{\Psi_v[n]\}) + (\mu Tr(D_{22} E\{\Psi_v[n]\}) - 2Tr(D_{24} E\{\Phi_v[n]\})) \mu^2 \sigma_r^2}{2Tr(D_{23} E\{\Phi_v[n]\}) - \mu Tr(D_{21} E\{\Psi_v[n]\})}, \end{aligned} \quad (5.64)$$

siendo las matrices D_{21} , D_{22} , D_{23} , D_{24} y D_{25} las definidas en (5.181), (5.182), (5.184), (5.185) y (5.187) respectivamente.

La expresión (5.64) puede simplificarse al igual que se hizo en la ecuación (5.56). Así, si consideramos que la varianza del ruido σ_r^2 es despreciable frente $E\{|e_v^a[n]|^2\}$ en (5.180) y (5.183), obtenemos:

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 \text{Tr}(D_{25}E\{\Psi_v[n]\})}{2\text{Tr}(D_{23}E\{\Phi_v[n]\}) - \mu\text{Tr}(D_{21}E\{\Psi_v[n]\})}. \quad (5.65)$$

Usando un valor de regularización pequeño ($\delta \approx 0$), tendríamos:

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 \text{Tr}(D_{25}E\{\Psi_v[n]\})}{2\text{Tr}(D_{23}E\{\Psi_v[n]\}) - \mu\text{Tr}(D_{21}E\{\Psi_v[n]\})}. \quad (5.66)$$

Dicha expresión puede aproximarse a $\frac{\mu\sigma_r^2}{2-\mu}$, tanto para valores pequeños de μ ($\mu \approx 0$) como para valores grandes ($\mu \approx 1$), ya que en ambos casos las matrices D_{21} , D_{23} y D_{25} serían iguales ente sí (la matriz identidad en el caso de $\mu \approx 0$ y una matriz de ceros excepto el primer elemento que es 1, para el caso de $\mu \approx 1$). No obstante, los valores del modelo para ($\mu \approx 1$) no son relevantes, puesto que en la mayoría de los casos el algoritmo diverge en dichos márgenes de funcionamiento.

Usando los modelos mencionados y procediendo del mismo modo que el descrito anteriormente para las simulaciones realizadas sin usar la actualización eficiente del vector de error, considerando solo los valores de la constante de convergencia, μ , que no provocan divergencia, se obtienen las gráficas representadas en las figuras 5.6-5.9. Aunque para los valores de μ en que las simulaciones no convergen no tenemos datos que representar en las gráficas, se ha optado por mantener el mismo rango de variación del eje de ordenadas (donde se representan los valores de μ) que en el resto de simulaciones para favorecer la comparación de resultados entre las mismas, aun cuando no se aprovecha el margen dinámico disponible en dichas gráficas.

Como puede observarse, los modelos obtenidos ofrecen resultados para el MSE algo más optimistas que las simulaciones. Sin duda, esta variante del algoritmo es más sensible ante las hipótesis consideradas por lo que los resultados son algo peores que para el caso de la estructura modificada sin cálculo eficiente del vector de error. No obstante, para valores de μ y órdenes de proyección pequeños el modelo ofrece estimaciones del MSE suficientemente buenas.

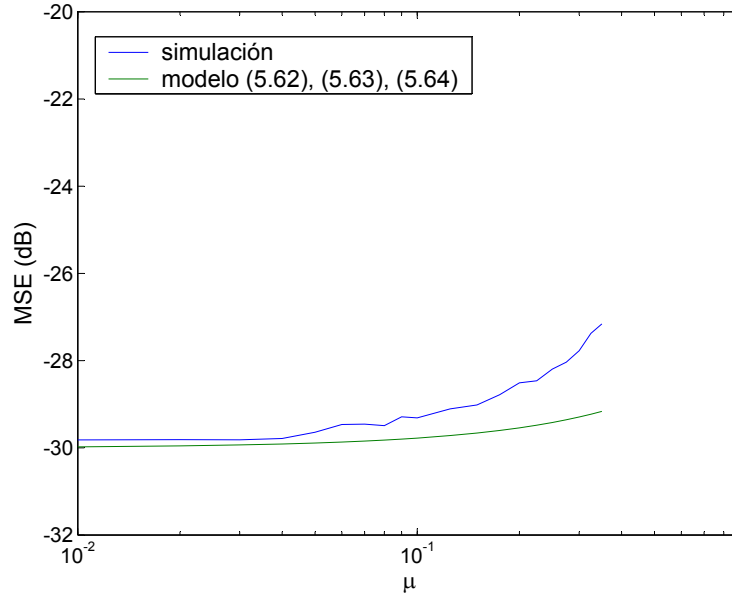


Figura 5.6: Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.

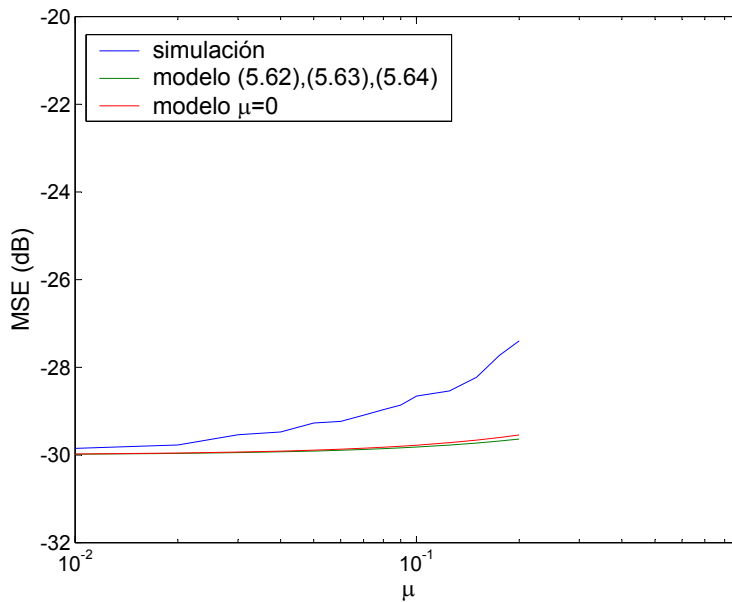


Figura 5.7: Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.

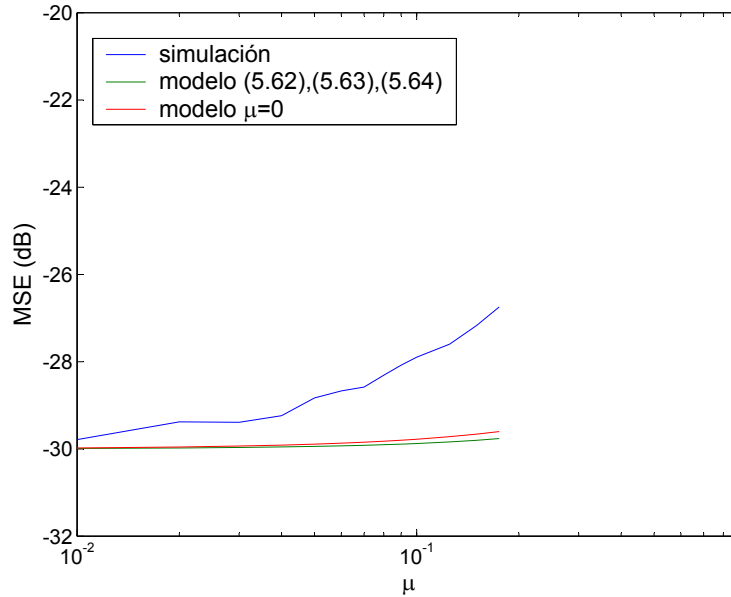


Figura 5.8: Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.

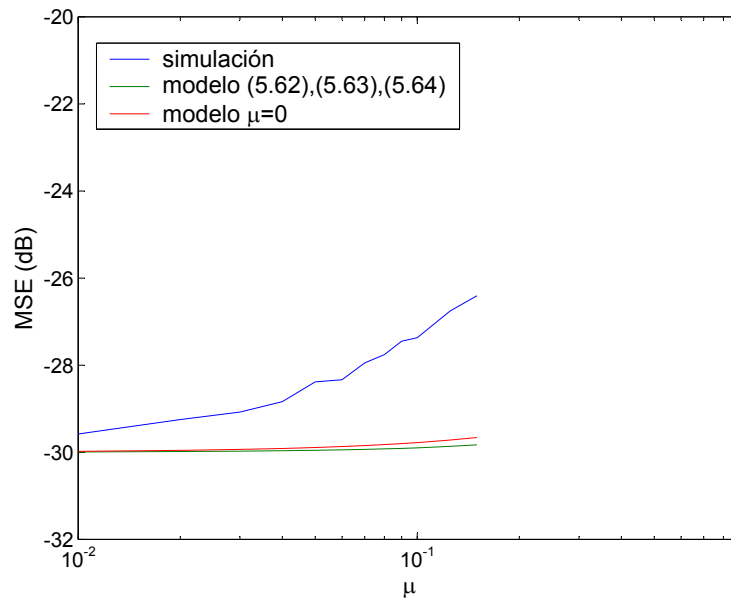


Figura 5.9: Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.

Bajo las condiciones de simulación consideradas, los diferentes modelos parecen converger a una única solución para órdenes de proyección pequeños, mientras que para órdenes grandes, se aprecian pequeñas diferencias entre ellos. En el margen de los valores de μ donde no se produce divergencia en las simulaciones, todos los modelos ofrecen un comportamiento parecido. Hasta $\mu = 0.2$ la diferencia entre los modelos y el valor del MSE obtenido mediante simulación se mantiene por debajo de los 2 dB.

5.3.2. MSE usando estructura de filtrado-x

Cuando se aplica la estructura de filtrado-x debemos, de nuevo, considerar un vector de error diferente del empleado en el algoritmo de proyección afín estándar ($\mathbf{e}_N[n] = \mathbf{d}_N[n] + \mathbf{V}[n]\mathbf{w}_L[n-1]$). En este caso dicho vector estaría formado directamente por los datos de la señal recogida por el sensor de error. Matemáticamente, se puede expresar como:

$$\begin{aligned} \mathbf{e}_N[n] &= \mathbf{d}_N[n] + \begin{pmatrix} \mathbf{v}_L^T[n]\mathbf{w}_L[n-1] \\ \mathbf{v}_L^T[n-1]\mathbf{w}_L[n-2] \\ \vdots \\ \mathbf{v}_L^T[n-N+1]\mathbf{w}_L[n-N] \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{v}_L^T[n](\mathbf{w}_L[n-1] - \mathbf{w}_L^0) \\ \mathbf{v}_L^T[n-1](\mathbf{w}_L[n-2] - \mathbf{w}_L^0) \\ \vdots \\ \mathbf{v}_L^T[n-N+1](\mathbf{w}_L[n-N] - \mathbf{w}_L^0) \end{pmatrix} + \mathbf{r}_N[n], \end{aligned} \quad (5.67)$$

siendo $\mathbf{d}_N[n] = -\mathbf{V}[n]\mathbf{w}_L^0 + \mathbf{r}_N[n]$. El vector $\mathbf{e}_N[n]$ puede escribirse, igual que en el caso anterior como $\mathbf{e}_N[n] = -\mathbf{e}_N^{\alpha'}[n] + \mathbf{r}_N[n]$, siendo $\mathbf{e}_N^{\alpha'}[n]$

$$\mathbf{e}_N^{\alpha'}[n] = \begin{pmatrix} \mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1] \\ \mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2] \\ \vdots \\ \mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N] \end{pmatrix} = \begin{pmatrix} e_v^{\alpha}[n] \\ e_v^{\alpha}[n-1] \\ \vdots \\ e_v^{\alpha}[n-N+1] \end{pmatrix}, \quad (5.68)$$

lo que permite seguir el desarrollo realizado anteriormente a partir de la ecuación (5.15). Considerando el nuevo vector $\mathbf{e}_N^{\alpha'}[n]$, podemos proceder de forma análoga a 5.6.3, y usando los nuevos vectores y matrices definidos en (5.49), (5.50), (5.53), (5.54) y (5.68) (considerando independencia estadística entre los datos de los vectores y matrices $\mathbf{r}_N[n]$, $\mathbf{e}_N^{\alpha'}[n]$ y $\Psi_v[n]$, y $\mathbf{r}_N[n]$, $\mathbf{e}_{vN}^{\alpha}[n]$ y $\Phi_v[n]$) llegamos a la siguiente expresión equivalente a (5.21),

$$\begin{aligned} \mu E\{(\mathbf{e}'_N[n])^T \boldsymbol{\Psi}_v[n] \mathbf{e}'_N[n]\} + \mu E\{\mathbf{r}_N^T[n] \boldsymbol{\Psi}_v[n] \mathbf{r}_N[n]\} \\ = 2E\{(\mathbf{e}'_N[n])^T \boldsymbol{\Phi}_v[n] \mathbf{e}'_N[n]\} \end{aligned} \quad (5.69)$$

que de igual forma que en (5.23) se puede escribir como:

$$\begin{aligned} \mu Tr(E\{\mathbf{e}'_N[n](\mathbf{e}'_N[n])^T\}E\{\boldsymbol{\Psi}_v[n]\}) + \mu Tr(E\{\mathbf{r}_N[n]\mathbf{r}_N^T[n]\}E\{\boldsymbol{\Psi}_v[n]\}) \\ = 2Tr(E\{\mathbf{e}'_N[n](\mathbf{e}'_N[n])^T\}E\{\boldsymbol{\Phi}_v[n]\}). \end{aligned} \quad (5.70)$$

Operando tal y como se expresa en 5.6.5, obtenemos el siguiente modelo para la obtención del EMSE.

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 Tr(E\{\boldsymbol{\Psi}_v[n]\})}{2Tr(E\{D_{33}\boldsymbol{\Phi}_v[n]\}) - \mu Tr(E\{\boldsymbol{\Psi}_v[n]\})}. \quad (5.71)$$

Siendo D_{33} la matriz diagonal

$$D_{33} = \begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & (1-\mu) & 0 & \dots & \vdots \\ \vdots & & (1-2\mu) & & \\ & & & \ddots & 0 \\ 0 \dots & & & 0 & [1-(N-1)\mu] \end{pmatrix}. \quad (5.72)$$

El límite cuando $n \rightarrow \infty$ de la expresión anterior nos daría el valor del EMSE buscado. Suponiendo que $\delta \approx 0$, podemos aproximar la expresión del EMSE como:

$$\begin{aligned} \text{EMSE} &= \lim_{n \rightarrow \infty} \left(\mu\sigma_r^2 \frac{Tr(E\{\boldsymbol{\Psi}_v[n]\})}{Tr((2D_{33} - \mu\mathbf{I})E\{\boldsymbol{\Psi}_v[n]\})} \right) \\ &= \lim_{n \rightarrow \infty} \left(\mu\sigma_r^2 \frac{Tr(E\{\boldsymbol{\Psi}_v[n]\})}{Tr(D_{34}E\{\boldsymbol{\Psi}_v[n]\})} \right), \end{aligned} \quad (5.73)$$

donde \mathbf{I} es la matriz identidad $N \times N$ y D_{34} la matriz

$$D_{34} = \begin{pmatrix} (2-\mu) & 0 & \dots & & 0 \\ 0 & (2-3\mu) & 0 & \dots & \vdots \\ \vdots & & (2-5\mu) & & \\ & & & \ddots & 0 \\ 0 & \dots & & 0 & [2-(2N-1)\mu] \end{pmatrix}. \quad (5.74)$$

Debe hacerse notar que existen valores de μ que proporcionarían valores negativos para el denominador de (5.71) y (5.73) (por ejemplo, los que hacen que aparezcan en la diagonal de (5.74) valores negativos). Estos valores, estarían en los márgenes de divergencia del algoritmo, por lo que no tendrían sentido para estimar el EMSE, puesto que el algoritmo diverge en esos casos. Por tanto, centrándonos en el modelo expresado en 5.73, solo tendría sentido para los valores que cumplen $\mu > \frac{2}{2N-1}$. Por otro lado, si $\mu \approx 0$ el EMSE para la estructura de filtrado-x se aproxima mucho al hallado para la estructura modificada:

EMSE = $\frac{\mu\sigma_r^2}{2}$ (a), o EMSE = $\frac{\mu\sigma_r^2}{2-\mu}$ (b) si partimos de la expresión (5.71). La aproximación $\mu \approx 1$ no tiene sentido en este caso, puesto que para dichos valores del parámetro de convergencia el algoritmo diverge.

Usando los modelos descritos en este apartado para obtener los valores del MSE, y comparándolos con valores del MSE procedente de simulaciones, obtenemos las curvas de las figuras reflejadas en 5.10-5.13.

De nuevo, los modelos ofrecen resultados más aproximados a las simulaciones para órdenes y valores del parámetro de convergencia pequeños. Conforme μ crece, estos modelos parecen alejarse de los resultados de las simulaciones, pero tenemos que ser conscientes, que dicho parámetro de convergencia no puede ser muy grande con dicha estructura, puesto que no garantizaría la convergencia. De hecho, mediante simulación, el algoritmo diverge antes del valor $\mu = \frac{2}{2N-1}$ comentado anteriormente. Así, para el régimen de funcionamiento normal de este algoritmo (μ no muy grandes), los modelos ofrecen aproximaciones suficientemente buenas.

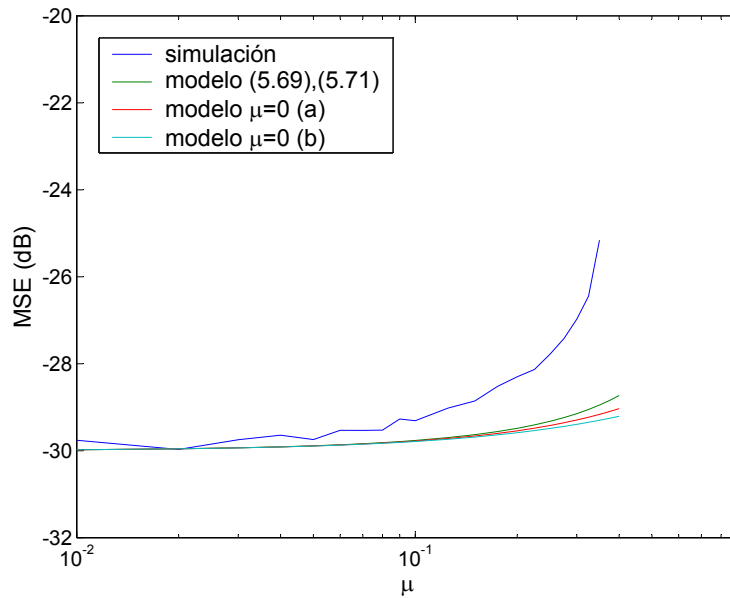


Figura 5.10: Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.

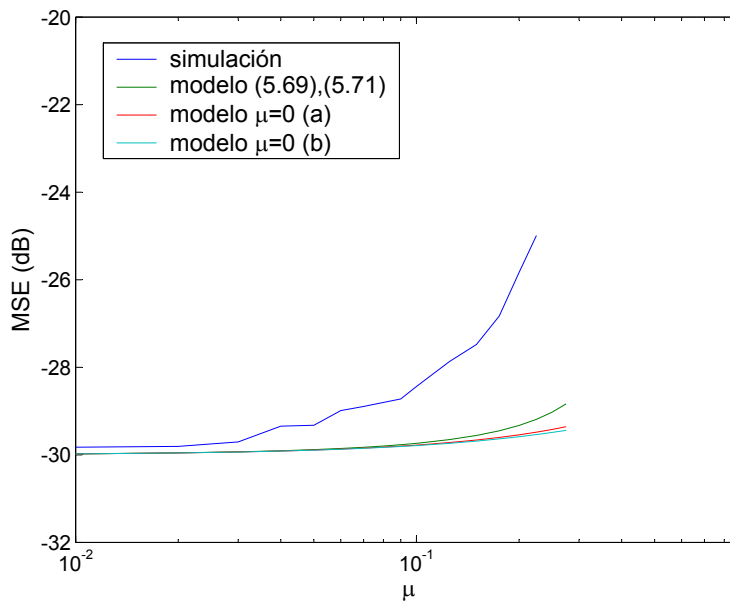


Figura 5.11: Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.

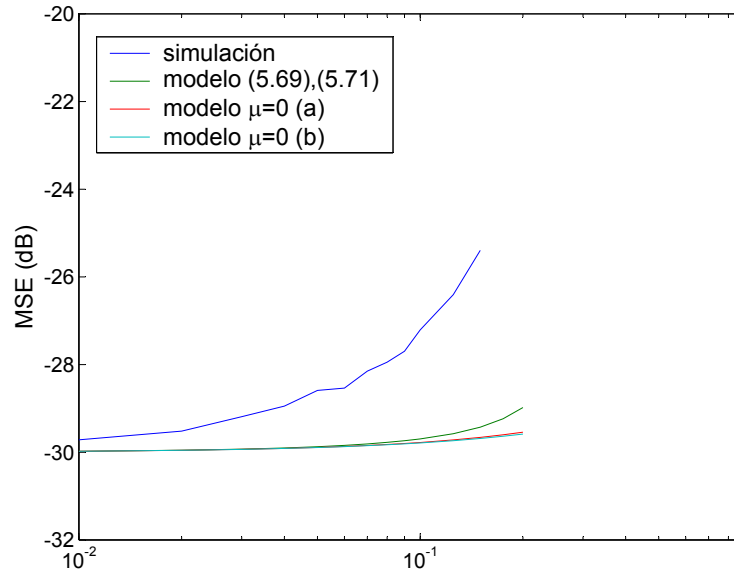


Figura 5.12: Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.

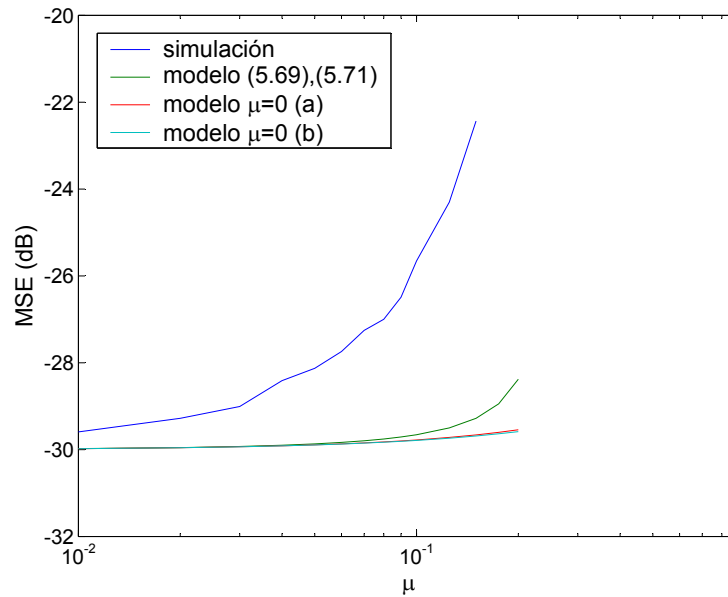


Figura 5.13: Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.

Aunque cuando usamos la estructura de filtrado-x es más eficiente construir el vector de error directamente a partir de los datos de la señal de error, si usásemos la estrategia de actualización eficiente del vector de error con esta estructura, obtendríamos un valor del EMSE ligeramente diferente ya que el vector de error $\mathbf{e}_N[n]$ no estaría formado directamente por los datos de la señal de error, sino que se calcularía según (5.75):

$$\begin{aligned} \mathbf{e}_N[n] &= \begin{pmatrix} e[n] \\ (1-\mu)\bar{\mathbf{e}}_N[n-1] \end{pmatrix} \\ &= \begin{pmatrix} d[n] + \mathbf{v}_L[n]\mathbf{w}_L[n-1] \\ (1-\mu)(d[n-1] + \mathbf{v}_L[n-1]\mathbf{w}_L[n-2]) \\ (1-\mu)(d[n-2] + \mathbf{v}_L[n-2]\mathbf{w}_L[n-3]) \\ \vdots \\ (1-\mu)(d[n-N+1] + \mathbf{v}_L[n-N+1]\mathbf{w}_L[n-N]) \end{pmatrix}. \end{aligned} \quad (5.75)$$

Introduciendo la relación $\mathbf{d}_N[n] = -\mathbf{V}[n]\mathbf{w}_L^0 + \mathbf{r}_N[n]$, podemos reescribir la ecuación anterior como

$$\mathbf{e}_N[n] = \begin{pmatrix} -\mathbf{v}_L[n]\tilde{\mathbf{w}}_L[n-1] + r[n] \\ (1-\mu)(-\mathbf{v}_L[n-1]\tilde{\mathbf{w}}_L[n-2]) + (1-\mu)r[n-1] \\ \vdots \\ (1-\mu)(-\mathbf{v}_L[n-N+1]\tilde{\mathbf{w}}_L[n-N]) + (1-\mu)r[n-N+1] \end{pmatrix}, \quad (5.76)$$

que agrupando adecuadamente los términos, podría expresarse como $\mathbf{e}_N[n] = -\mathbf{e}_N^{\prime\prime}[n] + \mathbf{r}'_N[n]$, donde en este caso

$$\mathbf{e}_N^{\prime\prime}[n] = \begin{pmatrix} \mathbf{v}_L[n]\tilde{\mathbf{w}}_L[n-1] \\ (1-\mu)(\mathbf{v}_L[n-1]\tilde{\mathbf{w}}_L[n-2]) \\ \vdots \\ (1-\mu)(\mathbf{v}_L[n-N+1]\tilde{\mathbf{w}}_L[n-N]) \end{pmatrix} \quad (5.77)$$

y $\mathbf{r}'_N[n]$ vuelve a ser el vector definido en (5.61).

Repitiendo, de nuevo, los pasos (5.165)-(5.167), volveríamos a obtener las mismas expresiones que en (5.69) y (5.63), donde hay que considerar el nuevo vector $\mathbf{e}_N^{\prime\prime}[n]$ definido en (5.77), resultando finalmente la expresión:

$$\begin{aligned} \mu Tr(E\{\mathbf{e}_N^{\prime\prime}[n](\mathbf{e}_N^{\prime\prime}[n])^T\}E\{\Psi_v[n]\}) + \mu Tr(E\{\mathbf{r}'_N[n]\mathbf{r}'_N[n]^T\}E\{\Psi_v[n]\}) \\ = 2Tr(E\{\mathbf{e}_N^{\prime\prime}[n](\mathbf{e}_N^{\prime\prime}[n])^T\}E\{\Phi_v[n]\}). \end{aligned} \quad (5.78)$$

Procediendo de forma análoga a lo realizado en los casos anteriores (como puede consultarse en el punto 5.6.6) se obtiene la expresión:

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 \text{Tr}(E\{D_{42}\Psi_v[n]\})}{2\text{Tr}(E\{D_{41}\Phi_v[n]\}) - \mu\text{Tr}(E\{D_{42}\Psi_v[n]\})} \quad (5.79)$$

siendo D_{41} la siguiente matriz diagonal:

$$\begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & (1-\mu)^2 & 0 & \dots & \vdots \\ \vdots & & (1-\mu)(1-\mu)^2 & & \\ & & & (1-2\mu)(1-\mu)^2 & \\ & & & \ddots & 0 \\ 0 & \dots & & & [1-(N-2)\mu](1-\mu)^2 \end{pmatrix}. \quad (5.80)$$

y D_{42} la matriz diagonal:

$$D_{42} = \begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & (1-\mu)^2 & 0 & \dots & \vdots \\ \vdots & & (1-\mu)^2 & & \\ & & & \ddots & 0 \\ 0 & \dots & & 0 & (1-\mu)^2 \end{pmatrix}. \quad (5.81)$$

De nuevo podremos obtener expresiones aproximadas de (5.79) dependiendo de los valores de δ y μ .

Así, usando un factor de regularización pequeño ($\delta \approx 0$), tendríamos:

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 \text{Tr}(E\{D_{42}\Psi_v[n]\})}{2\text{Tr}(E\{D_{41}\Phi_v[n]\}) - \mu\text{Tr}(E\{D_{42}\Psi_v[n]\})}, \quad (5.82)$$

- Si $\mu \approx 0$, $D_{41} = D_{42} = I$

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2}{2-\mu}.$$
- Si $\mu \approx 1$, $D_{41} = D_{42}$ (Matriz de ceros excepto el primer elemento que es 1)

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2}{2-\mu}.$$

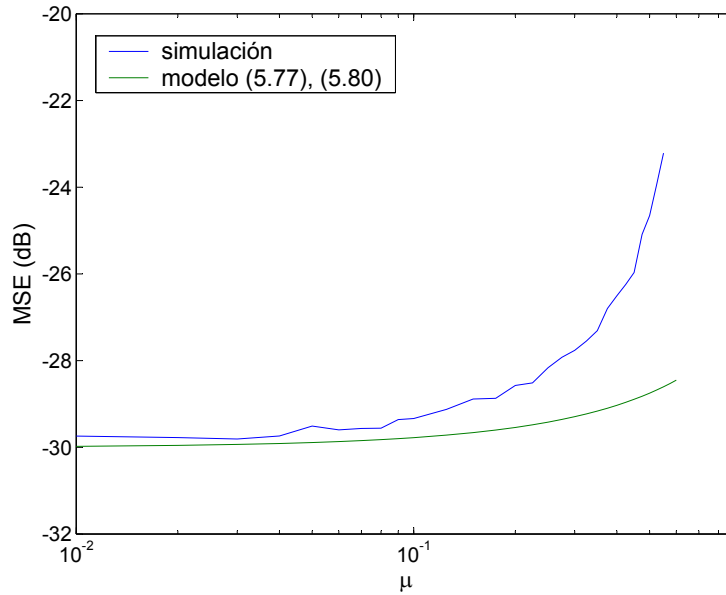


Figura 5.14: Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.

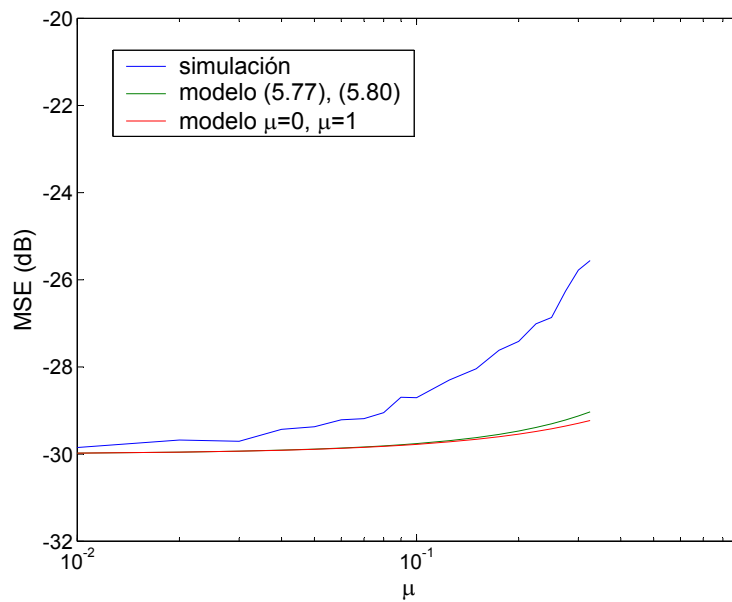


Figura 5.15: Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.

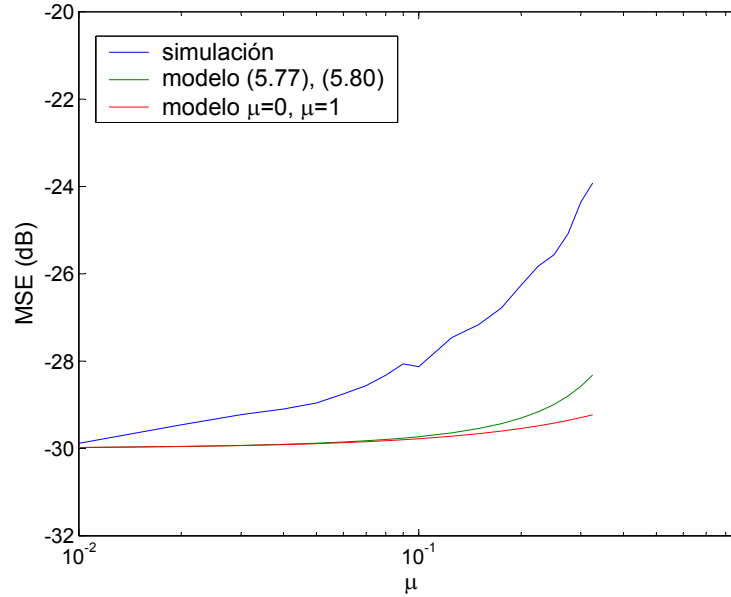


Figura 5.16: Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.

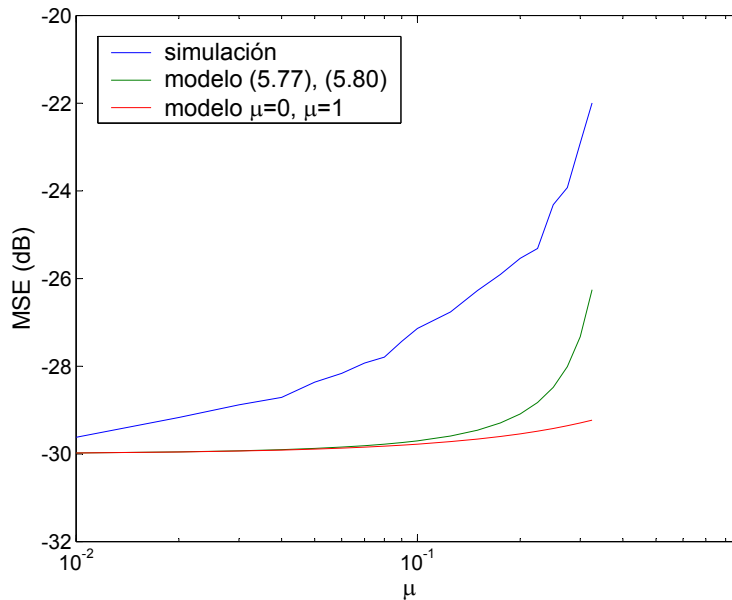


Figura 5.17: Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.

Hay que tener en cuenta, que este último caso no nos proporciona un modelo útil, puesto que para valores de $\mu \approx 1$ el algoritmo no converge.

Las curvas que se obtienen del MSE usando los modelos descritos, y comparándolos con valores procedentes de simulación (considerando las mismas condiciones de simulación que en los casos anteriores) se reflejan en las figuras 5.14-5.17. Se vuelve a mantener el mismo margen dinámico en el eje de ordenadas aun cuando solo existe representación gráfica para aquellos valores de μ que no provocan divergencia en las simulaciones.

De nuevo se observa como todos los modelos ofrecen resultados bastante buenos para valores de la constante de convergencia pequeña. También, los valores estimados por los modelos son más aproximados a los obtenidos mediante simulación cuando se emplean órdenes de proyección pequeños. Incluso se aprecia como para órdenes pequeños, los valores de los modelos coinciden entre sí, acentuándose las diferencias entre ellos conforme el orden de proyección crece.

5.4. Comparación de las diferentes estrategias y modelos

Puesto que hemos obtenido diferentes modelos para la estimación del MSE (o lo que es lo mismo, las prestaciones en estado estacionario de los algoritmos), veamos ahora, comparativamente, cómo sería el comportamiento de cada una de las variantes del algoritmo de proyección afín aplicado al CAR consideradas en los puntos anteriores, atendiendo a los valores proporcionados por los modelos, ya que dichos valores pueden sugerirnos el valor teórico que podría alcanzar el valor cuadrático medio de la señal de error para cada uno de estos algoritmos y variantes, por lo que resulta muy interesante poder estimar dicho valor antes de que el algoritmo se ejecute y poder deducir las diferencias entre los distintos algoritmos referentes a dicho parámetro. Considerando que vamos a trabajar normalmente con valores del parámetro de regularización $\delta \approx 0$, comparamos los modelos derivados de dicha condición, esperando obtener información de cuál es la mejor estrategia en cuanto al MSE de entre las alternativas barajadas.

En primer lugar, destacar que el MSE crece conforme el valor de la constante de convergencia y/o el orden de proyección es mayor. Curiosamente, ambos parámetros controlan la velocidad de convergencia, por lo que se puede deducir que conforme más rápido es el algoritmo para alcanzar el régimen estacionario, peor es el valor del error cuadrático medio alcanzado. Dicha tendencia del MSE, se refleja en la figura 5.18 donde se representan las curvas del

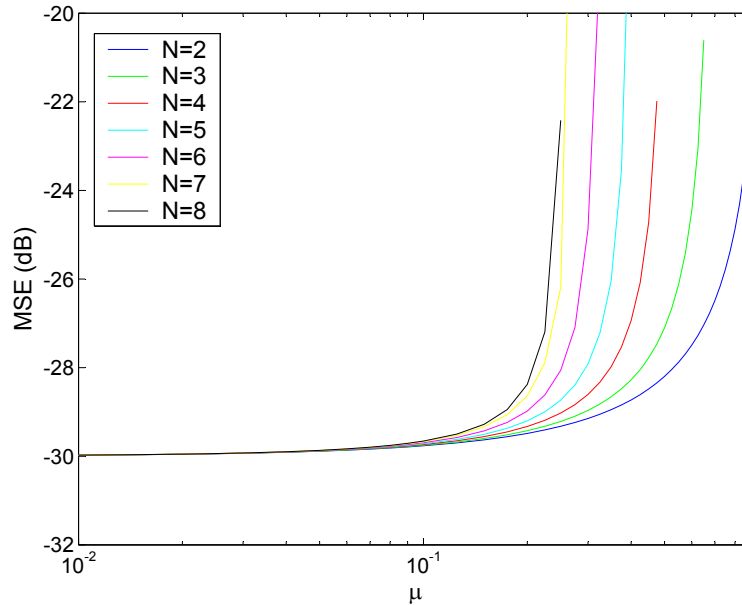


Figura 5.18: Curvas del MSE para diferentes órdenes de proyección, según el modelo (5.73).

modelo que mejor aproxima los resultados del MSE usando la estructura de filtrado-x del algoritmo de proyección afín aplicado al CAR (modelo expresado en (5.73)). Los resultados son extrapolables al resto de variantes del algoritmo y con cualquiera de los modelos, si bien hay modelos independientes del orden de proyección, obtenidos con aproximaciones de régimen de funcionamiento extremo ($\mu \approx 0$ y $\mu \approx 1$).

Debemos notar, que los modelos ofrecen resultados del MSE muy similares e independientes de N cuando el valor de la constante de convergencia usada es pequeño. De hecho, se puede deducir de los modelos obtenidos, que para valores de μ pequeños, todas las variantes de estos algoritmos proporcionan valores similares para el EMSE (o lo que es lo mismo, tienen un funcionamiento en estado estacionario idéntico). En la figura 5.19 se representan las curvas proporcionadas por las diferentes variantes de los algoritmos considerando los modelos obtenidos en régimen de funcionamiento $\mu \approx 0$. Precisamente, para los valores de μ donde esos modelos tienen validez (no se ha producido divergencia), es donde mayor similitud existe entre los modelos, aunque se aprecia que el algoritmo que usa estructura de filtrado-x convencional proporciona valores del MSE ligeramente mejores al resto. Aunque esta diferencia se acrecienta con el valor de μ , debemos notar que los valores proporcionados por

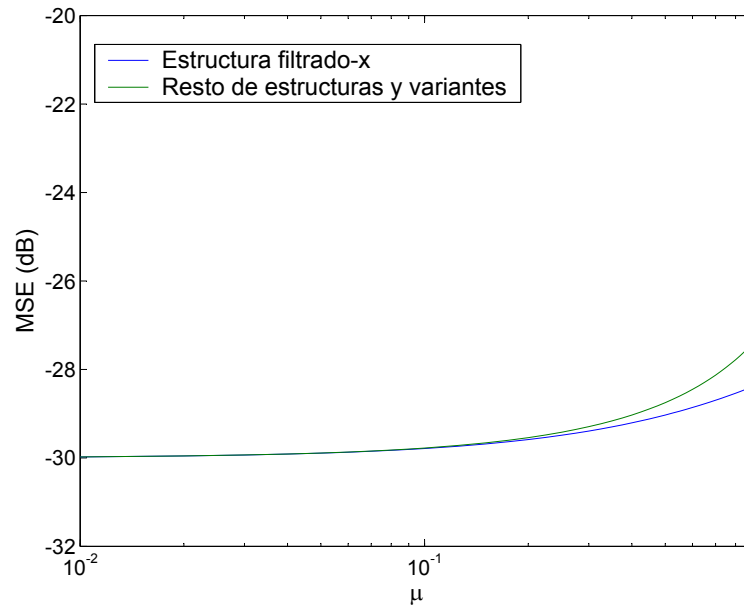
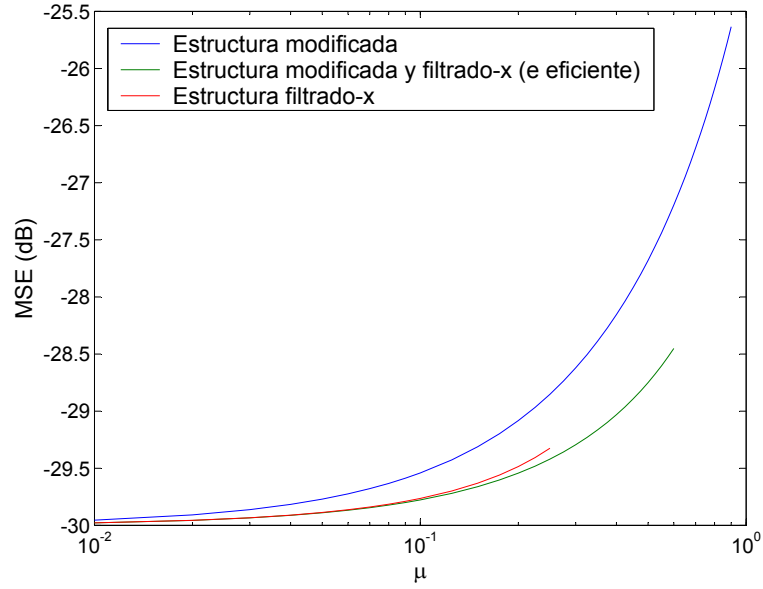
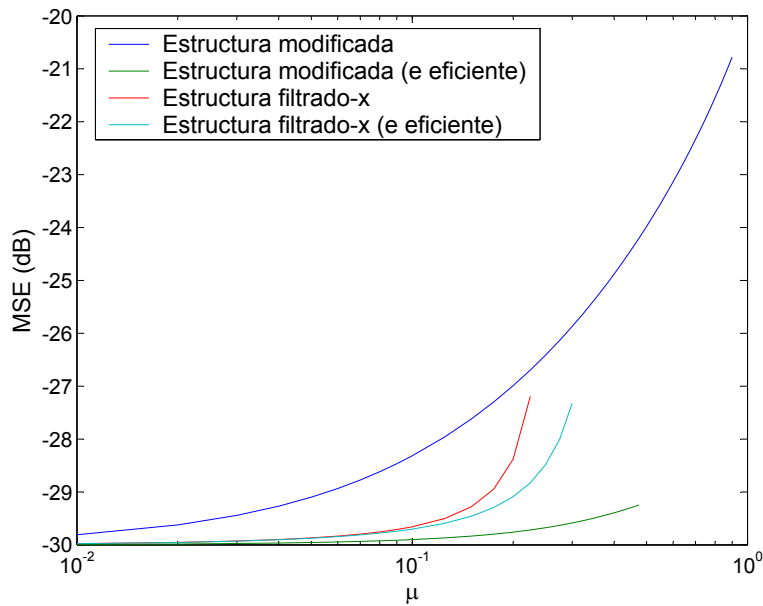


Figura 5.19: Curvas del MSE para diferentes algoritmos considerando modelos con la aproximación $\mu \approx 0$

este modelo en dichos márgenes no tienen relevancia puesto que el algoritmo en realidad diverge y dichos valores de μ no pueden usarse en sistemas reales. Las curvas que se obtienen, en este caso, son independientes del orden de proyección.

Figura 5.20: Curvas del MSE para diferentes algoritmos considerando $N = 2$ Figura 5.21: Curvas del MSE para diferentes algoritmos considerando $N = 8$

Para estudiar lo que pasa en otros márgenes de funcionamiento del parámetro de convergencia, usamos los modelos que mejor aproximan los resultados de simulación. Así, usamos el modelo (5.57) para la estructura modificada con la aproximación $\mu \approx 1$, y los modelos (5.66), (5.73) y (5.82) para la estructura modificada con cálculo eficiente del vector de error, filtrado-x convencional y filtrado-x con cálculo eficiente del vector de error, respectivamente. Los valores del MSE obtenidos se representan en las figuras 5.20 y 5.21 (solo para los valores de μ en los que no se produce divergencia en cada algoritmo). En este caso, sí se aprecia dependencia de las curvas con el orden de proyección. Destacar que para órdenes de proyección pequeños (como se aprecia en la figura 5.20) los modelos de las variantes que usan el cálculo eficiente del error proporcionan valores idénticos del MSE. Quizás, lo más relevante de estas comparativas es que la estructura modificada ofrece peores prestaciones que el resto. Si nos centramos en la estructura de filtrado-x clásica, podemos observar que mientras estamos en márgenes de la constante de convergencia donde el algoritmo no diverge, el modelo ofrecer mejores resultados que el de la modificada. Sin embargo, los valores para el MSE obtenidos por los modelos de las variantes que calculan de forma eficiente el vector de error mejoran los del modelo para la estructura de filtrado-x clásica. Hay que tener en cuenta que la aproximación en la que se basaba dicha estrategia eficiente era precisamente que los valores de $\mu \approx 1$, justo en los márgenes en los que se aprecia la mejoría de estos modelos y en los que, por la aproximación empleada y por la filosofía de la estrategia del cálculo eficiente del vector de error, el algoritmo de proyección afín se asemeja más a un LMS que al original, lo que provoca que mejoren el MSE, pero empeore la velocidad de convergencia. No obstante, los valores en los que se aprecia la mejoría ofrecida por el modelo que usa el cálculo eficiente del vector de error están en los márgenes del parámetro de convergencia para los que dicho algoritmo no converge.

5.5. Estudio del error cuadrático medio del algoritmo de proyección afín multicanal aplicado al CAR

Veamos ahora cómo podemos extender el resultado monocanal del cálculo del MSE para el algoritmo de proyección afín de orden N hasta un caso genérico de control activo de ruido multicanal con I señales de referencia, J actuadores y K sensores de error. Hay que tener en cuenta que en el caso multicanal deberíamos redefinir las matrices y vectores que intervienen en las ecuaciones que definen el algoritmo, así como las ecuaciones de actualización de los coeficientes de los filtros adaptativos, de las que tendremos, en un caso multicanal genérico, $I \times J$ (una para cada filtro adaptativo de tamaño L , cuyo vector de coeficientes es $\mathbf{w}_{ij}[n]$). Aunque existen múltiples estrategias para la actualización de los coeficientes (véase [83] y [84]), nosotros usaremos la siguiente:

$$\mathbf{w}_{ij}[n] = \mathbf{w}_{ij}[n-1] - \mu \sum_{k=1}^K \mathbf{V}_{ijk}[n] (\mathbf{V}_{ijk}^T[n] \mathbf{V}_{ijk}[n] + \delta \mathbf{I})^{-1} \mathbf{e}_k[n], \quad (5.83)$$

siendo $\mathbf{V}_{ijk}[n]$ una matriz de tamaño $L \times N$ con los datos de la señal de referencia i -ésima filtrados por la estimación del camino acústico que enlaza el actuador j -ésimo con el sensor k -ésimo, ordenados tal y como se describe a continuación:

$$\begin{aligned} \mathbf{V}_{ijk}[n] &= \begin{pmatrix} v_{ijk}[n] & \dots & v_{ijk}[n-N+1] \\ v_{ijk}[n-1] & \dots & v_{ijk}[n-N] \\ & \vdots & \\ v_{ijk}[n-L+1] & \dots & v_{ijk}[n-N-L+2] \end{pmatrix} \\ &= \left(\mathbf{v}_{ijk}[n] \quad \dots \quad \mathbf{v}_{ijk}[n-N+1] \right), \end{aligned} \quad (5.84)$$

con $\mathbf{v}_{ijk}[n]$, un vector columna conteniendo los últimos L elementos de la señal de referencia i -ésima, filtrada por la estimación del camino acústico que enlaza el actuador j -ésimo con el sensor k -ésimo y $\mathbf{e}_k[n]$ un vector columna que contiene N muestras de la señal de error calculada a partir de la señal deseada en el sensor k -ésimo ($\mathbf{d}_k[n]$, que almacena las N últimas muestras de dicha señal deseada en el sensor k -ésimo) de la siguiente forma:

$$\mathbf{e}_k[n] = \mathbf{d}_k[n] + \sum_{i=1}^I \sum_{j=1}^J \mathbf{V}_{ijk}[n]^T \mathbf{w}_{ij}[n-1]. \quad (5.85)$$

Conviene reescribir las ecuaciones anteriores de forma matricial, para buscar similitudes con el ya estudiado caso monocanal. Así, redefiniremos los vectores $\mathbf{w}[n]$ y $\mathbf{e}[n]$, empleados en el caso monocanal, pero ahora de tamaño $IJJ \times 1$ y $KN \times 1$, y conteniendo los datos de todos los filtros y todos los vectores de la señal de error, agrupados del siguiente modo:

$$\mathbf{w}[n] = (\mathbf{w}_{11}^T[n] \ \mathbf{w}_{21}^T[n] \ \dots \ \mathbf{w}_{I1}^T[n] \ \dots \ \mathbf{w}_{12}^T[n] \ \dots \ \mathbf{w}_{IJ}^T[n])^T \quad (5.86)$$

$$\mathbf{e}[n] = (\mathbf{e}_1^T[n] \ \mathbf{e}_2^T[n] \ \dots \ \mathbf{e}_K^T[n])^T. \quad (5.87)$$

De esta forma, la ecuación (5.83) se puede escribir como

$$\mathbf{w}[n] = \mathbf{w}[n-1] - \mu \mathbf{U}[n] \mathbf{e}[n], \quad (5.88)$$

donde μ es el parámetro de convergencia y $\mathbf{U}[n]$ es una matriz de tamaño $IJJ \times KN$ formada de la siguiente manera:

$$\mathbf{U}[n] = \begin{pmatrix} \mathbf{V}_{111}[n](\mathbf{V}_{111}^T[n]\mathbf{V}_{111}[n] + \delta\mathbf{I})^{-1} & \dots & \mathbf{V}_{11K}[n](\mathbf{V}_{11K}^T[n]\mathbf{V}_{11K}[n] + \delta\mathbf{I})^{-1} & \\ & \vdots & & \\ \mathbf{V}_{I11}[n](\mathbf{V}_{I11}^T[n]\mathbf{V}_{I11}[n] + \delta\mathbf{I})^{-1} & \dots & \mathbf{V}_{I1K}[n](\mathbf{V}_{I1K}^T[n]\mathbf{V}_{I1K}[n] + \delta\mathbf{I})^{-1} & \\ & \vdots & & \\ \mathbf{V}_{1J1}[n](\mathbf{V}_{1J1}^T[n]\mathbf{V}_{1J1}[n] + \delta\mathbf{I})^{-1} & \dots & \mathbf{V}_{1JK}[n](\mathbf{V}_{1JK}^T[n]\mathbf{V}_{1JK}[n] + \delta\mathbf{I})^{-1} & \\ & \vdots & & \\ \mathbf{V}_{IJ1}[n](\mathbf{V}_{IJ1}^T[n]\mathbf{V}_{IJ1}[n] + \delta\mathbf{I})^{-1} & \dots & \mathbf{V}_{IJK}[n](\mathbf{V}_{IJK}^T[n]\mathbf{V}_{IJK}[n] + \delta\mathbf{I})^{-1} & \end{pmatrix}, \quad (5.89)$$

siendo δ el factor de regularización y la matriz \mathbf{I} la identidad de tamaño $N \times N$.

Llegado a este punto, podemos extender la ecuación (5.85) de la forma:

$$\mathbf{e}[n] = \mathbf{d}[n] + \mathbf{V}^T[n] \mathbf{w}[n-1], \quad (5.90)$$

donde ahora aparecen el vector $\mathbf{d}[n]$ de tamaño IJJ , y la matriz $\mathbf{V}[n]$ de tamaño $IJJ \times KN$, que se definen como:

$$\mathbf{d}[n] = (\mathbf{d}_1^T[n] \ \mathbf{d}_2^T[n] \ \dots \ \mathbf{d}_K^T[n])^T, \quad (5.91)$$

$$\mathbf{V}[n] = \begin{pmatrix} \mathbf{V}_{111}[n] & \dots & \mathbf{V}_{11K}[n] \\ & \vdots & \\ \mathbf{V}_{I11}[n] & \dots & \mathbf{V}_{I1K}[n] \\ & \vdots & \\ \mathbf{V}_{1J1}[n] & \dots & \mathbf{V}_{1JK}[n] \\ & \vdots & \\ \mathbf{V}_{IJ1}[n] & \dots & \mathbf{V}_{IJK}[n] \end{pmatrix}. \quad (5.92)$$

Por último, definiremos la matriz $\mathbf{V}_k[n]$, de forma análoga a (5.92), pero donde en vez de tener todos los datos de todas las señales de referencia filtradas por todas las estimaciones de los caminos secundarios, solo tenemos las aportaciones de dichas señales de referencia filtradas por los caminos secundarios que enlazan cada actuador con el sensor k -ésimo. Así, por ejemplo, $\mathbf{V}_1[n]$ estaría compuesta por las primeras N columnas de $\mathbf{V}[n]$ replicadas K veces. Tendremos por tanto K matrices definidas de esta forma que nos servirán para desacoplar el sistema multicanal, aislando las aportaciones de cada sensor.

$$\mathbf{V}_k[n] = \begin{pmatrix} \mathbf{V}_{11k}[n] & \dots & \mathbf{V}_{11k}[n] \\ & \vdots & \\ \mathbf{V}_{I1k}[n] & \dots & \mathbf{V}_{I1k}[n] \\ & \vdots & \\ \mathbf{V}_{1Jk}[n] & \dots & \mathbf{V}_{1Jk}[n] \\ & \vdots & \\ \mathbf{V}_{IJk}[n] & \dots & \mathbf{V}_{IJk}[n] \end{pmatrix}. \quad (5.93)$$

Para un sistema multicanal, el error cuadrático medio (MSE) se define como:

$$\begin{aligned} \lim_{n \rightarrow \infty} \sum_{k=1}^K E\{|e_k[n]|^2\} &= \lim_{n \rightarrow \infty} \sum_{k=1}^K E\{|d_k[n] + \sum_{i=1}^I \sum_{j=1}^J \mathbf{v}_{ijk}[n]^T \mathbf{w}_{ij}[n-1]|^2\} \\ &= \lim_{n \rightarrow \infty} \sum_{k=1}^K E\{|d_k[n] + \mathbf{v}_k[n]^T \mathbf{w}[n-1]|^2\}, \end{aligned} \quad (5.94)$$

siendo $e_k[n]$ y $d_k[n]$ las señales de error y referencia en el sensor k -ésimo, en el instante n -ésimo, y $\mathbf{v}_k[n]$ el vector de datos de tamaño $IJK \times 1$ definido como

$$\mathbf{v}_k[n] = [\mathbf{v}_{11k}^T[n] \dots \mathbf{v}_{I1k}^T[n] \dots \mathbf{v}_{1Jk}^T[n] \dots \mathbf{v}_{IJk}^T[n]]^T, \quad (5.95)$$

que coincidiría con la primera columna de $\mathbf{V}_k[n]$ en (5.93).

La ecuación expresada en (5.85), también se puede reescribir matricialmente como:

$$\mathbf{e}_k[n] = \mathbf{d}_k[n] + \mathbf{V}_k^T[n]\mathbf{w}[n-1], \quad (5.96)$$

siendo $\mathbf{V}_k[n]$, la matriz formada por las primeras N columnas de $\mathbf{V}_k[n]$.

Si llamamos $\text{MSE}_k = \lim_{n \rightarrow \infty} E\{|e_k[n]|^2\}$, la ecuación (5.94) podría compactarse de la forma:

$$\text{MSE} = \sum_{k=1}^K \text{MSE}_k. \quad (5.97)$$

Para obtener el valor de cada MSE_k vamos a intentar calcular el error a priori obtenido en cada sensor de error, que nos puede proporcionar un valor del EMSE según (5.39), y que esté relacionado de una forma sencilla con el MSE_k correspondiente. Veamos, por tanto, cómo estimar el EMSE para los algoritmos de proyección afín multicanal usando tanto la estructura de filtrado-x modificada como la de filtrado-x convencional.

5.5.1. Estructura modificada multicanal

Con la estructura modificada, el vector de error para el caso multicanal vendría definido por (5.90). Además, suponemos que existe un vector \mathbf{w}^o de tamaño $IJJ \times 1$ que contiene los coeficientes de los filtros que generarían la señal deseada más una desviación, como se consideró en el caso monocanal, de la que ahora tendríamos N muestras en cada sensor de error. Por tanto en este caso, $\mathbf{r}[n]$ representa al vector de tamaño $KN \times 1$ formado por los K vectores de tamaño N que contienen las muestras de las desviaciones en cada sensor. Llamamos σ_r^2 a la varianza de la señal $r(n)$ (señal aleatoria de media nula y estadísticamente independiente de la que tendremos en el vector $\mathbf{r}[n]$ KN muestras). Haciendo uso de estas definiciones, se cumple que:

$$\mathbf{d}[n] = -\mathbf{V}^T[n]\mathbf{w}^o + \mathbf{r}[n]. \quad (5.98)$$

De la misma forma que en el caso monocanal, se define:

$$\tilde{\mathbf{w}}[n] = \mathbf{w}^o - \mathbf{w}[n], \quad (5.99)$$

expresión a partir de la cual se pueden redefinir los nuevos vectores de error a priori y posteriori en cada sensor de error:

$$\mathbf{e}_k^a[n] = \mathbf{V}_k^T[n]\tilde{\mathbf{w}}[n-1] \quad (5.100)$$

y

$$\mathbf{e}_k^p[n] = \mathbf{V}_k^T[n] \tilde{\mathbf{w}}[n]. \quad (5.101)$$

Tenemos que notar que el primer elemento de (5.100) (que coincidiría con los elementos en las posiciones $N + 1, 2N + 1, \dots, (K - 1)N + 1$) nos proporcionará información del EMSE en cada sensor de error, que se define como:

$$\text{EMSE}_k = \lim_{n \rightarrow \infty} E\{|e_k^a[n]|^2\}. \quad (5.102)$$

Al igual que en el caso monocal se cumple que la relación existente entre MSE_k y EMSE_k es:

$$\text{MSE}_k = \text{EMSE}_k + \sigma_r^2. \quad (5.103)$$

Partiendo de (5.88), es inmediato obtener la expresión:

$$\tilde{\mathbf{w}}[n] = \tilde{\mathbf{w}}[n - 1] + \mu \mathbf{U}[n] \mathbf{e}[n]. \quad (5.104)$$

A partir de ahora se procederá de forma similar al caso monocal, con la diferencia de que en vez de calcular las energías, trabajaremos con las energías ponderadas, que se definen a partir de la norma euclídea ponderada, que dada una matriz Σ (simétrica de dimensiones adecuadas), viene descrita según:

$$\|\mathbf{w}\|_{\Sigma}^2 = \mathbf{w}^T \Sigma \mathbf{w}. \quad (5.105)$$

Si definimos unos nuevos vectores de error a priori y posteriori ponderados (de tamaño KN) como:

$$\mathbf{e}_{\Sigma}^a[n] = \mathbf{U}^T[n] \Sigma \tilde{\mathbf{w}}[n - 1] \quad (5.106)$$

$$\mathbf{e}_{\Sigma}^p[n] = \mathbf{U}^T[n] \Sigma \tilde{\mathbf{w}}[n], \quad (5.107)$$

al multiplicar por la izquierda la expresión (5.104) por $\mathbf{U}^T[n] \Sigma$, se obtiene:

$$\mathbf{e}_{\Sigma}^p[n] = \mathbf{e}_{\Sigma}^a[n] + \mu \mathbf{U}^T[n] \Sigma \mathbf{U}[n] \mathbf{e}[n], \quad (5.108)$$

de donde podemos despejar $\mathbf{e}[n]$ e introducirlo en la expresión (5.104), alcanzando la siguiente igualdad:

$$\tilde{\mathbf{w}}[n] = \tilde{\mathbf{w}}[n - 1] + \mathbf{U}[n] (\mathbf{U}^T[n] \Sigma \mathbf{U}[n])^{-1} (\mathbf{e}_{\Sigma}^p[n] - \mathbf{e}_{\Sigma}^a[n]) \quad (5.109)$$

que reordenada adecuadamente, puede escribirse como:

$$\tilde{\mathbf{w}}[n] + \mathbf{U}[n] (\mathbf{U}^T[n] \Sigma \mathbf{U}[n])^{-1} \mathbf{e}_{\Sigma}^a[n] = \tilde{\mathbf{w}}[n - 1] + \mathbf{U}[n] (\mathbf{U}^T[n] \Sigma \mathbf{U}[n])^{-1} \mathbf{e}_{\Sigma}^p[n]. \quad (5.110)$$

Calculando la norma ponderada de ambos lados de la igualdad, tal y como se detalla en (5.6.7), se obtiene:

$$\begin{aligned} \|\tilde{\mathbf{w}}[n]\|_{\Sigma}^2 + (\mathbf{e}_{\Sigma}^a[n])^T (\mathbf{U}^T[n] \Sigma \mathbf{U}[n])^{-1} \mathbf{e}_{\Sigma}^a[n] \\ = \|\tilde{\mathbf{w}}[n-1]\|_{\Sigma}^2 + (\mathbf{e}_{\Sigma}^p[n])^T (\mathbf{U}^T[n] \Sigma \mathbf{U}[n])^{-1} \mathbf{e}_{\Sigma}^p[n]. \end{aligned} \quad (5.111)$$

En estado estacionario, ($n \rightarrow \infty$), $\|\tilde{\mathbf{w}}[n]\|_{\Sigma}^2 = \|\tilde{\mathbf{w}}[n-1]\|_{\Sigma}^2$ y sustituyendo $\mathbf{e}_{\Sigma}^p[n]$ en función de $\mathbf{e}_{\Sigma}^a[n]$ según (5.108), obtenemos:

$$-\mu \mathbf{e}^T[n] (\mathbf{U}^T[n] \Sigma \mathbf{U}[n]) \mathbf{e}[n] = (\mathbf{e}_{\Sigma}^a[n])^T \mathbf{e}[n] + \mathbf{e}^T[n] \mathbf{e}_{\Sigma}^a[n]. \quad (5.112)$$

La ecuación obtenida en (5.112) es similar a la hallada en el caso monocanal, con la dificultad de que tanto el vector $\mathbf{e}[n]$ como $\mathbf{e}_{\Sigma}^a[n]$ contienen información procedente de todos los sensores de error. Sería conveniente intentar desacoplar dicha información para proceder de forma análoga al caso monocanal. Así, cuando la matriz Σ se corresponde con:

$$\Sigma = \Sigma_k[n] = \mathbf{U}[n] (\mathbf{U}^T[n] \mathbf{U}[n])^{-1} \mathbf{V}_k^T[n], \quad (5.113)$$

el vector $\mathbf{e}_{\Sigma}^a[n]$ coincide con $\mathbf{e}_k^a[n]$, que solo contiene información del error a priori en el sensor k -ésimo. Hay que comentar que la matriz $\Sigma_k[n]$ no sería una matriz ni simétrica ni invertible, por lo que en principio no responde a la clase de matrices para la cual se ha descrito el desarrollo anterior. Sin embargo, aunque para esta matriz algunas de las relaciones anteriores no estarían justificadas, vamos a introducirla a partir de ahora considerando que la relación dada en (5.112) es una aproximación de la que se obtendría si usásemos $\Sigma_k[n]$ desde el principio (cuya expresión sería complicada de alcanzar de una forma tan compacta).

En cuanto a $\mathbf{e}[n]$, tenemos que considerar que se cumple que:

$$\mathbf{e}[n] = -\mathbf{e}^a[n] + \mathbf{r}[n] \quad (5.114)$$

siendo $\mathbf{e}^a[n] = \mathbf{V}^T[n] \tilde{\mathbf{w}}[n-1]$.

El valor de $\mathbf{e}[n]$ obtenido en (5.114), puede introducirse en la ecuación (5.112), pero antes, convendría desacoplar el vector $\mathbf{e}^a[n]$ de forma que solo tenga información del error a priori en el sensor k -ésimo. Así,

$$\begin{aligned} \mathbf{e}^a[n] &= \mathbf{V}^T[n] \tilde{\mathbf{w}}[n-1] \\ &= \{\mathbf{V}_k^T[n] (\mathbf{V}[n] \mathbf{V}^T[n])^{-1} \mathbf{V}[n]\}^{-1} \{\mathbf{V}_k^T[n] (\mathbf{V}[n] \mathbf{V}^T[n])^{-1} \mathbf{V}[n]\} \mathbf{V}^T[n] \tilde{\mathbf{w}}[n-1] \\ &= \{\mathbf{V}_k^T[n] (\mathbf{V}[n] \mathbf{V}^T[n])^{-1} \mathbf{V}[n]\}^{-1} \mathbf{e}_k^a[n] = \mathbf{M}_k^{-1}[n] \mathbf{e}_k^a[n]. \end{aligned} \quad (5.115)$$

Desgraciadamente, en la mayoría de los casos multicanal (cuando el número de sensores $K > 1$) la matriz $\mathbf{M}_k[n]$ no es invertible, por lo que la solución

alcanzada anteriormente no es implementable. En tales casos, para minimizar el inconveniente de la singularidad de $\mathbf{M}_k[n]$ realizamos la aproximación $\mathbf{e}^a[n] = \mathbf{e}_k^a[n]$ para poder así invertir dicha matriz (sería equivalente a considerar la matriz $\mathbf{M}_k[n]$ igual a la identidad). Aunque esta aproximación puede parecer poco rigurosa, hay que considerar que en la mayoría de los sistemas multicanal de CAR donde realizamos control local en torno a una zona espacial no muy extensa, los caminos acústicos secundarios son parecidos entre sí, por lo que las señales de referencia filtradas por las diferentes estimaciones de dichos caminos acústicos, también lo serán. Por tanto, las señales de error en los diferentes sensores no diferirán demasiado entre ellas y queda justificada dicha aproximación. No obstante, si no realizamos esta aproximación, resultaría imposible obtener las señales genéricas ($\mathbf{e}^a[n]$ o $\mathbf{e}[n]$) con la información de todos los sensores, a partir de las señales particulares en cada sensor de error ($\mathbf{e}_k^a[n]$ o $\mathbf{e}_{kk}[n]$ (vector que sería el definido en (5.85), replicado K veces)). La operación contraria sí sería posible, puesto que se cumple que:

$$\mathbf{e}_{kk}[n] = -\mathbf{e}_k^a[n] + \mathbf{r}_k[n], \quad (5.116)$$

siendo

$$\mathbf{e}_{kk}[n] = M_k \mathbf{e}[n], \quad \mathbf{e}_k^a[n] = M_k \mathbf{e}^a[n], \quad \mathbf{r}_k[n] = M_k \mathbf{r}[n]. \quad (5.117)$$

La matriz M_k (de tamaño $NK \times NK$), que aparece en (5.115) y (5.117) puede calcularse también como la matriz formada por la matriz identidad de tamaño $N \times N$ replicada K veces en la dimensión de las columnas, y colocada en las N filas que ocupan las posiciones de la $1 + N(k-1)$ hasta la Nk . Por ejemplo, para una sistema con $K = 2$ y $N = 3$, tendríamos las matrices:

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.118)$$

Para darle generalidad a la nomenclatura, seguiremos trabajando con la matriz $\mathbf{M}_k[n]$ según su definición original que se desprende de (5.115), teniendo en cuenta que cuando hubiera que invertirla en los casos en los que $K > 1$, usaríamos la matriz identidad.

Continuando con (5.112), y definiendo la matriz $\mathbf{N}_k[n] = \mathbf{U}^T[n]\Sigma_k[n]\mathbf{U}[n]$, llegamos a la expresión:

$$\begin{aligned} & -\mu\{\mathbf{r}_k^T[n](\mathbf{M}_k^T[n])^{-1} - (\mathbf{e}_k^a[n])^T(\mathbf{M}_k^T[n])^{-1}\}\mathbf{N}_k[n]\{-\mathbf{M}_k^{-1}[n]\mathbf{e}_k^a[n] + \mathbf{M}_k^{-1}[n]\mathbf{r}_k[n]\} \\ & = (\mathbf{e}_k^a[n])^T(\mathbf{M}_k^{-1}[n]\mathbf{r}_k[n] - \mathbf{M}_k^{-1}[n]\mathbf{e}_k^a[n]) \\ & + \{\mathbf{r}_k^T[n](\mathbf{M}_k^T[n])^{-1} - (\mathbf{e}_k^a[n])^T(\mathbf{M}_k^T[n])^{-1}\}\mathbf{e}_k^a[n]. \end{aligned} \quad (5.119)$$

Tomando valores medios en (5.119), y considerando independencia estadística entre los datos de $\mathbf{e}_k^a[n]$ y $\mathbf{r}_k[n]$, obtenemos:

$$\begin{aligned} & -\mu E\{\mathbf{r}_k^T[n](\mathbf{M}_k^T[n])^{-1}\mathbf{N}_k[n]\mathbf{M}_k^{-1}[n]\mathbf{r}_k[n]\} \\ & -\mu E\{(\mathbf{e}_k^a[n])^T(\mathbf{M}_k^T[n])^{-1}\mathbf{N}_k[n]\mathbf{M}_k^{-1}[n]\mathbf{e}_k^a[n]\} \\ & = -E\{(\mathbf{e}_k^a[n])^T(\mathbf{M}_k[n]^{-1})^T\mathbf{e}_k^a[n]\} - E\{(\mathbf{e}_k^a[n])^T\mathbf{M}_k^{-1}[n]\mathbf{e}_k^a[n]\}. \end{aligned} \quad (5.120)$$

Llamando $\mathbf{O}_k[n] = (\mathbf{M}_k^T[n])^{-1}\mathbf{N}_k[n]\mathbf{M}_k^{-1}[n]$ y como ya se dedujo en el caso monocanal, la expresión (5.120) puede reescribirse como:

$$\begin{aligned} & \mu Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\mathbf{O}_k[n]\}) + \mu Tr(E\{\mathbf{r}_k[n]\mathbf{r}_k^T[n]\mathbf{O}_k[n]\}) \\ & = Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T(\mathbf{M}_k^T[n])^{-1}\}) + Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\mathbf{M}_k^{-1}[n]\}), \end{aligned} \quad (5.121)$$

que considerando independencia estadística entre los datos de las matrices $\mathbf{O}_k[n]$ y $\mathbf{M}_k^{-1}[n]$ con los de $\mathbf{e}_k^a[n]$, y de los datos de $\mathbf{O}_k[n]$ con los de $\mathbf{r}[n]$, se obtiene

$$\begin{aligned} & \mu Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}E\{\mathbf{O}_k[n]\}) + \mu Tr(E\{\mathbf{r}_k[n]\mathbf{r}_k^T[n]\}E\{\mathbf{O}_k[n]\}) \\ & = Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}E\{(\mathbf{M}_k^T[n])^{-1}\}) + Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}E\{\mathbf{M}_k^{-1}[n]\}). \end{aligned} \quad (5.122)$$

Conviene darse cuenta que la expresión alcanzada en (5.122) coincide con la hallada en el caso monocanal en (5.23), puesto que en dicho caso, la matriz $\mathbf{U}[n]$ coincidiría con $\mathbf{V}[n]$, $\mathbf{M}[n] = (\mathbf{V}^T[n]\mathbf{V}[n] + \delta\mathbf{I})$ (matriz simétrica que nos permitiría agrupar los dos términos del miembro derecho de la igualdad) y $\mathbf{N}[n] = \mathbf{V}^T[n]\mathbf{V}[n]$.

Para seguir operando con la ecuación (5.122), al igual que en el caso monocanal, solo vamos a considerar los valores de la diagonal de $E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}$, y $E\{\mathbf{r}_k[n](\mathbf{r}_k[n])^T\}$, despreciando el resto. Por tanto, para calcular la traza de los productos matriciales que aparecen en dicha ecuación, es suficiente con realizar el producto de las diagonales de las matrices que se multiplican dentro del operador traza. Nos interesa, pues, calcular la diagonal de $E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}$, y $E\{\mathbf{r}_k[n](\mathbf{r}_k[n])^T\}$.

Respecto a $E\{\mathbf{r}_k[n](\mathbf{r}_k[n])^T\}$, es sencillo demostrar que se corresponde con el valor $\sigma_{r_k}^2 \mathbf{I}$, siendo \mathbf{I} la matriz identidad de tamaño $KN \times KN$, y $\sigma_{r_k}^2$ la varianza de la señal $r_k[n]$ formada a partir de la señal original $r[n]$ según (5.117), mientras que los valores que forman la diagonal de $E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}$, serían los expresados en (5.123) replicados K veces.

$$\begin{pmatrix} E\{|\mathbf{v}_k^T[n]\tilde{\mathbf{w}}[n-1]|^2\} \\ E\{|\mathbf{v}_k^T[n-1]\tilde{\mathbf{w}}[n-1]|^2\} \\ \vdots \\ E\{|\mathbf{v}_k^T[n-N+1]\tilde{\mathbf{w}}[n-1]|^2\} \end{pmatrix}. \quad (5.123)$$

De (5.123), solo podemos identificar el primer elemento (que coincide con los elementos en las posiciones $N+1, 2N+1, \dots$ de la diagonal de $E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}$), igual a $E\{|e_k^a[n]|^2\}$. Para calcular el resto, se procederá como en el caso monocanal. Hay que tener en cuenta que, con la notación introducida, se cumple:

$$\mathbf{e}^p[n] = \mathbf{e}^a[n] + \mu \mathbf{V}^T[n] \mathbf{U}[n] \mathbf{e}[n], \quad (5.124)$$

con $\mathbf{e}^p[n] = \mathbf{V}^T[n] \tilde{\mathbf{w}}[n]$. Introduciendo (5.114) en (5.124)

$$\mathbf{e}^p[n] = (\mathbf{I} - \mu \mathbf{V}^T[n] \mathbf{U}[n]) \mathbf{e}^a[n] + \mu \mathbf{V}^T[n] \mathbf{U}[n] \mathbf{r}[n]. \quad (5.125)$$

Realizando la multiplicación de $\mathbf{e}^p[n]$ por su traspuesta:

$$\begin{aligned} \mathbf{e}^p[n](\mathbf{e}^p[n])^T &= (\mathbf{I} - \mu \mathbf{V}^T[n] \mathbf{U}[n]) \mathbf{e}^a[n] (\mathbf{e}^a[n])^T (\mathbf{I} - \mu \mathbf{V}^T[n] \mathbf{U}[n])^T \\ &\quad + \mu \mathbf{V}^T[n] \mathbf{U}[n] \mathbf{r}[n] (\mathbf{e}^a[n])^T (\mathbf{I} - \mu \mathbf{V}^T[n] \mathbf{U}[n])^T \\ &\quad + \mu (\mathbf{I} - \mu \mathbf{V}^T[n] \mathbf{U}[n]) \mathbf{e}^a[n] \mathbf{r}^T[n] \mathbf{U}^T[n] \mathbf{V}[n] \\ &\quad + \mu^2 \mathbf{V}^T[n] \mathbf{U}[n] \mathbf{r}[n] \mathbf{r}^T[n] \mathbf{U}^T[n] \mathbf{V}[n]. \end{aligned} \quad (5.126)$$

Y tomando finalmente el valor medio, considerando independencia estadística de las variables $r[n]$ y $e^a[n]$, se obtiene:

$$E\{\mathbf{e}^p[n](\mathbf{e}^p[n])^T\} = (1 - \mu IJ)^2 E\{\mathbf{e}^a[n](\mathbf{e}^a[n])^T\} + \mu^2 (IJ)^2 \sigma_r^2 \mathbf{I}. \quad (5.127)$$

En la expresión anterior, se han despreciado los valores fuera de la diagonal del producto $E\{\mathbf{V}^T[n] \mathbf{U}[n]\}$ (que sería como si se considerara independencia estadística entre los datos de las variables de media nula de las señales de referencia filtradas por las estimaciones de los caminos acústicos que enlazan los actuadores con el sensor l -ésimo y la filtradas por las estimaciones de los caminos acústicos que enlazan los actuadores con el sensor q -ésimo, para $l \neq q$, tal y como se describe en 5.6.8). Si consideramos que existe independencia estadística de $\mathbf{M}_k[n]$ con las variables que aparecen en la ecuación (5.127) (lo

que se cumple si consideramos que dichas matrices son como las definidas en (5.118), y multiplicamos ambos miembros de dicha ecuación por la izquierda por $E\{\mathbf{M}_k[n]\}$ y por la derecha por $E\{\mathbf{M}_k^T[n]\}$, obtenemos la siguiente igualdad:

$$E\{\mathbf{e}_k^p[n](\mathbf{e}_k^p[n])^T\} = (1 - \mu IJ)^2 E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\} + \mu^2 (IJ)^2 \sigma_{rk}^2 \mathbf{I}. \quad (5.128)$$

Los elementos de la diagonal de la relación anterior, serán los que nos ayuden a calcular los elementos restantes de (5.123), siendo los N primeros (el resto se replica K veces):

$$\begin{aligned} & \begin{pmatrix} E\{|\mathbf{v}_k^T[n]\tilde{\mathbf{w}}[n]|^2\} \\ E\{|\mathbf{v}_k^T[n-1]\tilde{\mathbf{w}}[n]|^2\} \\ \vdots \\ E\{|\mathbf{v}_k^T[n-N+1]\tilde{\mathbf{w}}[n]|^2\} \end{pmatrix} = (IJ)^2 \mu^2 \sigma_{rk}^2 \\ & + (1 - IJ\mu)^2 \begin{pmatrix} E\{|\mathbf{v}_k^T[n]\tilde{\mathbf{w}}[n-1]|^2\} \\ E\{|\mathbf{v}_k^T[n-1]\tilde{\mathbf{w}}[n-1]|^2\} \\ \vdots \\ E\{|\mathbf{v}_k^T[n-N+1]\tilde{\mathbf{w}}[n-1]|^2\} \end{pmatrix}. \end{aligned} \quad (5.129)$$

Así, el primer elemento de (5.123), sabemos que se corresponde con $E\{|e_k^a[n]|^2\}$. Para calcular el segundo elemento, operando de igual forma que en el caso monocanal, retrasamos una unidad la expresión de la primera fila de la igualdad expresada en (5.129), y así sucesivamente se obtendría para cada elemento de (5.123):

$$\begin{aligned} E\{|\mathbf{v}_k^T[n]\tilde{\mathbf{w}}[n-1]|^2\} &= E\{|e_k^a[n]|^2\} \\ E\{|\mathbf{v}_k^T[n-1]\tilde{\mathbf{w}}[n-1]|^2\} &= (1 - \mu IJ)^2 E\{|e_k^a[n-1]|^2\} + (\mu IJ)^2 \sigma_{rk}^2 \\ E\{|\mathbf{v}_k^T[n-2]\tilde{\mathbf{w}}[n-1]|^2\} &= (1 - \mu IJ)^4 E\{|e_k^a[n-2]|^2\} + (1 - \mu IJ)^2 (\mu IJ)^2 \sigma_{rk}^2 \\ &\quad + (\mu IJ)^2 \sigma_{rk}^2 \\ &\vdots \end{aligned} \quad (5.130)$$

y considerando que en régimen permanente, $E\{|e_k^a[n]|^2\} = E\{|e_k^a[n-1]|^2\} = \dots = E\{|e_k^a[n-N+1]|^2\}$, dicha diagonal puede escribirse de forma matricial como:

$$E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\} \approx E\{|e_k^a[n]|^2\} D_{M11} + \mu^2 \sigma_{rk}^2 (IJ)^2 D_{M12}, \quad (5.131)$$

con:

$$D_{M11} = \begin{pmatrix} D_{M1} & 0 & \cdots & 0 \\ 0 & D_{M1} & 0 \cdots & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & 0 & D_{M1} \end{pmatrix} \quad (5.132)$$

y

$$D_{M12} = \begin{pmatrix} D_{M2} & 0 & \cdots & 0 \\ 0 & D_{M2} & 0 \cdots & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & 0 & D_{M2} \end{pmatrix} \quad (5.133)$$

siendo D_{M1} y D_{M2} , las matrices de dimensiones $N \times N$ similares a las obtenidas para el caso equivalente monocanal (definidas en (5.33) y (5.34)) y que se replican K veces a lo largo de la diagonal de la matriz correspondiente de dimensiones $KN \times KN$.

$$D_{M1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & (1 - \mu IJ)^2 & 0 & \cdots & \vdots \\ \vdots & 0 & (1 - \mu IJ)^4 & & \\ & & & \ddots & 0 \\ 0 & & & & (1 - \mu IJ)^{2(N-1)} \end{pmatrix} \quad (5.134)$$

y

$$D_{M2} = \begin{pmatrix} 0 & \cdots & & & 0 \\ 0 & 1 & 0 & \cdots & \vdots \\ \vdots & 0 & [1 + (1 - \mu IJ)^2] & 0 & \\ & & & \ddots & 0 \\ 0 & \cdots & & 0 & [1 + (1 - \mu IJ)^2 + \cdots + (1 - \mu IJ)^{2(N-2)}] \end{pmatrix}. \quad (5.135)$$

De esta forma, podemos escribir que:

$$\begin{aligned} & Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}E\{\mathbf{O}_k[n]\}) \\ & \approx E\{|e_k^a[n]|^2\}Tr(D_{M11}E\{\mathbf{O}_k[n]\}) + Tr(D_{M12}E\{\mathbf{O}_k[n]\})(IJ)^2\mu^2\sigma_{rk}^2, \end{aligned} \quad (5.136)$$

$$\begin{aligned} & Tr(E\{\mathbf{e}_k^a[n](\mathbf{e}_k^a[n])^T\}E\{\mathbf{M}_k^{-1}[n]\}) \\ & \approx E\{|e_k^a[n]|^2\}Tr(D_{M11}E\{\mathbf{M}_k^{-1}[n]\}) + Tr(D_{M12}E\{\mathbf{M}_k^{-1}[n]\})(IJ)^2\mu^2\sigma_{rk}^2 \end{aligned} \quad (5.137)$$

y

$$Tr(E\{\mathbf{r}_k[n]\mathbf{r}_k^T[n]\}E\{\mathbf{O}_k\}) \approx \sigma_{rk}^2 Tr(E\{\mathbf{O}_k[n]\}). \quad (5.138)$$

Finalmente, reformulando la expresión de (5.122), obtenemos:

$$\begin{aligned} \mu E\{|e_k^a[n]|^2\}Tr(D_{M11}E\{\mathbf{O}_k[n]\}) + \mu Tr(D_{M12}E\{\mathbf{O}_k[n]\})(IJ)^2\mu^2\sigma_{rk}^2 \\ + \mu\sigma_{rk}^2Tr(E\{\mathbf{O}_k[n]\}) = 2E\{|e_k^a[n]|^2\}Tr(D_{M11}E\{\mathbf{M}_k^{-1}[n]\}) \\ + 2Tr(D_{M12}E\{\mathbf{M}_k^{-1}[n]\})(IJ)^2\mu^2\sigma_{rk}^2 \end{aligned} \quad (5.139)$$

donde se ha considerado que la matriz $\mathbf{M}_k^{-1}[n]$ es simétrica (lo que claramente se cumple cuando la consideramos la matriz identidad).

A partir de (5.139), podemos obtener el valor de $E\{|e_k^a[n]|^2\}$

$$\begin{aligned} E\{|e_k^a[n]|^2\} \\ = \frac{\mu\sigma_{rk}^2Tr(E\{\mathbf{O}_k[n]\}) + \{\mu Tr(D_{M12}E\{\mathbf{O}_k[n]\}) - 2Tr(D_{M12}E\{\mathbf{M}_k^{-1}[n]\})\}(IJ)^2\mu^2\sigma_{rk}^2}{2Tr(D_{M11}E\{\mathbf{M}_k^{-1}[n]\}) - \mu Tr(D_{M11}E\{\mathbf{O}_k[n]\})} \end{aligned} \quad (5.140)$$

que si en (5.136) y (5.137) despreciamos los términos que dependen de σ_{rk}^2 frente a los valores de $e_k^a[n]$, quedaría:

$$E\{|e_k^a[n]|^2\} = \frac{\mu\sigma_{rk}^2Tr(E\{\mathbf{O}_k[n]\})}{2Tr(D_{M11}E\{\mathbf{M}_k^{-1}[n]\}) - \mu Tr(D_{M11}E\{\mathbf{O}_k[n]\})}. \quad (5.141)$$

El límite cuando $n \rightarrow \infty$ de las expresiones anteriores daría una estimación del $EMSE_k$ siendo, finalmente, el valor del MSE multicanal buscado, según (5.97) y (5.103):

$$MSE = \sum_{k=1}^K (EMSE_k + \sigma_r^2). \quad (5.142)$$

Al igual que para los modelos monocanal, trataremos de contrastar los resultados obtenidos por el modelo con simulaciones realizadas para un sistema multicanal (en este caso, con una señal de referencia ($I = 1$), dos actuadores secundarios ($J = 2$) y dos sensores de error ($K = 2$)). La señal de referencia se definirá del mismo modo que en el caso monocanal, mientras que los caminos secundarios (4 en total) serán modelados por filtros FIR de 8 coeficientes, siendo el tamaño de los filtros adaptativos de 16 coeficientes. La varianza de la señal de desviación $r[n]$ introducida en (5.98) se ha fijado en $\sigma_r = 10^{-3}$. Usando el modelo descrito en (5.141) para el cálculo del valor del EMSE en cada sensor, y (5.142) para la obtención de la estimación del error cuadrático medio del algoritmo multicanal según se describe en (5.94), se obtienen las figuras representadas en 5.22-5.25, donde se pueden comparar los valores ofrecidos por el modelo y los obtenidos mediante simulación para el MSE, en función del parámetro de convergencia (μ) y para diferentes órdenes de proyección.

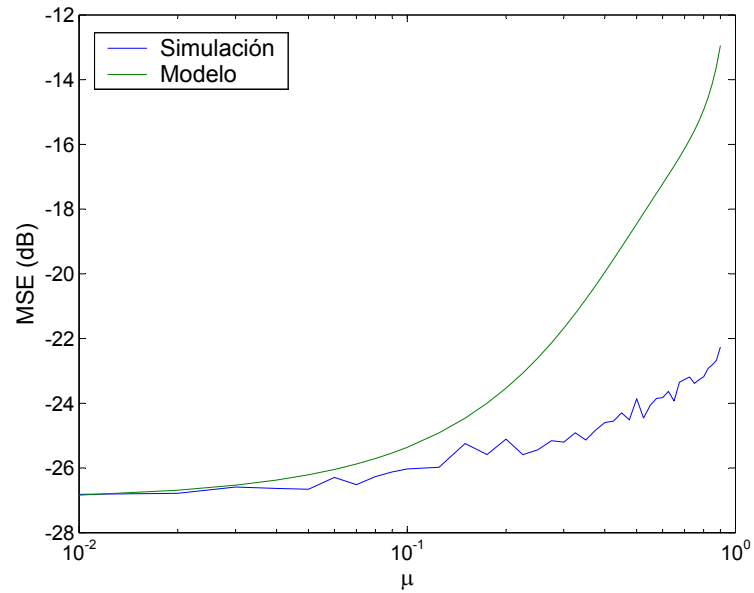


Figura 5.22: Comparación entre el MSE multicanal para $N = 2$ obtenido mediante modelo y simulación.

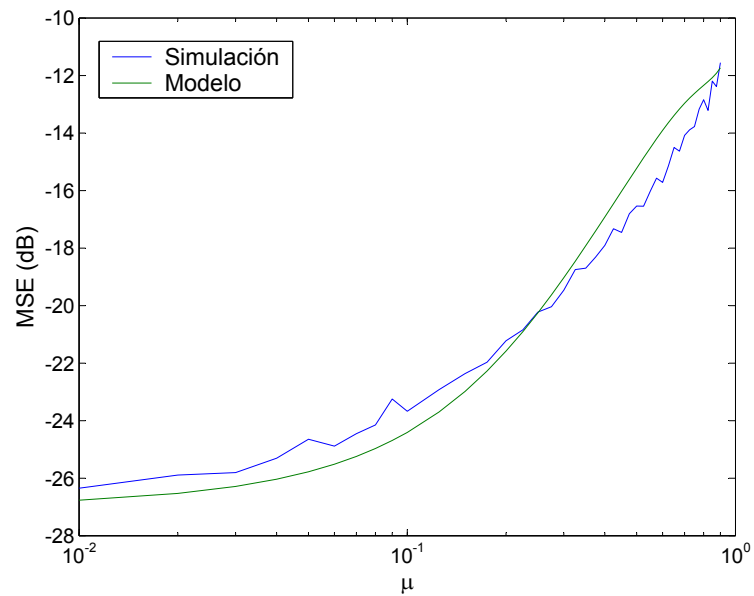


Figura 5.23: Comparación entre el MSE multicanal para $N = 3$ obtenido mediante modelo y simulación.

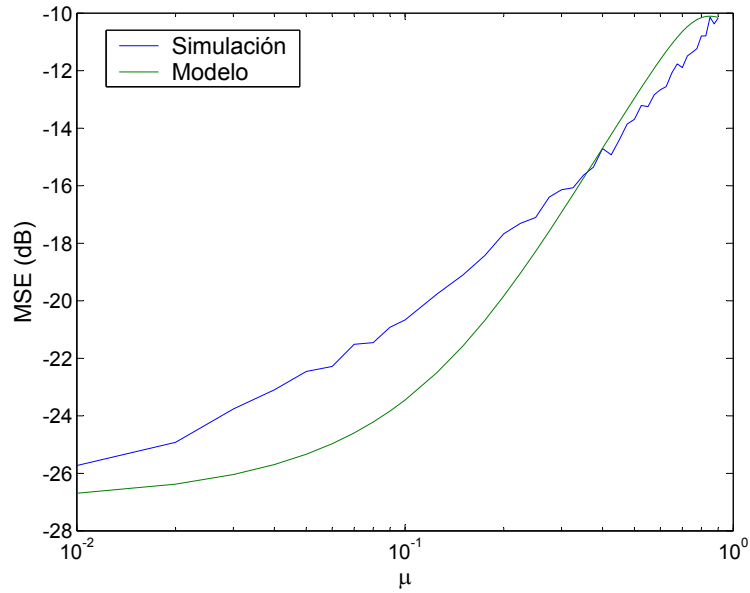


Figura 5.24: Comparación entre el MSE multicanal para $N = 4$ obtenido mediante modelo y simulación.

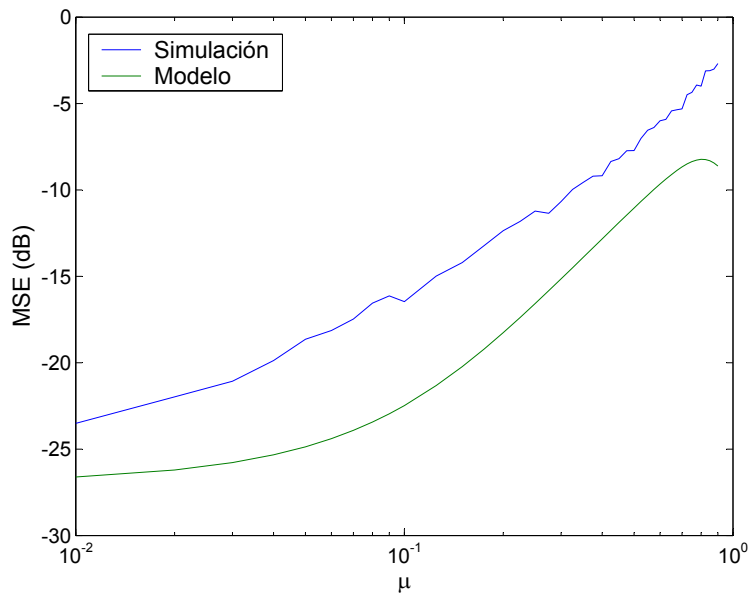


Figura 5.25: Comparación entre el MSE multicanal para $N = 5$ obtenido mediante modelo y simulación.

Al igual que en el caso monocal, conforme N aumenta, la diferencia entre el valor estimado por el modelo y el proporcionado mediante simulación es mayor. No obstante, los valores estimados por el modelo pueden considerarse una buena aproximación del funcionamiento del algoritmo en estado estacionario.

También puede observarse que para $N = 2$ y μ grandes, las simulaciones ofrecen resultados del MSE mucho más optimistas que las estimaciones proporcionadas por el modelo alcanzado. Aunque la tendencia observada para el MSE (considerando el resto de los órdenes) según los datos de simulación, es a crecer en la misma proporción que la que se presenta en el modelo, para $N = 2$ las simulaciones han presentado en este caso un comportamiento sustancialmente diferente.

5.5.2. Estructura de filtrado-x multicanal

Para extender los resultados obtenidos en el caso de la estructura modificada a la de filtrado-x, actuaremos de forma similar al caso monocal.

El vector de error multicanal, en este caso, vendría expresado por:

$$\mathbf{e}[n] = -\mathbf{e}^{a'}[n] + \mathbf{r}[n], \quad (5.143)$$

siendo $\mathbf{e}^{a'}[n]$, el vector de tamaño $KN \times 1$, definido como:

$$\mathbf{e}^{a'}[n] = \begin{pmatrix} \mathbf{v}_1^T[n]\tilde{\mathbf{w}}[n-1] \\ \mathbf{v}_1^T[n-1]\tilde{\mathbf{w}}[n-2] \\ \vdots \\ \mathbf{v}_1^T[n-N+1]\tilde{\mathbf{w}}[n-N] \\ \mathbf{v}_2^T[n]\tilde{\mathbf{w}}[n-1] \\ \mathbf{v}_2^T[n-1]\tilde{\mathbf{w}}[n-2] \\ \vdots \\ \mathbf{v}_2^T[n-N+1]\tilde{\mathbf{w}}[n-N] \\ \vdots \\ \mathbf{v}_K^T[n]\tilde{\mathbf{w}}[n-1] \\ \mathbf{v}_K^T[n-1]\tilde{\mathbf{w}}[n-2] \\ \vdots \\ \mathbf{v}_K^T[n-N+1]\tilde{\mathbf{w}}[n-N] \end{pmatrix} = \begin{pmatrix} e_1^a[n] \\ e_1^a[n-1] \\ \vdots \\ e_1^a[n-N+1] \\ e_2^a[n] \\ e_2^a[n-1] \\ \vdots \\ e_2^a[n-N+1] \\ \vdots \\ e_K^a[n] \\ e_K^a[n-1] \\ \vdots \\ e_K^a[n-N+1] \end{pmatrix}. \quad (5.144)$$

La introducción de este vector nos permite continuar el desarrollo realizado para la estructura modificada, a partir de la ecuación (5.112), sin más que sustituir dicho valor cuando aparece el vector de error $\mathbf{e}[n]$.

Como en el caso con estructura modificada, conviene desacoplar de alguna forma dicho vector para trabajar con la información en cada sensor de forma independiente.

En este caso, también se cumple:

$$\mathbf{e}_k[n] = M_k \mathbf{e}[n], \quad \mathbf{e}_k^{a'}[n] = M_k \mathbf{e}^{a'}[n], \quad \mathbf{r}_k[n] = M_k \mathbf{r}[n], \quad (5.145)$$

donde al igual que con la estructura modificada, la matriz M_k es una matriz de tamaño $NK \times NK$ de la misma forma que los ejemplos mostrados en (5.118).

Estas transformaciones nos permitirían obtener las aportaciones en cada sensor de cada señal, replicada tantas veces como sensores haya en el sistema. Como ya se comentó en el desarrollo para la estructura modificada, es evidente que dicha transformación no es invertible, por lo que no se podrá obtener el vector $\mathbf{e}^{a'}[n]$ a partir de cualquiera de los $\mathbf{e}_k^{a'}[n]$. Así que para poder continuar con nuestro desarrollo a partir de (5.112), realizamos la aproximación $\mathbf{e}_k^{a'}[n] \approx \mathbf{e}^{a'}[n]$. De esta forma, podemos escribir:

$$\begin{aligned} & -\mu(\mathbf{r}_k^T[n] - (\mathbf{e}_k^{a'}[n])^T) \mathbf{N}_k[n] (-\mathbf{e}_k^{a'}[n] + \mathbf{r}_k[n]) \\ & = (\mathbf{e}_k^a[n])^T (-\mathbf{e}_k^{a'}[n] + \mathbf{r}_k[n]) + (\mathbf{r}_k^T[n] - (\mathbf{e}_k^{a'}[n])^T) \mathbf{e}_k^a[n]. \end{aligned} \quad (5.146)$$

Tomando la esperanza matemática y considerando independencia estadística entre los datos de los vectores $\mathbf{r}_k[n]$ y $\mathbf{e}_k^{a'}[n]$, se obtiene:

$$\begin{aligned} & -\mu E\{\mathbf{r}_k^T[n] \mathbf{N}_k[n] \mathbf{r}_k[n]\} - \mu E\{(\mathbf{e}_k^{a'}[n])^T \mathbf{N}_k[n] \mathbf{e}_k^{a'}[n]\} \\ & = -2E\{(\mathbf{e}_k^{a'}[n])^T \mathbf{e}_k^a[n]\}, \end{aligned} \quad (5.147)$$

expresión análoga a (5.120), de la que, de igual forma que se obtuvo (5.121), ahora alcanzaremos la siguiente igualdad:

$$\begin{aligned} & \mu Tr(E\{\mathbf{e}_k^{a'}[n] (\mathbf{e}_k^{a'}[n])^T \mathbf{N}_k[n]\}) + \mu Tr(E\{\mathbf{r}_k[n] \mathbf{r}_k^T[n] \mathbf{N}_k[n]\}) \\ & = 2Tr(E\{\mathbf{e}_k^{a'}[n] (\mathbf{e}_k^a[n])^T\} \mathbf{I}). \end{aligned} \quad (5.148)$$

Considerando independencia estadística entre los datos de la matriz $\mathbf{N}_k[n]$ y los de los vectores con $\mathbf{e}_k^{a'}[n]$ y $\mathbf{r}_k[n]$, se obtiene

$$\begin{aligned} & \mu Tr(E\{\mathbf{e}_k^{a'}[n] (\mathbf{e}_k^{a'}[n])^T\} E\{\mathbf{N}_k[n]\}) + \mu Tr(E\{\mathbf{r}_k[n] \mathbf{r}_k^T[n]\} E\{\mathbf{N}_k[n]\}) \\ & = 2Tr(E\{\mathbf{e}_k^{a'}[n] (\mathbf{e}_k^a[n])^T\} \mathbf{I}). \end{aligned} \quad (5.149)$$

De nuevo, nos centraremos en calcular los valores de las diagonales de $E\{\mathbf{e}_k^{a'}[n] (\mathbf{e}_k^{a'}[n])^T\}$, $E\{\mathbf{e}_k^{a'}[n] (\mathbf{e}_k^a[n])^T\}$ y $E\{\mathbf{r}_k[n] (\mathbf{r}_k[n])^T\}$.

Los valores de la diagonal de $E\{\mathbf{e}_k^{a'}[n](\mathbf{e}_k^{a'}[n])^T\}$ estarían formados por los valores del vector $[E\{|e_k^a[n]|^2\}, E\{|e_k^a[n-1]|^2\}, \dots, E\{|e_k^a[n-N+1]|^2\}]$, replicados K veces. En estado estacionario, todos los valores de dicha diagonal coincidirían con $E\{|e_k^a[n]|^2\}$.

Para los valores de la diagonal de $E\{\mathbf{r}_k[n](\mathbf{r}_k[n])^T\}$, se obtendría el valor $\sigma_{r_k}^2$ (varianza de la señal aleatoria $r_k[n]$, formada según se indica en (5.145)). Nos faltaría calcular $E\{\mathbf{e}_k^{a'}[n](\mathbf{e}_k^a[n])^T\}$.

Para ello, volvemos a hacer uso de la relación definida en (5.124), donde introduciendo (5.143) obtenemos:

$$\begin{aligned} \mathbf{e}^p[n] &= \mathbf{e}^a[n] + \mu \mathbf{V}^T[n] \mathbf{U}[n] (-\mathbf{e}^{a'}[n] + \mathbf{r}[n]) \\ &= \mathbf{e}^a[n] - \mu \mathbf{V}^T[n] \mathbf{U}[n] \mathbf{e}^{a'}[n] + \mu \mathbf{V}^T[n] \mathbf{U}[n] \mathbf{r}[n]. \end{aligned} \quad (5.150)$$

Realizando la multiplicación de $\mathbf{e}^{a'}[n](\mathbf{e}^p[n])^T$, se obtiene la siguiente igualdad:

$$\begin{aligned} \mathbf{e}^{a'}[n](\mathbf{e}^p[n])^T &= \mathbf{e}^{a'}[n](\mathbf{e}^a[n])^T - \mu \mathbf{e}^{a'}[n](\mathbf{e}^{a'}[n])^T \mathbf{U}^T[n] \mathbf{V}[n] \\ &\quad + \mu \mathbf{e}^{a'}[n] \mathbf{r}^T[n] \mathbf{U}^T[n] \mathbf{V}[n]. \end{aligned} \quad (5.151)$$

Tomando la esperanza matemática de dicha expresión y considerando independencia estadística entre los datos de las señales de referencia contenidos en las matrices $\mathbf{U}[n]$ y $\mathbf{V}[n]$, los datos del vector $\mathbf{e}^{a'}[n]$ y del vector $\mathbf{r}[n]$, obtenemos:

$$E\{\mathbf{e}^{a'}[n](\mathbf{e}^p[n])^T\} = E\{\mathbf{e}^{a'}[n](\mathbf{e}^a[n])^T\} - \mu I J E\{\mathbf{e}^{a'}[n](\mathbf{e}^{a'}[n])^T\}. \quad (5.152)$$

Para alcanzar el resultado expresado en (5.152) se ha tenido en cuenta, de nuevo, el resultado expresado en el apartado 5.6.8.

Multiplicamos ambos miembros de dicha ecuación por la izquierda por $E\{\mathbf{M}_k[n]\}$ y por la derecha por $E\{\mathbf{M}_k^T[n]\}$, se obtiene:

$$E\{\mathbf{e}_k^{a'}[n](\mathbf{e}_k^p[n])^T\} = E\{\mathbf{e}_k^{a'}[n](\mathbf{e}_k^a[n])^T\} - \mu I J E\{\mathbf{e}_k^{a'}[n](\mathbf{e}_k^{a'}[n])^T\}. \quad (5.153)$$

de donde, al igual que se realizó en el caso monocal (descrito en 5.6.5), operando solo con las diagonales de (5.153), ecuación que como puede comprobarse es equivalente a la ecuación (5.193) del caso monocal, se consigue expresar $E\{\mathbf{e}_k^{a'}[n](\mathbf{e}_k^a[n])^T\}$ en función del error a priori, tal y como se muestra en la siguiente relación:

$$E\{\mathbf{e}'_k[n](\mathbf{e}_k[n])^T\} = E\{|e_k^a[n]|^2\}D_{M33}. \quad (5.154)$$

La matriz D_{M33} , es una matriz de tamaño $(KN \times KN)$, en cuya diagonal aparecen replicados K veces los valores:

$$[1, (1 - \mu IJ), (1 - 2\mu IJ), \dots, (1 - \mu IJ(N - 1))].$$

Introduciendo en la ecuación (5.149) los resultados alcanzados, se obtiene:

$$\begin{aligned} \mu E\{|e_k^a[n]|^2\}Tr(E\{\mathbf{N}_k\}) + \mu\sigma_{rk}^2 Tr(E\{\mathbf{N}_k\}) \\ = 2E\{|e_k^a[n]|^2\}Tr(D_{M33}), \end{aligned} \quad (5.155)$$

de donde podemos despejar:

$$E\{|e_k^a[n]|^2\} = \frac{\mu\sigma_{rk}^2 Tr(E\{\mathbf{N}_k\})}{2Tr(D_{M33} - \mu E\{\mathbf{N}_k\})}. \quad (5.156)$$

Finalmente, al igual que en (5.142), podemos expresar el valor del MSE como:

$$MSE = K\sigma_r^2 + \sum_{k=1}^K \lim_{n \rightarrow \infty} E\{|e_k^a[n]|^2\}. \quad (5.157)$$

Usando el mismo sistema acústico que en las simulaciones realizadas para el algoritmo con estructura modificada, y las mismas señales de referencia y ruido, se han vuelto a obtener valores del MSE mediante simulación, pero para el algoritmo con estructura de filtrado-x. En este caso, los márgenes de variación del parámetro de convergencia μ han sido restringidos solo a aquellos valores de dicho parámetro para el cual la simulaciones convergen. En las figuras 5.26-5.29, podemos comparar las curvas ofrecidas por el modelo frente a las obtenidas mediante simulación, para diferentes órdenes de proyección.

Al igual que en los casos anteriores, las diferencias entre los valores proporcionados por el modelo y obtenidos mediante simulación aumenta conforme el orden de proyección es mayor. En este caso, las diferencias son algo más acentuadas conforme el orden de proyección crece, en comparación con los resultados obtenidos para la estructura modificada, pero aun así, para órdenes de proyección pequeños (inferiores a 5) y dentro del margen de funcionamiento sin divergencia de estos algoritmos (μ apropiados para que el algoritmo no diverja) los valores estimados por el modelo se aproximan bastante bien a los obtenidos mediante simulación.

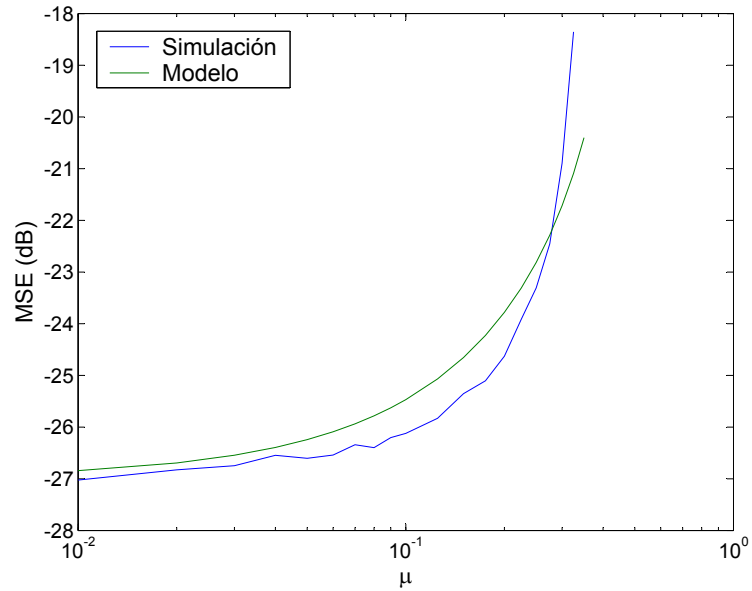


Figura 5.26: Comparación entre el MSE multicanal con estructura de filtrado-x para $N = 2$ obtenido mediante modelo y simulación.

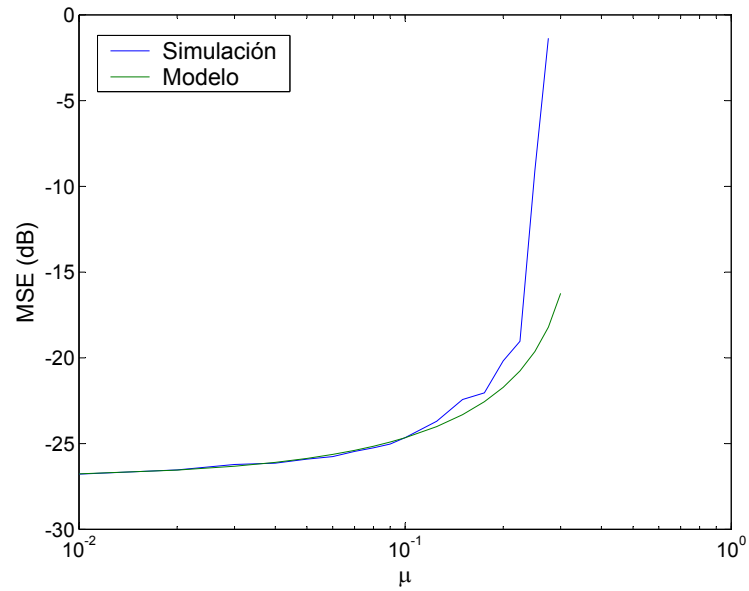


Figura 5.27: Comparación entre el MSE multicanal con estructura de filtrado-x para $N = 3$ obtenido mediante modelo y simulación.

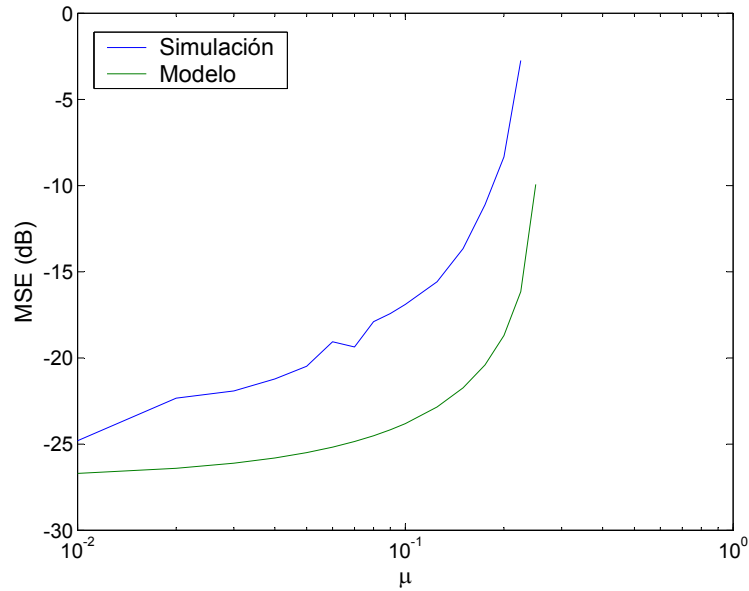


Figura 5.28: Comparación entre el MSE multicanal con estructura de filtrado-x para $N = 4$ obtenido mediante modelo y simulación.

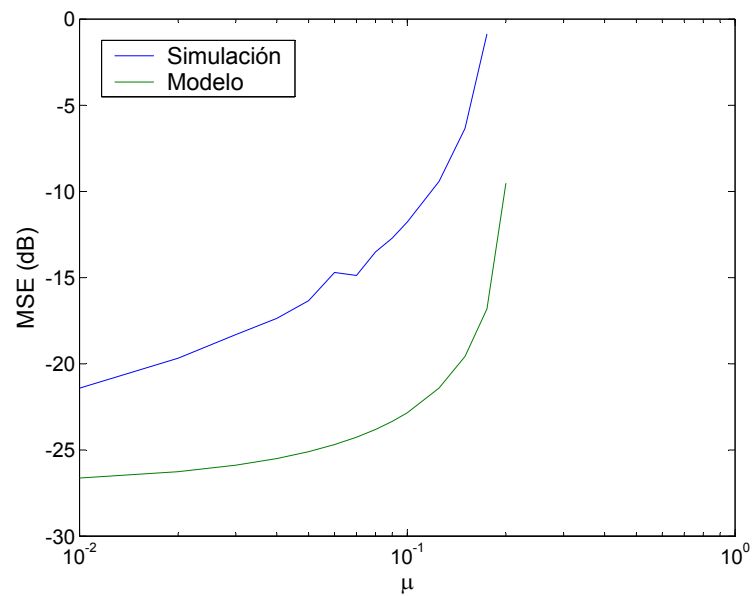


Figura 5.29: Comparación entre el MSE multicanal con estructura de filtrado-x para $N = 5$ obtenido mediante modelo y simulación.

5.6. Resumen de relaciones matemáticas

5.6.1. Obtención de la relación de conservación de la energía

La energía de un vector \mathbf{x} se define como $\|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{x}$. Comenzando con el término de la izquierda de (5.11):

$$\begin{aligned}
& \|\tilde{\mathbf{w}}_L[n] + \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\|^2 \\
&= \{\tilde{\mathbf{w}}_L[n] + \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\}^T \{\tilde{\mathbf{w}}_L[n] + \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\} \\
&= \tilde{\mathbf{w}}_L^T[n]\tilde{\mathbf{w}}_L[n] + \{\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\}^T \{\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\} \\
&\quad + \tilde{\mathbf{w}}_L^T[n]\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n] + \{\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\}^T \tilde{\mathbf{w}}_L[n] \\
&= |\tilde{\mathbf{w}}_L[n]|^2 + \{(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\}^T \mathbf{A}[n]\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n] \\
&\quad + \tilde{\mathbf{w}}_L^T[n]\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n] + \{(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\}^T \mathbf{A}[n]\tilde{\mathbf{w}}_L[n] \\
&= |\tilde{\mathbf{w}}_L[n]|^2 + (\mathbf{e}_N^a[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n] \\
&\quad + (\mathbf{e}_N^p[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n] + (\mathbf{e}_N^a[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n].
\end{aligned} \tag{5.158}$$

Análogamente, para el término de la derecha:

$$\begin{aligned}
& \|\tilde{\mathbf{w}}_L[n-1] + \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^p[n]\|^2 \\
&= |\tilde{\mathbf{w}}_L[n-1]|^2 + (\mathbf{e}_N^p[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n] \\
&\quad + (\mathbf{e}_N^a[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n] + (\mathbf{e}_N^p[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n].
\end{aligned} \tag{5.159}$$

Igualando ambos términos nos quedará la expresión ya descrita en (5.12):

$$\begin{aligned}
& \|\tilde{\mathbf{w}}_L[n]\|^2 + (\mathbf{e}_N^a[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^a[n] \\
&= \|\tilde{\mathbf{w}}_L[n-1]\|^2 + (\mathbf{e}_N^p[n])^T (\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \mathbf{e}_N^p[n].
\end{aligned} \tag{5.160}$$

5.6.2. Manipulaciones matemáticas a la ecuación de conservación de la energía

Partiendo de (5.6), e introduciendo dicha ecuación en el segundo miembro de la igualdad de (5.14), tenemos:

$$\begin{aligned}
& E\{(\mathbf{e}_N^p[n])^T(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^p[n]\} \\
&= E\{(\mathbf{e}_N^a[n] - \mu\mathbf{A}[n]\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{e}_N[n])^T(\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \\
&\quad \cdot (\mathbf{e}_N^a[n] - \mu\mathbf{A}[n]\mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{e}_N[n])\} \\
&= E\{(\mathbf{e}_N^a[n])^T(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\} \\
&\quad - E\{\mu\mathbf{e}_N^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}(\mathbf{A}[n]\mathbf{A}^T[n])(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\} \\
&\quad - E\{(\mathbf{e}_N^a[n])^T(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mu(\mathbf{A}[n]\mathbf{A}^T[n])(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{e}_N[n]\} \\
&\quad + E\{\mu\mathbf{e}_N^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}(\mathbf{A}[n]\mathbf{A}^T[n])(\mathbf{A}[n]\mathbf{A}^T[n])^{-1} \\
&\quad \cdot \mu(\mathbf{A}[n]\mathbf{A}^T[n])(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{e}_N[n]\} \\
&= E\{(\mathbf{e}_N^a[n])^T(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\} \\
&\quad - \mu E\{\mathbf{e}_N^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{e}_N^a[n]\} \\
&\quad - \mu E\{(\mathbf{e}_N^a[n])^T(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{e}_N[n]\} \\
&+ \mu^2 E\{\mathbf{e}_N^T[n](\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}(\mathbf{A}[n]\mathbf{A}^T[n])(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}\mathbf{e}_N[n]\} \\
&= E\{(\mathbf{e}_N^a[n])^T(\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N^a[n]\} - \mu E\{\mathbf{e}_N^T[n]\Phi[n]\mathbf{e}_N^a[n]\} \\
&\quad - \mu E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N[n]\} + \mu^2 E\{\mathbf{e}_N^T[n]\Psi[n]\mathbf{e}_N[n]\}.
\end{aligned} \tag{5.161}$$

Siendo

$$\Phi[n] = (\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1} \tag{5.162}$$

y

$$\Psi[n] = (\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}(\mathbf{A}[n]\mathbf{A}^T[n])(\mathbf{A}[n]\mathbf{A}^T[n] + \delta I)^{-1}. \tag{5.163}$$

Si nos damos cuenta, el primer término del resultado obtenido en (5.161) coincide con la parte izquierda de la ecuación (5.14). Por tanto, la ecuación (5.14), puede reescribirse como sigue:

$$\mu E\{\mathbf{e}_N^T[n]\Psi[n]\mathbf{e}_N[n]\} = E\{\mathbf{e}_N^T[n]\Phi[n]\mathbf{e}_N^a[n]\} + E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N[n]\}. \tag{5.164}$$

5.6.3. Eliminación de la dependencia de $\mathbf{e}_N[n]$ en la ecuación de conservación de energía

Para el término de la izquierda de la igualdad:

$$\begin{aligned}
E\{\mathbf{e}_N^T[n]\Psi[n]\mathbf{e}_N[n]\} &= E\{(\mathbf{e}_N^a[n] + \mathbf{r}_N[n])^T\Psi[n](\mathbf{e}_N^a[n] + \mathbf{r}_N[n])\} \\
&= E\{(\mathbf{e}_N^a[n])^T\Psi[n]\mathbf{e}_N^a[n] + (\mathbf{e}_N^a[n])^T\Psi[n]\mathbf{r}_N[n] \\
&\quad + \mathbf{r}_N^T[n]\Psi[n]\mathbf{e}_N^a[n] + \mathbf{r}_N^T[n]\Psi[n]\mathbf{r}_N[n]\} \\
&= E\{(\mathbf{e}_N^a[n])^T\Psi[n]\mathbf{e}_N^a[n]\} + E\{\mathbf{r}_N^T[n]\Psi[n]\mathbf{r}_N[n]\} + E\{2(\mathbf{r}_N^T[n]\Psi[n]\mathbf{e}_N^a[n])\},
\end{aligned} \tag{5.165}$$

donde se ha considerado la simetría de la matriz $\Psi[n]$.

Para los términos de la derecha de (5.15):

$$\begin{aligned} E\{\mathbf{e}_N^T[n]\Phi[n]\mathbf{e}_N^a[n]\} &= E\{(\mathbf{e}_N^a[n] + \mathbf{r}_N[n])^T\Phi[n]\mathbf{e}_N^a[n]\} \\ &= E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N^a[n]\} + E\{\mathbf{r}_N^T[n]\Phi[n]\mathbf{e}_N^a[n]\} \end{aligned} \quad (5.166)$$

$$\begin{aligned} E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N[n]\} &= E\{(\mathbf{e}_N^a[n])^T\Phi[n](\mathbf{e}_N^a[n] + \mathbf{r}_N[n])\} \\ &= E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N^a[n]\} + E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{r}_N[n]\}, \end{aligned} \quad (5.167)$$

con lo que, considerando que $\Phi[n]$ es una matriz simétrica, la ecuación (5.15) puede reescribirse como:

$$\begin{aligned} \mu E\{(\mathbf{e}_N^a[n])^T\Psi[n]\mathbf{e}_N^a[n]\} + \mu E\{\mathbf{r}_N^T[n]\Psi[n]\mathbf{r}_N[n]\} + \mu E\{2(\mathbf{r}_N^T[n]\Psi[n]\mathbf{e}_N^a[n])\} \\ = 2E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N^a[n]\} + E\{2(\mathbf{r}_N^T[n]\Phi[n]\mathbf{e}_N^a[n])\}, \end{aligned} \quad (5.168)$$

que a su vez, se puede expresar de la siguiente forma:

$$\begin{aligned} \mu E\{(\mathbf{e}_N^a[n])^T\Psi[n]\mathbf{e}_N^a[n]\} + \mu E\{\mathbf{r}_N^T[n]\Psi[n]\mathbf{r}_N[n]\} \\ = 2E\{(\mathbf{e}_N^a[n])^T\Phi[n]\mathbf{e}_N^a[n]\}. \end{aligned} \quad (5.169)$$

Considerando que tanto la señal $r[n]$ como $e_N^a[n]$ son estadísticamente independientes de la señal $x[n]$ (o sea, de las matrices $\Phi[n]$ y $\Psi[n]$ formadas por valores de $x[n]$). También podría haberse considerado que $\mu \approx 1$ y $\delta \approx 0$ (caso frecuente), llegándose a la misma conclusión.

5.6.4. Obtención del modelo del EMSE monocanal usando actualización eficiente del vector de error (estructura de filtrado-x modificada)

Partiendo de (5.63), y considerando que solo vamos a calcular las diagonales de las matrices $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$, $E\{\mathbf{e}_N^a[n](\mathbf{e}_{vN}^a[n])^T\}$ y $E\{\mathbf{r}'[n]\mathbf{r}'^T[n]\}$, tratamos de encontrar una expresión similar a (5.41) de donde pueda deducirse el valor del EMSE definido ahora como

$$\lim_{n \rightarrow \infty} E\{|e_v^a[n]|^2\} \quad (5.170)$$

siendo, $e_v^a[n] = \mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]$.

Así, para $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$, buscaremos los valores de

$$\left(\begin{array}{c} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ (1-\mu)^2 E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\} \\ (1-\mu)^2 E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2]\} \\ \vdots \\ (1-\mu)^2 E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-2]\} \end{array} \right) \quad (5.171)$$

para $E\{\mathbf{e}'_N[n](\mathbf{e}^a_{vN}[n])^T\}$, buscaremos los valores de

$$\left(\begin{array}{c} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ (1-\mu)E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\} \\ (1-\mu)E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\} \\ \vdots \\ (1-\mu)E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1]\} \end{array} \right) \quad (5.172)$$

mientras que para la diagonal de $E\{\mathbf{r}'[n](\mathbf{r}'[n])^T\}$, se obtendrá

$$\left(\begin{array}{c} \sigma_r^2 \\ \sigma_r^2(1-\mu)^2 \\ \sigma_r^2(1-\mu)^2 \\ \vdots \\ \sigma_r^2(1-\mu)^2 \end{array} \right). \quad (5.173)$$

Los dos primeros términos de (5.171) se corresponden con $E\{|e_v^a[n]|^2\}$ y $(1-\mu)^2 E\{|e_v^a[n-1]|^2\}$, mientras que el primer término de (5.172) sería $E\{|e_v^a[n]|^2\}$. Para calcular el resto de términos, hemos de considerar que en un sistema de CAR la relación existente entre los vectores de error a priori y a posteriori (si $\delta \approx 0$) definidos en (5.53) y (5.54) es ahora:

$$\mathbf{e}^p_{vN}[n] = \mathbf{e}^a_{vN}[n] + \mu \mathbf{e}_N[n]. \quad (5.174)$$

Al introducir el valor de $\mathbf{e}_N[n]$ en (5.174) y transponer, se obtiene

$$(\mathbf{e}^p_{vN}[n])^T = (\mathbf{e}^a_{vN}[n])^T - \mu(\mathbf{e}'_N[n])^T + \mu \mathbf{r}'^T_N[n]. \quad (5.175)$$

Multiplicando por la izquierda ambos términos de dicha igualdad por $\mathbf{e}^a_{vN}[n]$ y tomando esperanzas matemáticas (considerando independencia estadística entre $\mathbf{e}^a_{vN}[n]$ y $\mathbf{r}'[n]$) llegamos a la expresión:

$$E\{\mathbf{e}^a_{vN}[n](\mathbf{e}^p_{vN}[n])^T\} = E\{\mathbf{e}^a_{vN}[n](\mathbf{e}^a_{vN}[n])^T\} - \mu E\{\mathbf{e}^a_{vN}[n](\mathbf{e}'_N[n])^T\}. \quad (5.176)$$

Multiplicando ahora por la izquierda la expresión (5.175) por $\mathbf{e}_{vN}^p[n]$ y volviendo a tomar esperanzas matemáticas, considerando que $\mathbf{e}_N^{\alpha'}[n]$ también es independiente de $\mathbf{r}'[n]$, llegamos a la igualdad:

$$E\{\mathbf{e}_{vN}^p[n](\mathbf{e}_{vN}^p[n])^T\} = E\{\mathbf{e}_{vN}^{\alpha}[n](\mathbf{e}_{vN}^{\alpha}[n])^T\} - 2\mu E\{\mathbf{e}_{vN}^{\alpha}[n](\mathbf{e}_N^{\alpha'}[n])^T\} + \mu^2 E\{\mathbf{e}_N^{\alpha'}[n](\mathbf{e}_N^{\alpha'}[n])^T\} + \mu^2 E\{\mathbf{r}'[n]\mathbf{r}'_N^T[n]\}. \quad (5.177)$$

Calculemos término a término cada elemento de las diagonales de (5.176) y (5.177) y para simplificar nomenclatura trataremos de expresar cada término como una combinación lineal de $E\{|e_v^{\alpha}[n]|^2\}$ y $\mu^2\sigma_r^2$. Hay que recordar que la diagonal de $E\{\mathbf{e}_{vN}^{\alpha}[n](\mathbf{e}_{vN}^{\alpha}[n])^T\}$ estaría formada por:

$$\begin{pmatrix} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n]\} \\ E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n]\} \\ E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n]\} \\ \vdots \\ E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n]\} \end{pmatrix} = \begin{pmatrix} E_1^{ap}[n] \\ E_2^{ap}[n] \\ E_3^{ap}[n] \\ \vdots \\ E_N^{ap}[n] \end{pmatrix} \quad (5.178)$$

y la de $E\{\mathbf{e}_{vN}^p[n](\mathbf{e}_{vN}^p[n])^T\}$

$$\begin{pmatrix} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n]\} \\ E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n]\} \\ E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n]\} \\ \vdots \\ E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n]\} \end{pmatrix} = \begin{pmatrix} E_1^{pp}[n] \\ E_2^{pp}[n] \\ E_3^{pp}[n] \\ \vdots \\ E_N^{pp}[n] \end{pmatrix}. \quad (5.179)$$

El primer elemento de la diagonal de (5.176) (que coincide con $E_1^{ap}[n]$ de (5.178)) sería

$$E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} - \mu E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} = (1-\mu)E\{|e_v^{\alpha}[n]|^2\} = k_1^{ap}E\{|e_v^{\alpha}[n]|^2\} + k_1^{\sigma ap}\mu^2\sigma_r^2, \quad y$$

el primer elemento de la diagonal de (5.177) (que coincide con $E_1^{pp}[n]$ de (5.179)) vendría dado por

$$\begin{aligned} & E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ & - 2\mu E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ & + \mu^2 E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} + \mu^2\sigma_r^2 \\ & = (1-\mu)^2 E\{|e_v^{\alpha}[n]|^2\} + \mu^2\sigma_r^2 = k_1^{pp}E\{|e_v^{\alpha}[n]|^2\} + k_1^{\sigma pp}\mu^2\sigma_r^2. \end{aligned}$$

Los segundos elementos de (5.178) y (5.179) pueden expresarse, apoyándose en las diagonales de las matrices (5.176) y (5.177) como:

$$\begin{aligned} E_2^{ap}[n] &= E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\} \\ & - \mu(1-\mu)E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\} \\ & = E_1^{pp}[n-1] - \mu(1-\mu)E_1^{ap}[n-1] = k_2^{ap}E\{|e_v^{\alpha}[n-1]|^2\} + k_2^{\sigma ap}\mu^2\sigma_r^2. \end{aligned}$$

$$\begin{aligned}
E_2^{pp}[n] &= E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\} \\
&\quad - 2\mu(1-\mu)E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\} \\
&\quad + \mu^2(1-\mu)^2E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\} + (1-\mu)^2\mu^2\sigma_r^2 \\
&= E_1^{pp}[n-1] - 2\mu(1-\mu)E_1^{ap}[n-1] + \mu^2(1-\mu)^2E\{|e_v^a[n-1]|^2\} \\
&\quad + (1-\mu)^2\mu^2\sigma_r^2 = k_2^{pp}E\{|e_v^a[n-1]|^2\} + k_2^{\sigma pp}\mu^2\sigma_r^2.
\end{aligned}$$

Para el tercer elemento tendríamos en cada caso:

$$\begin{aligned}
E_3^{ap}[n] &= E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\} \\
&\quad - \mu(1-\mu)E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2]\} \\
&= E_2^{pp}[n-1] - \mu(1-\mu)E_2^{ap}[n-1] = k_3^{ap}E\{|e_v^a[n-2]|^2\} + k_3^{\sigma ap}\mu^2\sigma_r^2.
\end{aligned}$$

$$\begin{aligned}
E_3^{pp}[n] &= E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\} \\
&\quad - 2\mu(1-\mu)E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2]\} \\
&\quad + \mu^2(1-\mu)^2E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2]\} + (1-\mu)^2\mu^2\sigma_r^2 \\
&= E_2^{pp}[n-1] - 2\mu(1-\mu)E_2^{ap}[n-1] + \mu^2(1-\mu)^2E_1^{pp}[n-2] \\
&\quad + (1-\mu)^2\mu^2\sigma_r^2 = k_3^{pp}E\{|e_v^a[n-2]|^2\} + k_3^{\sigma pp}\mu^2\sigma_r^2
\end{aligned}$$

de donde se puede deducir un término general para los valores j -ésimos

$$E_j^{ap}[n] = E_{j-1}^{pp}[n-1] - \mu(1-\mu)E_{j-1}^{ap}[n-1] = k_j^{ap}E\{|e_v^a[n-j+1]|^2\} + k_j^{\sigma ap}\mu^2\sigma_r^2.$$

$$\begin{aligned}
E_j^{pp}[n] &= E_{j-1}^{pp}[n-1] - 2\mu(1-\mu)E_{j-1}^{ap}[n-1] + \mu^2(1-\mu)^2E_{j-2}^{pp}[n-2] \\
&\quad + (1-\mu)^2\mu^2\sigma_r^2 = k_j^{pp}E\{|e_v^a[n-j+1]|^2\} + k_j^{\sigma pp}\mu^2\sigma_r^2.
\end{aligned}$$

Las constantes que aparecen en los términos anteriores, también pueden calcularse usando relaciones iterativas. Así:

$$\begin{array}{ll}
k_1^{ap} = (1-\mu) & k_1^{\sigma ap} = 0 \\
k_2^{ap} = k_1^{pp} - \mu(1-\mu)k_1^{ap} & k_2^{\sigma ap} = k_1^{\sigma pp} \\
k_3^{ap} = k_2^{pp} - \mu(1-\mu)k_2^{ap} & k_3^{\sigma ap} = k_2^{\sigma pp} - \mu(1-\mu)k_2^{\sigma ap} \\
\vdots & \vdots \\
k_j^{ap} = k_{j-1}^{pp} - \mu(1-\mu)k_{j-1}^{ap} & k_j^{\sigma ap} = k_{j-1}^{\sigma pp} - \mu(1-\mu)k_{j-1}^{\sigma ap}
\end{array}$$

$$\begin{aligned}
k_1^{pp} &= (1-\mu)^2 \\
k_2^{pp} &= k_1^{pp} - 2\mu(1-\mu)k_1^{ap} + \mu^2(1-\mu)^2 \\
k_3^{pp} &= k_2^{pp} - 2\mu(1-\mu)k_2^{ap} + \mu^2(1-\mu)^2k_1^{pp} \\
&\quad \vdots \\
k_j^{pp} &= k_{j-1}^{pp} - 2\mu(1-\mu)k_{j-1}^{ap} + \mu^2(1-\mu)^2k_{j-2}^{pp}
\end{aligned}$$

$$\begin{aligned}
k_1^{\sigma pp} &= 1 \\
k_2^{\sigma pp} &= k_1^{\sigma pp} + (1-\mu)^2 \\
k_3^{\sigma pp} &= k_2^{\sigma pp} - 2\mu(1-\mu)k_2^{\sigma ap} + \mu^2(1-\mu)^2k_1^{\sigma pp} + (1-\mu)^2 \\
&\quad \vdots \\
k_j^{\sigma pp} &= k_{j-1}^{\sigma pp} - 2\mu(1-\mu)k_{j-1}^{\sigma ap} + \mu^2(1-\mu)^2k_{j-2}^{\sigma pp} + (1-\mu)^2.
\end{aligned}$$

Por simple inspección visual se puede deducir que los términos j -ésimos (a partir del tercero) de la matriz (5.171) se corresponden con los valores de $(1 - \mu)^2 E_{j-2}^{pp}[n - 2]$, mientras que los términos j -ésimos (a partir del segundo) de (5.172) se corresponden con $(1 - \mu) E_{j-1}^{ap}[n - 1]$.

Además, considerando que en estado estacionario, ($n \rightarrow \infty$), se cumple que $E\{|e_v^\alpha[n]|^2\} = E\{|e_v^\alpha[n - 1]|^2\} = \dots = E\{|e_v^\alpha[n - N + 1]|^2\}$, podemos obtener la siguientes relaciones:

$$E\{\mathbf{e}_N^{\alpha'}[n](\mathbf{e}_N^{\alpha'}[n])^T\} \approx E\{|e_v^\alpha[n]|^2\} D_{21} + \mu^2 \sigma_r^2 D_{22}, \quad (5.180)$$

siendo D_{21} y D_{22} las siguientes matrices diagonales:

$$D_{21} = \begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & (1 - \mu)^2 & 0 & \dots & \vdots \\ \vdots & & (1 - \mu)^2 k_1^{pp} & & \\ & & & \ddots & 0 \\ 0 & \dots & & & (1 - \mu)^2 k_{N-2}^{pp} \end{pmatrix} \quad (5.181)$$

y

$$D_{22} = \begin{pmatrix} 0 & \dots & & & 0 \\ \vdots & 0 & & & \vdots \\ & & (1 - \mu)^2 k_1^{\sigma pp} & \dots & \\ & & & \ddots & 0 \\ 0 & \dots & & & (1 - \mu)^2 k_{N-2}^{\sigma pp} \end{pmatrix}. \quad (5.182)$$

Por tanto,

$$E\{\mathbf{e}_N^{\alpha'}[n](\mathbf{e}_N^{\alpha'}[n])^T\} \approx E\{|e_v^\alpha[n]|^2\} D_{23} + \mu^2 \sigma_r^2 D_{24} \quad (5.183)$$

siendo D_{23} y D_{24} las siguientes matrices diagonales:

$$D_{23} = \begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & (1 - \mu) k_1^{ap} & 0 & \dots & \vdots \\ \vdots & & (1 - \mu) k_2^{ap} & & \\ & & & \ddots & 0 \\ 0 & \dots & & & (1 - \mu) k_{N-1}^{ap} \end{pmatrix} \quad (5.184)$$

y

$$D_{24} = \begin{pmatrix} 0 & \cdots & & & 0 \\ \vdots & 0 & \cdots & & \vdots \\ & & (1-\mu) & & \\ & & & \ddots & \\ 0 & \cdots & & & 0 \\ & & & & (1-\mu)k_{N-1}^{\sigma_{ap}} \end{pmatrix}. \quad (5.185)$$

Además,

$$E\{\mathbf{r}'[n](\mathbf{r}'^T[n])\} = D_{25}\sigma_r^2 \quad (5.186)$$

con D_{25} :

$$D_{25} = \begin{pmatrix} 1 & 0 & \cdots & & 0 \\ 0 & (1-\mu)^2 & 0 & \cdots & \vdots \\ \vdots & & (1-\mu)^2 & & \\ & & & \ddots & 0 \\ 0 & \cdots & & 0 & (1-\mu)^2 \end{pmatrix} \quad (5.187)$$

Considerando (5.180), (5.183) y (5.186), podemos despejar el valor de $E\{|e_v^a[n]|^2\}$ de la ecuación (5.63)

$$\begin{aligned} & E\{|e_v^a[n]|^2\} \\ &= \frac{\mu\sigma_r^2 \text{Tr}(D_{25}E\{\Psi_v[n]\}) + (\mu\text{Tr}(D_{22}E\{\Psi_v[n]\}) - 2\text{Tr}(D_{24}E\{\Phi_v[n]\}))\mu^2\sigma_r^2}{2\text{Tr}(D_{23}E\{\Phi_v[n]\}) - \mu\text{Tr}(D_{21}E\{\Psi_v[n]\})}, \end{aligned} \quad (5.188)$$

obteniendo la expresión que aparece en (5.64), como se quería demostrar.

5.6.5. Obtención del modelo del EMSE (estructura de filtrado-x convencional)

Nos volvemos a centrar en el cálculo de las diagonales de las matrices. Así, la diagonal de la operación $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$ que aparece en el primer término del lado izquierdo de la igualdad en (5.70) estaría formado por los valores $E\{|e_v^a[n]|^2\}, E\{|e_v^a[n-1]|^2\}, \dots, E\{|e_v^a[n-N+1]|^2\}$ que en estado estacionario ($n \rightarrow \infty$), satisface que $E\{|e_v^a[n]|^2\} = E\{|e_v^a[n-1]|^2\} = \dots = E\{|e_v^a[n-N+1]|^2\}$.

Por otro lado, para simplificar de algún modo la expresión de la parte derecha de (5.70) tenemos que tratar de operar con $E\{\mathbf{e}_N^{\alpha'}[n](\mathbf{e}_{vN}^{\alpha}[n])^T\}$. La diagonal de dicho producto estaría formado por los elementos del vector:

$$\begin{pmatrix} \mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1] \\ \mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1] \\ \mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1] \\ \vdots \\ \mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1] \end{pmatrix}. \quad (5.189)$$

El primer elemento de dicho vector puede identificarse fácilmente puesto que:

$$\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1] = |e_v^{\alpha}[n]|^2. \quad (5.190)$$

Para el resto de términos consideramos que en un sistema de CAR, la relación existente entre los vectores de error a priori y a posteriori definidos en (5.53) y (5.54), coincide con la definida en (5.174)

$$\mathbf{e}_{vN}^p[n] = \mathbf{e}_{vN}^{\alpha}[n] + \mu \mathbf{e}_N[n] \quad (5.191)$$

y teniendo en cuenta $\mathbf{e}_N[n] = -\mathbf{e}_N^{\alpha'}[n] + \mathbf{r}_N[n]$, podemos reescribir la ecuación (5.191) del siguiente modo:

$$(\mathbf{e}_{vN}^p[n])^T = (\mathbf{e}_{vN}^{\alpha}[n])^T - \mu(\mathbf{e}_N^{\alpha'}[n])^T + \mu\mathbf{r}_N^T[n]. \quad (5.192)$$

Multiplicando por la izquierda ambos términos de la igualdad por $\mathbf{e}_N^{\alpha'}[n]$ y tomando esperanzas matemáticas (considerando independencia estadística entre $\mathbf{e}_N^{\alpha'}[n]$ y $\mathbf{r}_N[n]$) llegamos a la expresión:

$$E\{\mathbf{e}_N^{\alpha'}[n](\mathbf{e}_{vN}^p[n])^T\} = E\{\mathbf{e}_N^{\alpha'}[n](\mathbf{e}_{vN}^{\alpha}[n])^T\} - \mu E\{\mathbf{e}_N^{\alpha'}[n](\mathbf{e}_N^{\alpha'}[n])^T\}. \quad (5.193)$$

Considerando solo las diagonales de (5.193), tendremos:

$$= \begin{pmatrix} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n]\} \\ E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n]\} \\ E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n]\} \\ \vdots \\ E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n]\} \end{pmatrix} \\ -\mu \begin{pmatrix} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\} \\ E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\} \\ \vdots \\ E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1]\} \end{pmatrix} \\ -\mu \begin{pmatrix} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\} \\ E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\} \\ \vdots \\ E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\} \end{pmatrix}. \quad (5.194)$$

Si nos quedamos con la primera fila de (5.194) y retrasamos una unidad dicha igualdad, obtenemos:

$$E\{|\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|\} = E\{|e_v^a[n-1]|^2\} - \mu E\{|e_v^a[n-1]|^2\}. \quad (5.195)$$

Lo que nos sirve para evaluar el segundo elemento de (5.189), quedando:

$$E\{|\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|\} = (1-\mu)E\{|e_v^a[n-1]|^2\}. \quad (5.196)$$

Procediendo del mismo modo para la segunda fila de (5.194), obtenemos

$$E\{|\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]|\} \\ = \mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-2] - \mu E\{|e_v^a[n-2]|^2\}, \quad (5.197)$$

donde el miembro de la izquierda de la igualdad se corresponde con el tercer elemento de (5.189), y en el primer término del lado derecho de la igualdad aparece el segundo elemento de (5.189) retrasado un unidad, por lo que podemos reescribirla como:

$$E\{|\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]|\} \\ = (1-\mu)E\{|e_v^a[n-2]|^2\} - \mu E\{|e_v^a[n-2]|^2\} = (1-2\mu)E\{|e_v^a[n-2]|^2\}. \quad (5.198)$$

De esta forma, podemos encontrar iterativamente una expresión para cada uno de los elementos de (5.189) y puesto que en estado estacionario $E\{|e_v^a[n]|^2\} = E\{|e_v^a[n-1]|^2\} = \dots = E\{|e_v^a[n-N+1]|^2\}$, podemos escribir:

$$E\{\mathbf{e}_N^a[n](\mathbf{e}_{vN}^a[n])^T\} \approx E\{|e_v^a[n]|^2\}D_{33}, \quad (5.199)$$

siendo D_{33} la siguiente matriz diagonal:

$$D_{33} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & (1-\mu) & 0 & \dots & \vdots \\ \vdots & & (1-2\mu) & & \\ & & & \ddots & 0 \\ 0 \dots & & & 0 & (1-(N-1)\mu) \end{pmatrix}. \quad (5.200)$$

Finalmente, la expresión (5.70), podemos reescribirla como

$$\begin{aligned} \mu E\{|e_v^a[n]|^2\}Tr(E\{\Psi_v[n]\}) + \mu\sigma_r^2 Tr(E\{\Psi_v[n]\}) \\ = 2E\{|e_v^a[n]|^2\}Tr(E\{D_{33}\Phi_v[n]\}) \end{aligned} \quad (5.201)$$

y despejando el valor de $E\{|e_v^a[n]|^2\}$ se obtiene:

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 Tr(E\{\Psi_v[n]\})}{2Tr(E\{D_{33}\Phi_v[n]\}) - \mu Tr(E\{\Psi_v[n]\})}, \quad (5.202)$$

tal y como queríamos demostrar.

5.6.6. Obtención del modelo del EMSE usando cálculo eficiente del vector de error (estructura de filtrado-x convencional)

Para operar con (5.78), de nuevo sólo nos interesa calcular los valores de las diagonales de las matrices $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$, $E\{\mathbf{e}_N^a[n](\mathbf{e}_{vN}^a[n])^T\}$ y $E\{\mathbf{r}'_N[n]\mathbf{r}'_N[n]^T\}$, ya que como se ha supuesto en los casos anteriores, las matrices $E\{\Psi_v[n]\}$ y $E\{\Phi_v[n]\}$ se consideran diagonalmente dominantes.

Los elementos de la diagonal de $E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\}$ vienen dados por:

$$\begin{aligned}
& \begin{pmatrix} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ (1-\mu)^2 E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\} \\ (1-\mu)^2 E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\} \\ \vdots \\ (1-\mu)^2 E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\} \end{pmatrix} \\
&= \begin{pmatrix} E\{|e_v^a[n]|^2\} \\ (1-\mu)^2 E\{|e_v^a[n-1]|^2\} \\ (1-\mu)^2 E\{|e_v^a[n-2]|^2\} \\ \vdots \\ (1-\mu)^2 E\{|e_v^a[n-N+1]|^2\} \end{pmatrix}.
\end{aligned} \tag{5.203}$$

Para la diagonal de $E\{\mathbf{r}'_N[n]\mathbf{r}'_N^T[n]\}$, se obtendrá

$$\begin{pmatrix} \sigma_r^2 \\ \sigma_r^2(1-\mu)^2 \\ \sigma_r^2(1-\mu)^2 \\ \vdots \\ \sigma_r^2(1-\mu)^2 \end{pmatrix}, \tag{5.204}$$

mientras que para $E\{\mathbf{e}_N^{\alpha''}[n](\mathbf{e}_{vN}^a[n])^T\}$, buscaremos los valores de

$$\begin{pmatrix} E\{\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\} \\ (1-\mu)E\{\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]\} \\ (1-\mu)E\{\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]\} \\ \vdots \\ (1-\mu)E\{\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N]\mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-1]\} \end{pmatrix}. \tag{5.205}$$

El primer elemento de (5.205) sería $E\{|e_v^a[n]|^2\}$.

Para calcular el resto de valores haremos uso de la siguiente relación obtenida de forma similar a (5.193)

$$E\{\mathbf{e}_N^{\alpha''}[n](\mathbf{e}_{vN}^p[n])^T\} = E\{\mathbf{e}_N^{\alpha''}[n](\mathbf{e}_{vN}^a[n])^T\} - \mu E\{\mathbf{e}_N^{\alpha''}[n](\mathbf{e}_N^{\alpha''}[n])^T\}, \tag{5.206}$$

pero considerando el nuevo valor de $\mathbf{e}_N[n]$. Los elementos de la diagonal de (5.206) se calculan de la forma siguiente:

Para el primer elemento, tendríamos,

$$\begin{aligned}
& E\{|\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n]|^2\} \\
&= E\{|e_v^a[n]|^2\} - \mu E\{|e_v^a[n]|^2\} = (1-\mu)E\{|e_v^a[n]|^2\},
\end{aligned} \tag{5.207}$$

que retrasando una unidad dicha expresión nos serviría para evaluar el segundo elemento de (5.205), quedando:

$$(1 - \mu)E\{|\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|\} = (1 - \mu)^2E\{|e_v^a[n-1]|^2\}. \quad (5.208)$$

Para el segundo elemento:

$$\begin{aligned} & (1 - \mu)E\{|\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n]|\} \\ &= (1 - \mu)E\{|\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2]\mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-1]|\} \\ & \quad - \mu(1 - \mu)^2E\{|e_v^a[n-1]|^2\} = (1 - \mu)^2(1 - \mu)E\{|e_v^a[n-1]|^2\}, \end{aligned} \quad (5.209)$$

que también se relaciona con el tercer elemento de (5.205) retrasándolo una unidad:

$$(1 - \mu)E\{|\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]|\} = (1 - \mu)^2(1 - \mu)E\{|e_v^a[n-2]|^2\}. \quad (5.210)$$

Sirviéndose del tercer elemento de (5.206) y de las expresiones anteriores:

$$\begin{aligned} & (1 - \mu)E\{|\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n]|\} \\ &= (1 - \mu)E\{|\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-3]\mathbf{v}_L^T[n-2]\tilde{\mathbf{w}}_L[n-1]|\} \\ & \quad - \mu(1 - \mu)^2E\{|e_v^a[n-2]|^2\} = (1 - \mu)^2(1 - 2\mu)E\{|e_v^a[n-2]|^2\}, \end{aligned} \quad (5.211)$$

que retrasándolo una unidad proporciona, para el cuarto elemento de (5.205),

$$(1 - \mu)E\{|\mathbf{v}_L^T[n-3]\tilde{\mathbf{w}}_L[n-4]\mathbf{v}_L^T[n-3]\tilde{\mathbf{w}}_L[n-1]|\} = (1 - \mu)^2(1 - 2\mu)E\{|e_v^a[n-3]|^2\}. \quad (5.212)$$

Así, se puede obtener para el elemento j -ésimo de (5.205), siendo $j = 3, \dots, N$,

$$\begin{aligned} & (1 - \mu)E\{|\mathbf{v}_L^T[n-j+1]\tilde{\mathbf{w}}_L[n-j]\mathbf{v}_L^T[n-j+1]\tilde{\mathbf{w}}_L[n-1]|\} \\ &= (1 - \mu)^2(1 - (j-2)\mu)E\{|e_v^a[n-j]|^2\}. \end{aligned} \quad (5.213)$$

Dado que en estado estacionario ($E\{|e_v^a[n]|^2\} = E\{|e_v^a[n-1]|^2\} = \dots = E\{|e_v^a[n-N+1]|^2\}$), tenemos que

$$E\{\mathbf{e}_N^{a''}[n](\mathbf{e}_v^a[n])^T\} \approx E\{|e_v^a[n]|^2\}D_{41}, \quad (5.214)$$

siendo D_{41} la siguiente matriz diagonal:

$$\begin{pmatrix} 1 & 0 & \cdots & & 0 \\ 0 & (1-\mu)^2 & 0 & \cdots & \vdots \\ \vdots & & (1-\mu)(1-\mu)^2 & & \\ & & & (1-2\mu)(1-\mu)^2 & \\ & & & \ddots & 0 \\ 0 & \cdots & & & [1-(N-2)\mu](1-\mu)^2 \end{pmatrix}. \quad (5.215)$$

Llamando D_{42} a la matriz diagonal

$$D_{42} = \begin{pmatrix} 1 & 0 & \cdots & & 0 \\ 0 & (1-\mu)^2 & 0 & \cdots & \vdots \\ \vdots & & (1-\mu)^2 & & \\ & & & \ddots & 0 \\ 0 & \cdots & & 0 & (1-\mu)^2 \end{pmatrix} \quad (5.216)$$

obtenemos que

$$E\{\mathbf{e}_N^{a''}[n](\mathbf{e}_N^{a''}[n])^T\} \approx E\{|e_v^a[n]|^2\}D_{42}, \quad (5.217)$$

y

$$E\{\mathbf{r}'_N[n]\mathbf{r}'_N^T[n]\} = \sigma_r^2 D_{42}. \quad (5.218)$$

A partir de las expresiones anteriores, considerando (5.78) obtenemos para el valor de $E\{|e_v^a[n]|^2\}$

$$E\{|e_v^a[n]|^2\} = \frac{\mu\sigma_r^2 \text{Tr}(E\{D_{42}\Psi_v[n]\})}{2\text{Tr}(E\{D_{41}\Phi_v[n]\}) - \mu\text{Tr}(E\{D_{42}\Psi_v[n]\})}. \quad (5.219)$$

5.6.7. Cálculo de la norma ponderada en el algoritmo multicanal

Calculamos primero la norma ponderada del lado izquierdo de la igualdad. Así:

$$\begin{aligned} & (\tilde{\mathbf{w}}[n] + \mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_\Sigma^a[n])^T \Sigma (\tilde{\mathbf{w}}[n] + \mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_\Sigma^a[n]) \\ & = \tilde{\mathbf{w}}^T[n]\Sigma\tilde{\mathbf{w}}[n] + \tilde{\mathbf{w}}^T[n]\Sigma\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_\Sigma^a[n] \\ & \quad + (\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_\Sigma^a[n])^T \Sigma (\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_\Sigma^a[n]) \\ & \quad + (\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_\Sigma^a[n])^T \Sigma \tilde{\mathbf{w}}[n] \end{aligned} \quad (5.220)$$

Considerando que las matrices $\mathbf{U}[n]$ y Σ son invertibles, entonces se cumple que:

$$(\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1} = \mathbf{U}^{-1}[n]\Sigma^{-1}(\mathbf{U}^T[n])^{-1} \quad (5.221)$$

$$((\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1})^T = \mathbf{U}^{-1}[n](\Sigma^T)^{-1}(\mathbf{U}^T[n])^{-1}. \quad (5.222)$$

Teniendo en cuenta las relaciones definidas en (A.5) y (A.6), la expresión (5.220) puede escribirse como:

$$\|\tilde{\mathbf{w}}[n]\|_{\Sigma}^2 + \tilde{\mathbf{w}}^T[n]\Sigma\tilde{\mathbf{w}}[n-1] + (\mathbf{e}_{\Sigma}^a[n])^T((\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1})^T\mathbf{e}_{\Sigma}^a[n] + \tilde{\mathbf{w}}^T[n-1]\Sigma\tilde{\mathbf{w}}[n]. \quad (5.223)$$

Análogamente para el lado derecho de la igualdad se obtiene:

$$\begin{aligned} & \|(\tilde{\mathbf{w}}[n-1] + \mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_{\Sigma}^p[n])\|_{\Sigma}^2 \\ &= \tilde{\mathbf{w}}^T[n-1]\Sigma\tilde{\mathbf{w}}[n-1] + \tilde{\mathbf{w}}^T[n-1]\Sigma\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_{\Sigma}^p[n] \\ & \quad + (\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_{\Sigma}^p[n])^T\Sigma(\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_{\Sigma}^p[n]) \\ & \quad + (\mathbf{U}[n](\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1}\mathbf{e}_{\Sigma}^p[n])^T\Sigma\tilde{\mathbf{w}}[n-1] \\ &= \|\tilde{\mathbf{w}}[n-1]\|_{\Sigma}^2 + \tilde{\mathbf{w}}^T[n-1]\Sigma\tilde{\mathbf{w}}[n] + (\mathbf{e}_{\Sigma}^p[n])^T((\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1})^T\mathbf{e}_{\Sigma}^p[n] \\ & \quad + \tilde{\mathbf{w}}^T[n]\Sigma\tilde{\mathbf{w}}[n-1]. \end{aligned} \quad (5.224)$$

Igualando (5.223) y (5.224) se obtiene

$$\begin{aligned} & \|\tilde{\mathbf{w}}[n]\|_{\Sigma}^2 + (\mathbf{e}_{\Sigma}^a[n])^T((\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1})^T\mathbf{e}_{\Sigma}^a[n] \\ & \quad + \tilde{\mathbf{w}}^T[n]\Sigma\tilde{\mathbf{w}}[n-1] + \tilde{\mathbf{w}}^T[n-1]\Sigma\tilde{\mathbf{w}}[n] \\ &= \|\tilde{\mathbf{w}}[n-1]\|_{\Sigma}^2 + (\mathbf{e}_{\Sigma}^p[n])^T((\mathbf{U}^T[n]\Sigma\mathbf{U}[n])^{-1})^T\mathbf{e}_{\Sigma}^p[n-1] \\ & \quad + \tilde{\mathbf{w}}^T[n-1]\Sigma\tilde{\mathbf{w}}[n] + \tilde{\mathbf{w}}^T[n]\Sigma\tilde{\mathbf{w}}[n-1], \end{aligned} \quad (5.225)$$

y teniendo en cuenta que Σ es una matriz simétrica, nos conduce a la igualdad expresada en (A.10).

5.6.8. Justificación del valor aproximado de $E\{(\mathbf{V}^T[n]\mathbf{U}[n])\}$

Tratemos de obtener el valor de la expresión $E\{(\mathbf{V}^T[n]\mathbf{U}[n])\}$, siendo $\mathbf{V}[n]$, la matriz definida en (5.92) y $\mathbf{U}[n]$ la matriz definida en (5.89).

El primer elemento de la diagonal del producto matricial del cual se pretende calcular el valor medio, sería:

$$\sum_{j=1}^J \sum_{i=1}^I \mathbf{V}_{ij1}^T[n]\mathbf{V}_{ij1}[n](\mathbf{V}_{ij1}^T[n]\mathbf{V}_{ij1}[n] + \delta\mathbf{I})^{-1} \quad (5.226)$$

que en el caso de $\delta \approx 0$, resulta la suma de IJ veces la matriz identidad tamaño $N \times N$.

Para el resto de valores de la diagonal, de forma análoga se obtendría el mismo resultado, ya que en la posición genérica (k, k) del producto matricial se obtendría:

$$\sum_{j=1}^J \sum_{i=1}^I \mathbf{V}_{ijk}^T[n] \mathbf{V}_{ijk}[n] (\mathbf{V}_{ijk}^T[n] \mathbf{V}_{ijk}[n] + \delta \mathbf{I})^{-1} \quad (5.227)$$

Por otro lado, para las posiciones fuera de la diagonal (por ejemplo, en la (k_1, k_2)), se obtendría:

$$\sum_{j=1}^J \sum_{i=1}^I \mathbf{V}_{ijk_1}^T[n] \mathbf{V}_{ijk_2}[n] (\mathbf{V}_{ijk_2}^T[n] \mathbf{V}_{ijk_2}[n] + \delta \mathbf{I})^{-1}. \quad (5.228)$$

Al calcular el valor medio de dichas expresiones, si consideramos independencia estadística de los datos de las matrices $\mathbf{V}_{ijk_1}[n]$ y $\mathbf{V}_{ijk_2}[n]$ para $k_1 \neq k_2$, teniendo en cuenta que contienen variables aleatorias de media cero, dichos términos valdrían cero. Durante el desarrollo seguido para obtener el modelo multicanal, se han despreciado los términos fuera de la diagonal para poder obtener una expresión compacta, y aunque la suposición de independencia no se cumpla, la aproximación resulta apropiada.

Capítulo 6

Resultados de simulación

6.1. Introducción

En este capítulo, se pretende mostrar el comportamiento de algunos de los algoritmos estudiados en esta tesis doctoral bajo diferentes regímenes de funcionamiento, siendo así fácilmente comparables sus prestaciones. Básicamente, las simulaciones se han centrado en el estudio del algoritmo de proyección afín con estructura de filtrado-x convencional (etiquetado en las gráficas como *fxap*), comparándolo con los algoritmos de proyección afín que usan estructura de filtrado-x modificada (*mfxap*) así como con el algoritmo LMS (con estructura de filtrado-x convencional (*fxlms*) y modificada (*mfxlms*) habitualmente usadas como versiones de referencia.

Aunque comparar diferentes algoritmos entre sí (controlados por parámetros distintos y por tanto, dependientes de los mismos) no resulta evidente, se han realizado varias simulaciones mostrando el comportamiento individual y comparativo de estos algoritmos manteniendo fijos la mayoría de parámetros comunes de forma que la diferencia de comportamiento sea la debida a la diferencia de funcionamiento del algoritmo y no a la diferente selección de los parámetros que lo controlan. También se han realizado simulaciones con los algoritmos funcionando a su mayor velocidad de convergencia, de forma que se pueda apreciar el comportamiento de los mismos cuando estos trabajan en el límite de sus prestaciones (máximo parámetro de convergencia μ posible).

Las simulaciones se han realizado para distintos sistemas multicanal 1:2:2, con diferentes señales de ruido, con la ayuda del simulador desarrollado para este trabajo descrito en el anexo [D](#).

6.2. Comparativa genérica

En esta primera comparativa se va a mostrar el funcionamiento de los algoritmos de proyección afín para distintos órdenes de proyección ($N = 2, 3, 5$ y 10), manteniendo el parámetro de convergencia μ constante en todos ellos. Además, se mostrarán, como referencia, las curvas de convergencia de los algoritmos LMS, tanto con la estructura de filtrado-x modificada como con la convencional. La elección del paso de convergencia para los algoritmos de proyección se ajustó al más restrictivo de entre todos los algoritmos (*fxap* con $N = 10$) mediante prueba y error, escogiendo el valor que conseguía la mayor velocidad de convergencia sin divergencia (resultando $\mu = 0.02$) y usando dicha constante en todas las simulaciones de los algoritmos de proyección afín para los distintos órdenes de proyección y diferentes estructuras. Se escogió una constante de regularización $\delta = 10^{-4}$, y los filtros adaptativos fueron de 50 coeficientes tanto para los algoritmos AP como para los LMS. Para estos últimos, el parámetro de convergencia tuvo que rebajarse hasta $\mu = 5 \cdot 10^{-5}$ para garantizar la convergencia. Se ha usado ruido blanco con función de densidad de probabilidad constante como señal de ruido a cancelar.

La figura 6.1, representa las curvas de convergencia (o curvas de atenuación, calculadas como el cociente en unidades logarítmicas entre la potencia de la señal de error y la potencia de la señal deseada) en cada uno de los dos sensores de error del sistema CAR simulado, para los algoritmos de proyección afín de filtrado-x con estructura modificada y convencional con órdenes de proyección $N = 2$, y $N = 3$ (representando el resto de órdenes en otra figura para mayor claridad), comparadas con los algoritmos *fxlms* y *mfxlms*.

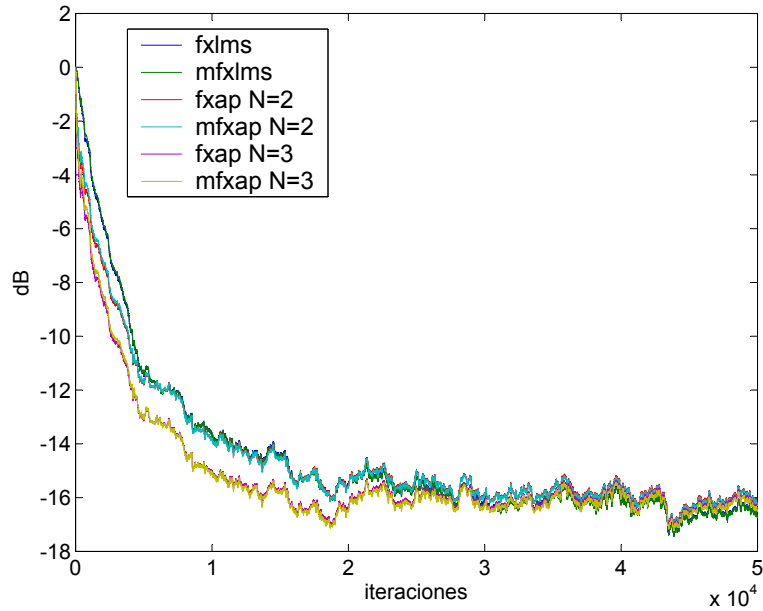
Como se muestra en dicha figura, si usamos el mismo parámetro de convergencia μ , las curvas obtenidas para cada algoritmo son casi idénticas entre sí, independientemente de la estructura usada. Así, las curvas para el algoritmo *fxlms* y *mfxlms* se confunden entre sí, igual que las de los algoritmos AP del mismo orden de proyección y diferentes estructuras. También podemos observar como habiendo usado en este caso un parámetro de convergencia pequeño, las atenuaciones finales (parámetro relacionado con el *MSE*) son muy parecidas para todos los algoritmos (lo que se observa especialmente en los resultados en el sensor 1, figura 6.1(a)). Por otro lado, como cabía esperar, la velocidad de convergencia es mayor conforme el orden de proyección crece.

Para órdenes de proyección más elevados, el comportamiento de los algoritmos sigue la misma tendencia (velocidad de convergencia mayor conforme el orden de proyección crece y mismo comportamiento independientemente de la estructura). La figura 6.2 representa la misma comparativa que la representada en la figura 6.1, pero usando órdenes de proyección 5 y 10 en los algoritmos de proyección afín. Se llegan a observar algunas diferencias en el comportamiento

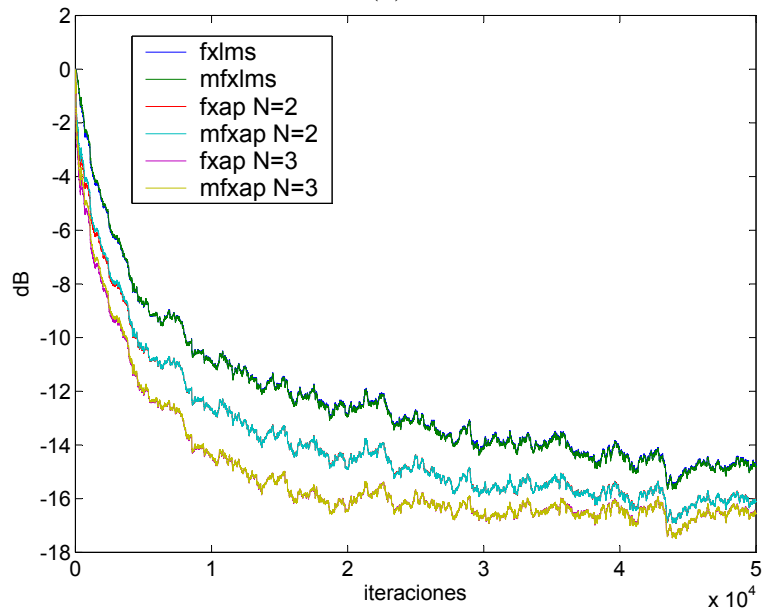
transitorio de los algoritmos de proyección afín de orden 10, más acusadas que en el resto de casos, pero sin ser demasiado significativas.

De los resultados anteriores se demuestra que las propiedades de convergencia son similares para los algoritmos que usan la estructura de filtrado-x modificada y la convencional, si el paso de convergencia que se usa en ambos casos es el mismo. Sin embargo, la estructura modificada surge como una mejora en la velocidad de convergencia de los algoritmos que usan la estructura convencional de filtrado-x y permite usar pasos de convergencia mayores. Podría ser interesante comparar los algoritmos en su régimen de funcionamiento extremo (parámetro de convergencia mayor sin que se produzca divergencia) para poder comprobar esta característica. En las simulaciones presentadas en la figura 6.3 el paso de convergencia ha sido elegido por prueba y error de forma independiente para cada algoritmo, de manera que todos los algoritmos presentan la mayor velocidad de convergencia que pueden alcanzar, pudiéndose observar así, las prestaciones reales de cada algoritmo en cuanto a la velocidad de convergencia.

Aun cuando se intuye un mejor comportamiento en la velocidad de convergencia de los algoritmos que usan la estructura de filtrado-x modificada frente a los que usan la estructura de filtrado-x convencional, tampoco en este caso es demasiado significativo. Estos resultados redundan en los beneficios de usar la estructura de filtrado-x (mucho más eficiente computacionalmente, y sin introducir demasiado perjuicio en la velocidad de convergencia). También es cierto que el verdadero beneficio de la estructura modificada aparece cuando el retardo introducido por los caminos secundarios del sistema acústico es elevado. En estos casos, el parámetro de convergencia usado en los algoritmos que usan la estructura de filtrado-x convencional no puede ser muy grande, ya que existiría un gran retardo entre la señal que se controla (señal de error) y la que se genera para su cancelación. Puesto que dicho retardo desaparece con la estructura modificada, esta estructura permitiría velocidades de convergencia mayores (independientes del retardo de los caminos acústicos). Los resultados de simulación presentados en 6.3 han sido obtenidos para un sistema acústico estándar donde la separación entre sensores de error y actuadores era de unos pocos metros. No obstante, se han realizado otras simulaciones recreando entornos acústicos donde dichos retardos son mayores.



(a)



(b)

Figura 6.1: Niveles de atenuación en sensor 1 (a), y sensor 2 (b), cuando comparamos los diferentes algoritmos bajo el mismo régimen de funcionamiento empleando ruido blanco como señal a cancelar.

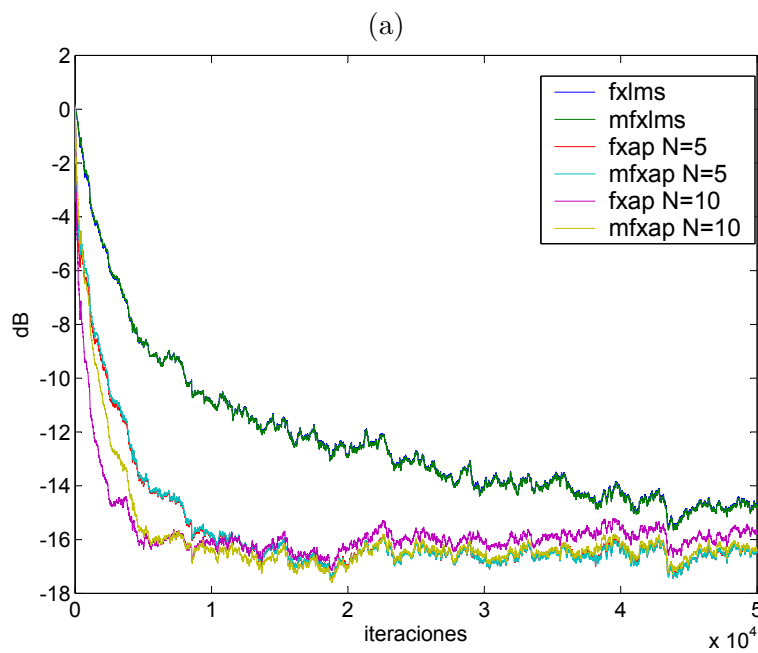
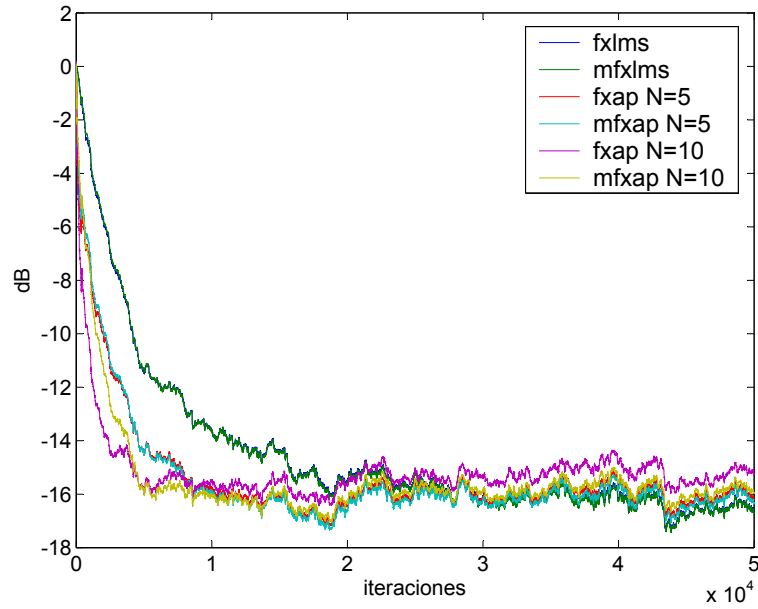
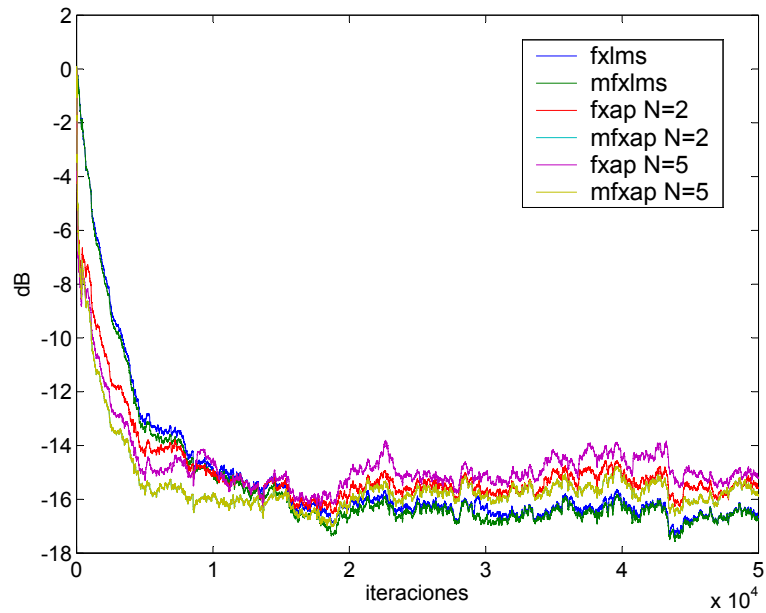
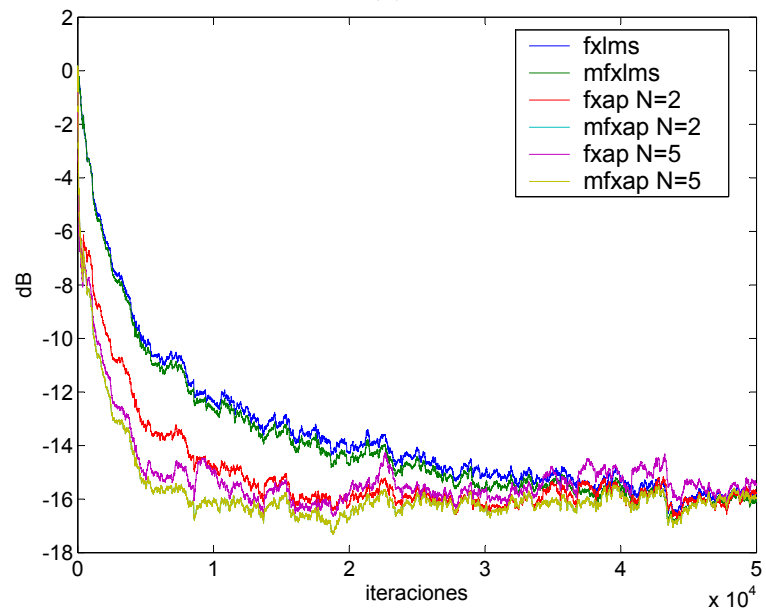


Figura 6.2: Niveles de atenuación en sensor 1 (a), y sensor 2 (b), cuando comparamos los diferentes algoritmos bajo el mismo régimen de funcionamiento empleando ruido blanco como señal a cancelar.



(a)



(b)

Figura 6.3: Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia empleando ruido blanco como señal a cancelar.

6.3. Comparativa para diferentes tipos de ruido

Las simulaciones presentadas en los apartados anteriores se han obtenido usando ruido aleatorio con densidad espectral de potencia constante como señal a cancelar, pero podemos preguntarnos qué pasaría si el ruido tuviera un espectro que no fuera plano. Para analizar estos casos, se han realizado diferentes simulaciones con distintos tipos de ruidos (ruido aleatorio coloreado, ruido periódico y tonos) como señales a cancelar, presentando los resultados de simulación a continuación.

Hay que comentar, que cuando los parámetros que controlan la convergencia se mantienen constantes (como en la simulaciones presentadas en el primer caso) la tendencia del comportamiento de los algoritmos es similar a la comentada con anterioridad independientemente del tipo de ruido usado, es decir, no se aprecian grandes diferencias entre los algoritmos que usan la estructura modificada y los que usan la de filtrado-x convencional (cuando solo se diferencian en la estructura y el resto de parámetros de control es idéntico), mientras que la velocidad de convergencia aumenta con el orden de proyección en los algoritmos de proyección afín y es siempre mayor que en los algoritmos basados en el algoritmo LMS. Comprobemos si el comportamiento es el mismo con los algoritmos funcionando a su máxima velocidad y en qué medidas existen diferencias dependientes de la señal de ruido a cancelar.

En la figura 6.4 se presentan los resultados de la simulaciones cuando el ruido usado es ruido blanco Gaussiano coloreado con un filtro autorregresivo de primer orden (proporcionando un espectro final del ruido paso-bajo). En este caso, el comportamiento es bastante similar al comentado en el caso de ruido blanco (figura 6.3). De nuevo, las estructuras modificadas presentan velocidades de convergencia algo mayores que las de filtrado-x convencional a igualdad de algoritmos, y también crecen conforme el orden de proyección de los algoritmos AP es mayor.

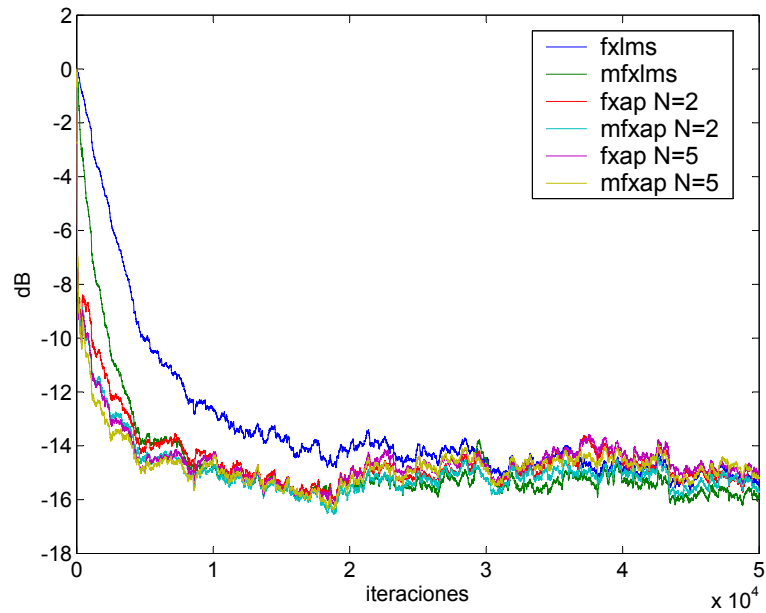
Si usamos una señal periódica (tono de frecuencia discreta 0.1) como señal a cancelar, el comportamiento de los algoritmos es el que se aprecia en la figura 6.5. En este caso, los algoritmos consiguen atenuaciones casi ideales y todos tienen comportamientos parecidos (las diferencias se podrían atribuir más a imprecisiones numéricas (debidas a los niveles tan bajos que se obtienen en las atenuaciones) que al comportamiento propio de los algoritmos).

Una señal periódica (ruido periódico, puesto que las fases iniciales son aleatorias) con contenido espectral más rico (10 tonos con frecuencias discretas separadas 0.02 situados entre las frecuencias 0.02 y 0.2) también ha sido usada como señal a cancelar, proporcionando los resultados representados en la figura 6.6. En este caso, los resultados obtenidos parecen no haber llegado

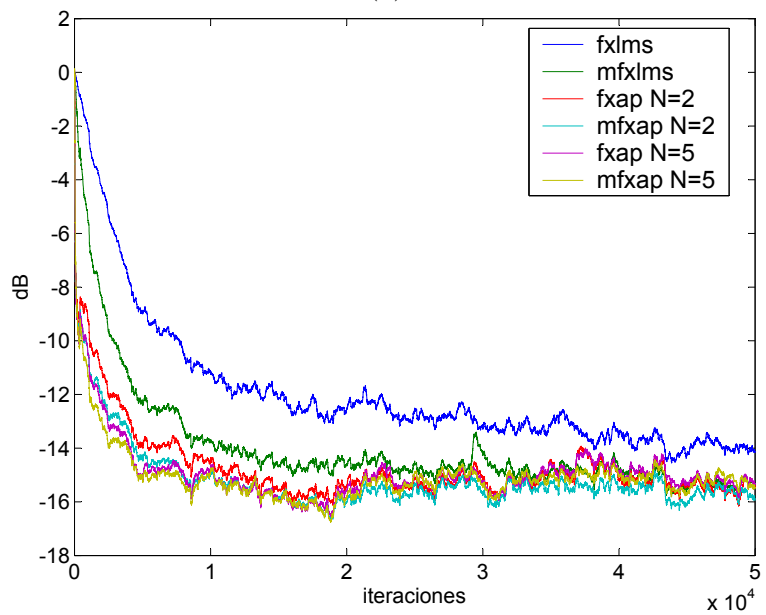
todavía a su régimen permanente, apreciándose diferentes tramos donde la velocidad de convergencia va variando. Sin duda, se debe a que la velocidad con la que cada uno de los tonos se atenúa es diferente, puesto que la potencia con la que llegan a los sensores de error también lo es (están coloreados por los caminos acústicos), además de que las condiciones acústicas del sistema simulado pueden provocar la existencia de alguna resonancia cerca de alguna de las frecuencias de la señal de ruido periódica o que los sensores o actuadores estén colocados en nulos de presión o velocidad acústica, provocando que ciertas frecuencias se cancelen más lentamente (o no se lleguen a cancelar). En cualquier caso, en la figura 6.6 (considerando que toda ella representa régimen transitorio) se aprecia la tendencia más o menos esperada de cada uno de los algoritmos. El único algoritmo que parece no comportarse como en los casos anteriores es el LMS basado en la estructura de filtrado-x modificada. Hay que recordar que el valor del parámetro de convergencia ha sido elegido por prueba y error, procurando que los algoritmos converjan a su mayor velocidad sin divergencia y que dicho algoritmo es el que tolera un paso de convergencia mayor, pero esto implica que el error en régimen permanente sea también mayor. Como se aprecia en la figura 6.6, este algoritmo presenta una velocidad inicial muy grande y luego se ralentiza, debido a que el parámetro de convergencia estaba optimizado para cancelar los tonos de mayor potencia, pero una vez cancelados éstos, el resto de señal se cancela más lentamente puesto que su nivel es muy pequeño en referencia al nivel inicial y el paso de convergencia usado es grande.

6.4. Efecto del retardo de los caminos secundarios en la velocidad de convergencia

Como ya se ha comentado anteriormente, la existencia de retardos en los caminos acústicos secundarios (atribuidos a la separación física existente entre los sensores de error y los actuadores) provoca en los algoritmos que usan la estructura de filtrado-x convencional un desajuste entre la señal que se genera para conseguir la cancelación, y la señal de referencia que es obtenida de los sensores de error, ya que la señal generada no actuará directamente en la señal obtenida en los sensores, si no que tendrá que viajar a través del camino acústico. Esta limitación se traduce en una velocidad de convergencia menor. Para solucionar este problema, surgió la estructura modificada. Así pues, usar la estructura de filtrado-x en los algoritmos de proyección aún puede suponer un beneficio grande en cuanto al coste computacional, pero también podría ser un factor negativo en la velocidad de convergencia.



(a)



(b)

Figura 6.4: Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia, y ruido coloreado como señal a cancelar.

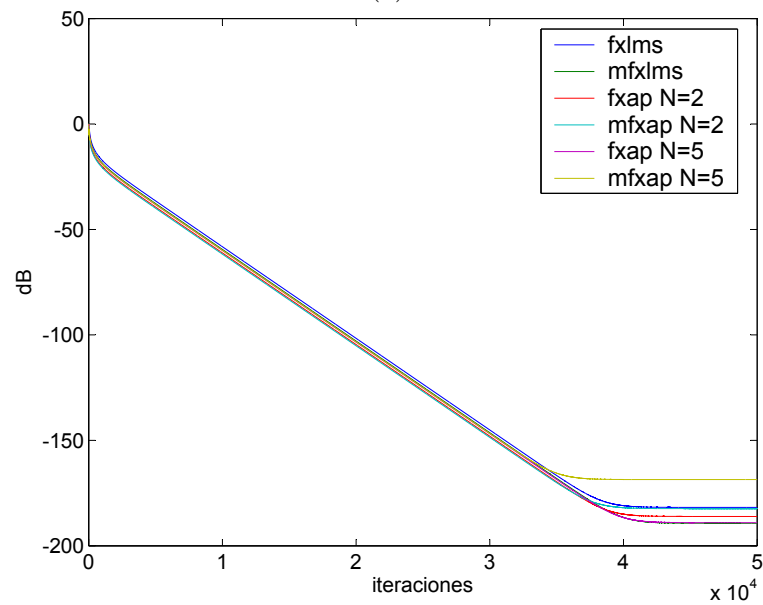
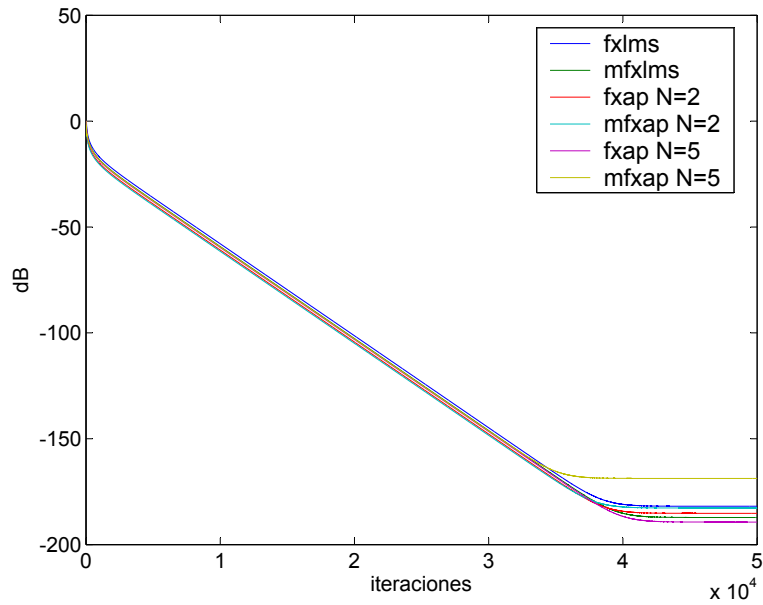
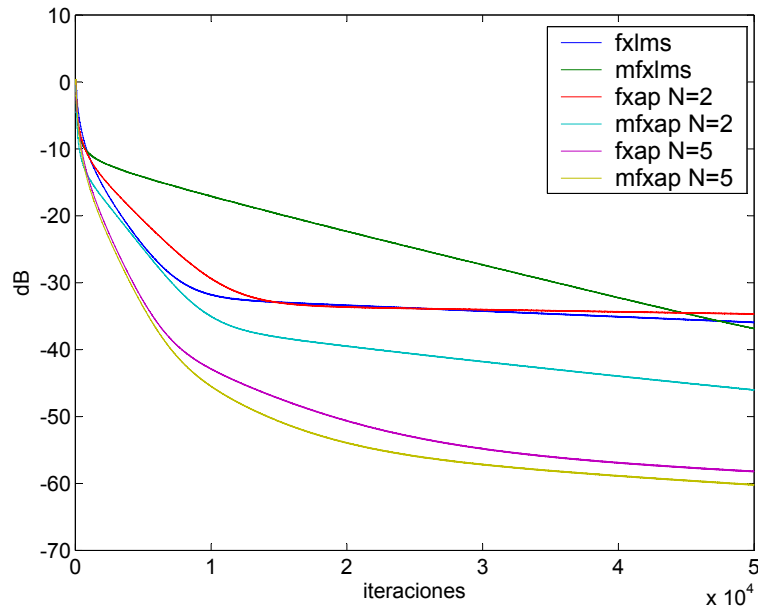
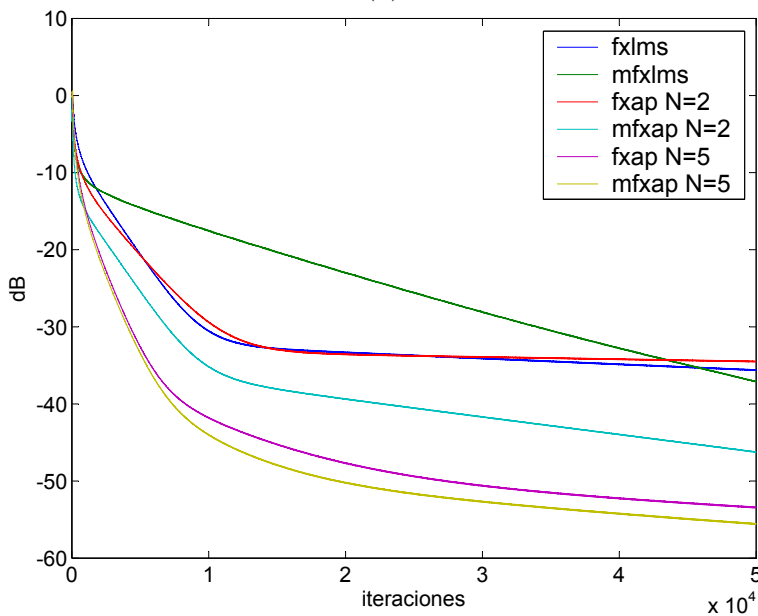


Figura 6.5: Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia, y un tono ($f = 0,1$) como señal de ruido.



(a)



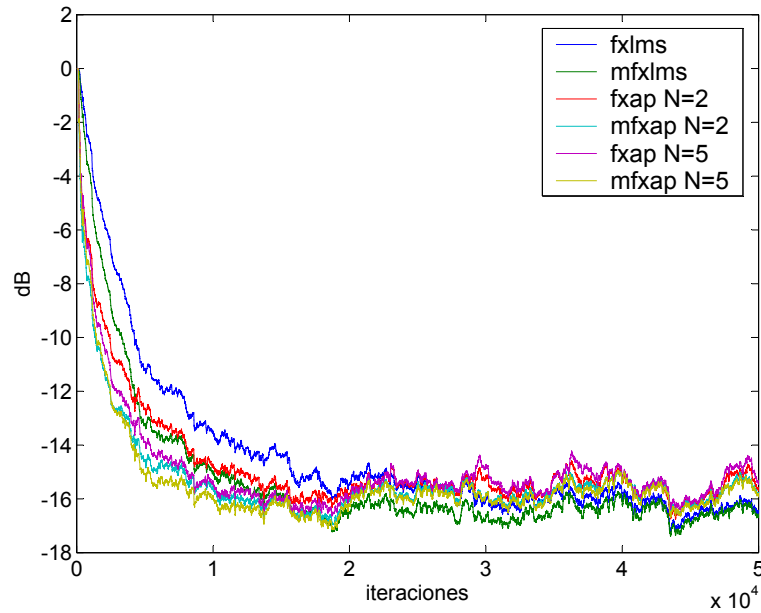
(b)

Figura 6.6: Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia, y señal periódica como señal a cancelar.

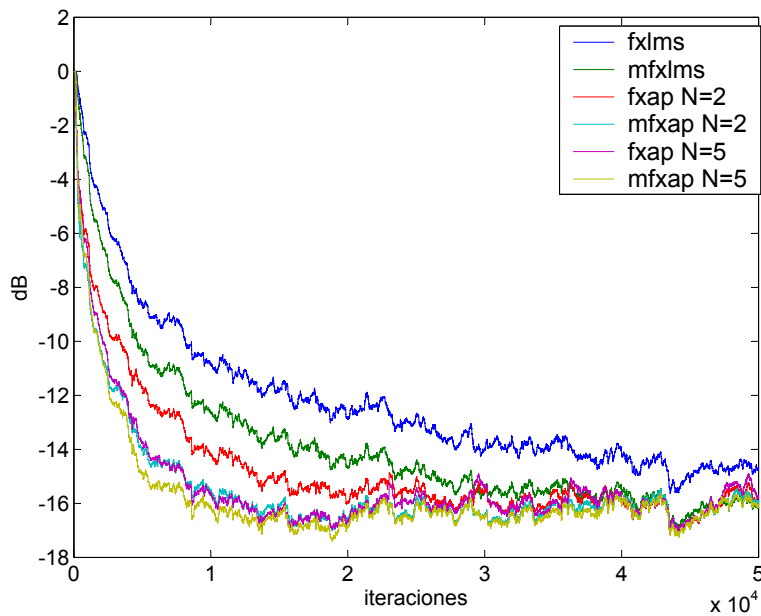
Según se desprende de los resultados alcanzados en las simulaciones comentadas anteriormente, el decremento de la velocidad de convergencia, puede considerarse tolerable para un sistema de control activo de ruido donde los retardos de los caminos acústicos no sean muy grandes (en nuestro sistema simulado, alrededor de unas 10 muestras). En la mayoría de los sistemas prácticos, estos retardos no son demasiado elevados y las prestaciones ofrecidas por los algoritmos con estructura modificada no introducen demasiada ganancia en cuanto a su velocidad de convergencia frente a los resultados ofrecidos por los algoritmos que usan estructura de filtrado-x convencional, pero en todo caso conviene examinar el comportamiento de los algoritmos de proyección afín basados en la estructura de filtrado-x cuando los caminos acústicos introducen retardos de mayor duración.

Así, se ha añadido a los caminos acústicos usados en las simulaciones retardos adicionales para hacerlos más largos, pudiéndose observar el deterioro que sufre la velocidad de convergencia en los algoritmos basados en la estructura de filtrado-x convencional.

En la figura 6.7 se representan las curvas de convergencia para los mismos algoritmos usados en las simulaciones anteriores y un retardo de los caminos acústicos de 50 muestras (que puede ser considerado como largo, teniendo en cuenta que es del tamaño de los filtros adaptativos). En este caso puede observarse como los algoritmos con estructuras de filtrado-x modificada presentan velocidades de convergencia mayores que sus equivalentes con estructura de filtrado-x convencional, lo que no nos sorprende porque ya sucedía con los caminos acústicos originales, aunque en este caso se puede apreciar una leve ganancia de la velocidad de convergencia de las estructuras modificadas respecto a las convencionales de filtrado-x. Dentro de lo que cabía esperar, los algoritmos basados en las estructuras de filtrado-x convencional, sufren una disminución en su velocidad de convergencia al aumentar el retardo de los caminos acústicos, pero aun así, las prestaciones de los mismos siguen siendo bastante buenas. Si triplicamos el retardo (retardo introducido de 150 muestras), el detrimento en la velocidad de convergencia sí es significativo, puesto que la peor de las velocidades de convergencia de los algoritmos basados en la estructura de filtrado-x modificada sigue siendo más rápida que cualquiera de las velocidades de convergencia de los algoritmos basados en la estructura de filtrado-x convencional (figura 6.8). En estos casos extremos, el uso de la estructura modificada está justificado si la velocidad de convergencia es un parámetro crítico y los algoritmos de proyección afín basados en la estructura de filtrado-x no ofrecerían mejores prestaciones en cuanto a su velocidad de convergencia que la que proporciona el algoritmo LMS con estructura modificada (*mfxlms*).



(a)



(b)

Figura 6.7: Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia y caminos secundarios con retardo adicional de 50 muestras, empleando ruido blanco como señal a cancelar.

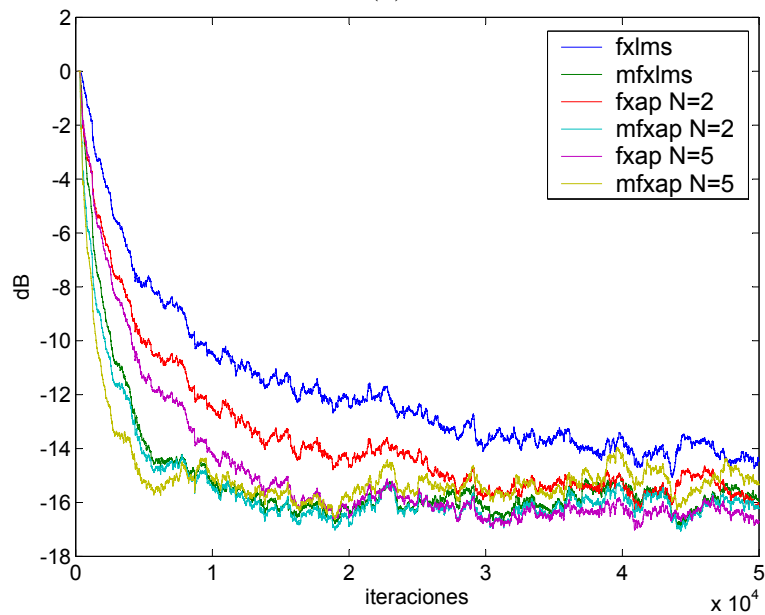
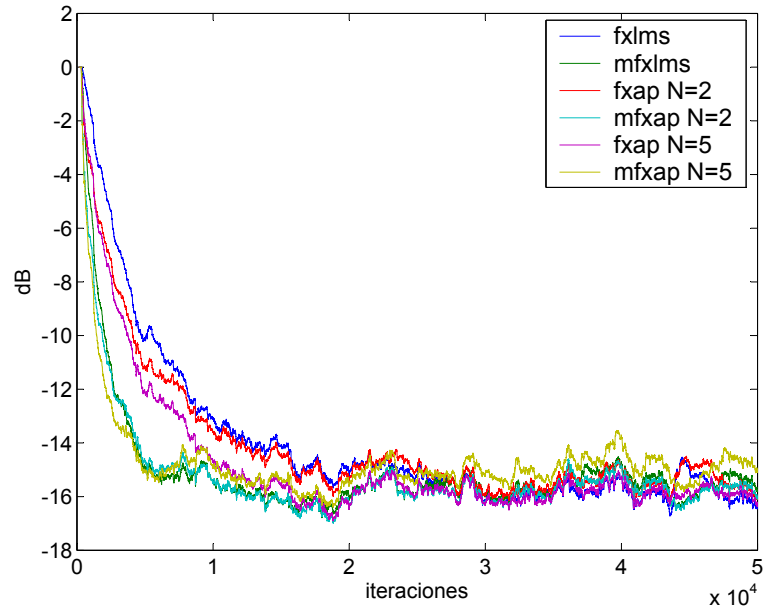


Figura 6.8: Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia y caminos secundarios con retardo adicional de 150 muestras, empleando ruido blanco como señal a cancelar.

Capítulo 7

Implementación práctica del algoritmo de proyección afín

7.1. Introducción

Puesto que el objetivo final de todo desarrollo algorítmico es poder ser implementado en un sistema real, se ha desarrollado un sistema de control activo de ruido multicanal basado en una plataforma DSP donde se han programado algunos de los diferentes algoritmos desarrollados en este trabajo. De esta forma podemos corroborar que los algoritmos desarrollados cumplen perfectamente la misión para la que fueron concebidos, y obtener medidas reales de sus capacidades realizando todo tipo de experimentación acerca de su funcionalidad. El sistema experimental real se ha instalado en una sala acondicionada acústicamente, realizando múltiples medidas con diferentes señales de ruido, tanto de las prestaciones de los diferentes algoritmos programados, como de la estabilidad y robustez de los mismos.

7.2. Descripción del prototipo y sistema de medidas

El sistema de control activo usado, ha sido un sistema multicanal con dos sensores de error y dos actuadores. En la figura 7.1 puede verse un gráfico esquematizado de la disposición del sistema acústico del prototipo y de las medidas de la sala, mientras que en 7.2 se muestran fotos del sistema. Como sensores de error se han usado micrófonos capacitivos (electret) de bajo coste (ES1 y ES2 en la figura 7.1), y como actuadores, altavoces autoamplificados de alta fidelidad, modelo Event 20/20 bas (SS1 y SS2). Para generar el ruido acústico a cancelar por el sistema, se ha usado otro altavoz del mismo

tipo al cual se le inyecta una señal procedente de un generador de funciones. Las señales recogidas por los micrófonos constituían las entradas del sistema controlador DSP, cuyas salidas estaban conectadas a los altavoces. Para monitorizar la incidencia del sistema en un supuesto usuario colocado en la zona de control, se ha usado un maniquí (más concretamente el modelo B&K Type 4100) compuesto de cabeza y torso con micrófonos de alta calidad en los pabellones auditivos simulados (MS_I y MS_D).

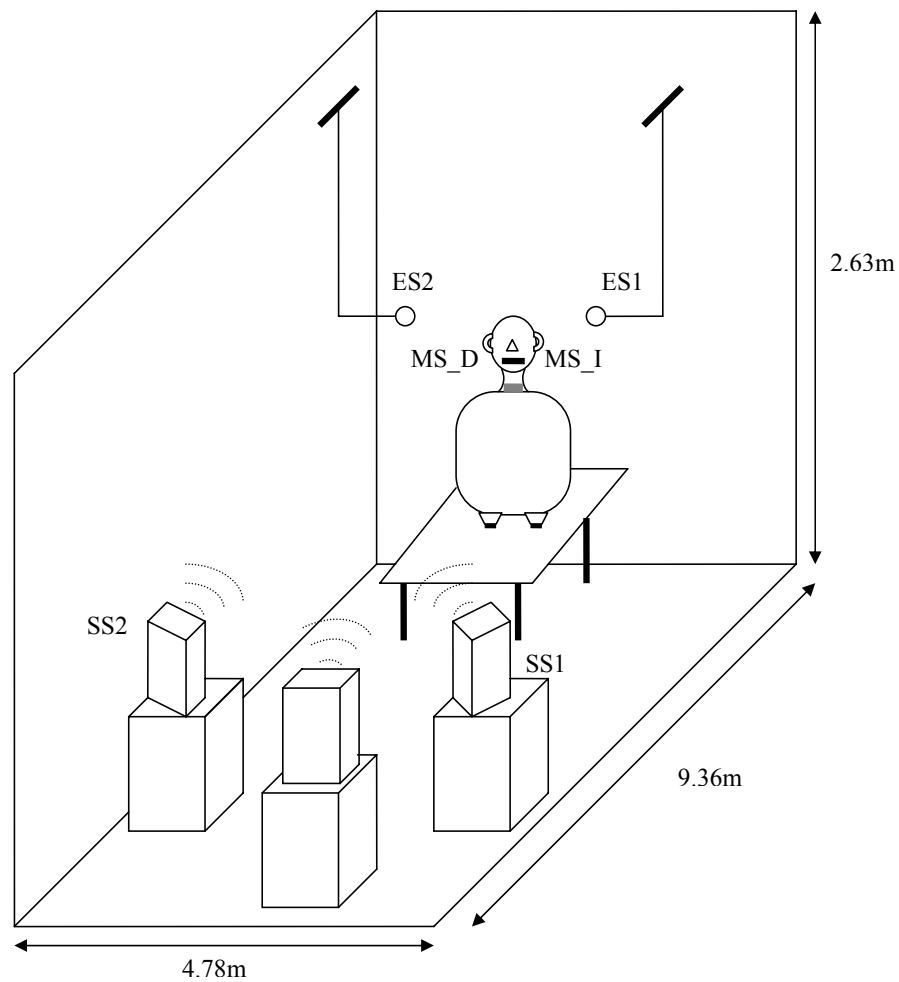


Figura 7.1: Esquema del sistema de CAR experimental.

Considerando que la sala podría aproximarse a un prisma rectangular cuyas dimensiones son $4.78\text{m} \times 9.36\text{m} \times 2.63\text{m}$, podríamos estimar las frecuencias de resonancia de la misma, que vendrían dadas por la expresión:

$$f_{lmr} = \frac{c}{2\pi} \sqrt{\left(\frac{l\pi}{L_x}\right)^2 + \left(\frac{m\pi}{L_y}\right)^2 + \left(\frac{r\pi}{L_z}\right)^2} \quad (7.1)$$

siendo L_x, L_y y L_z las dimensiones del recinto, l, m , y r números enteros (que determinarían el modo de resonancia de la frecuencia en cuestión), y c la velocidad del sonido (346 m/s). Para las dimensiones del recinto, las primeras frecuencias de resonancia así como el modo al que pertenecen se expresan en la tabla 7.1.

En cuanto al sistema electrónico, los algoritmos de control han sido programados en la plataforma DSP de Texas Instrument TMS320C6713 DSK [85], basada en el DSP de punto flotante C6713 [86]. Este dispositivo está capacitado para trabajar a 225 MHz ejecutando hasta 1800 millones de instrucciones por segundo (MIPs), permitiendo computación en paralelo gracias a sus ocho unidades de ejecución compuestas de seis unidades aritmético-lógicas y dos unidades multiplicadoras. Puesto que dicha plataforma solo dispone de un codec estéreo de audio (lo que no es suficiente para desarrollar un sistema multicanal de CAR), se ha usado una tarjeta de audio preparada para este DSK (ADK6713 [87]) que proporciona dos entradas y cuatro salidas estéreo analógicas (un total de 4 canales de audio de entrada y 8 de salida que usan convertidores A/D-D/A de 24 bits y tecnología sigma-delta con frecuencias de muestreo programables entre 48 kHz, 96 kHz y 192 kHz). Los algoritmos programados en el DSP se ejecutan en una rutina de atención a una interrupción software controlada por un doble *buffer* de entrada-salida relleno por los datos procedentes de los conversores. Así, cuando los *buffers* se completan de datos, la rutina toma el control de la ejecución, por lo que la frecuencia de trabajo a la cual se ejecuta el algoritmo depende del tamaño de dichos *buffers* aunque la frecuencia de muestreo de los conversores fue fijada a 48 kHz. Puesto que la frecuencia de trabajo siempre es menor que la frecuencia de muestreo, es necesario usar filtros antialiasing y de reconstrucción de frecuencias de corte ajustadas a la mitad de la frecuencia de trabajo de los algoritmos, para lo cual se usaron filtros paso bajo digitales programados en plataformas DSP de punto fijo y de bajo coste (como las ADSP2181). En la figura 7.3 puede apreciarse un esquema de dicho sistema electrónico.

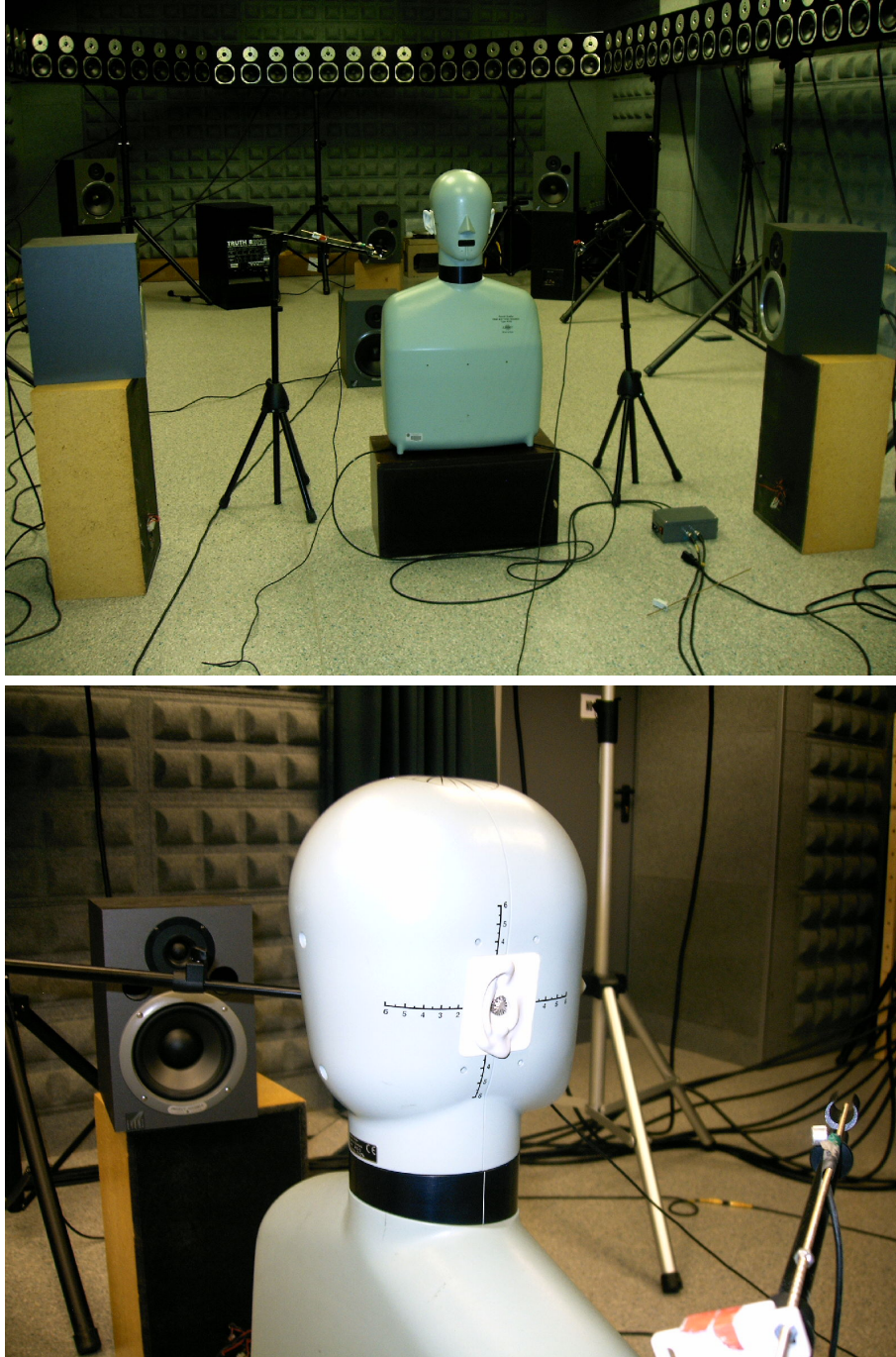


Figura 7.2: Imagen de la sala con el maniquí y el sistema 1:2:2 de CAR y detalle de la cabeza del maniquí y uno de los sensores de monitorización.

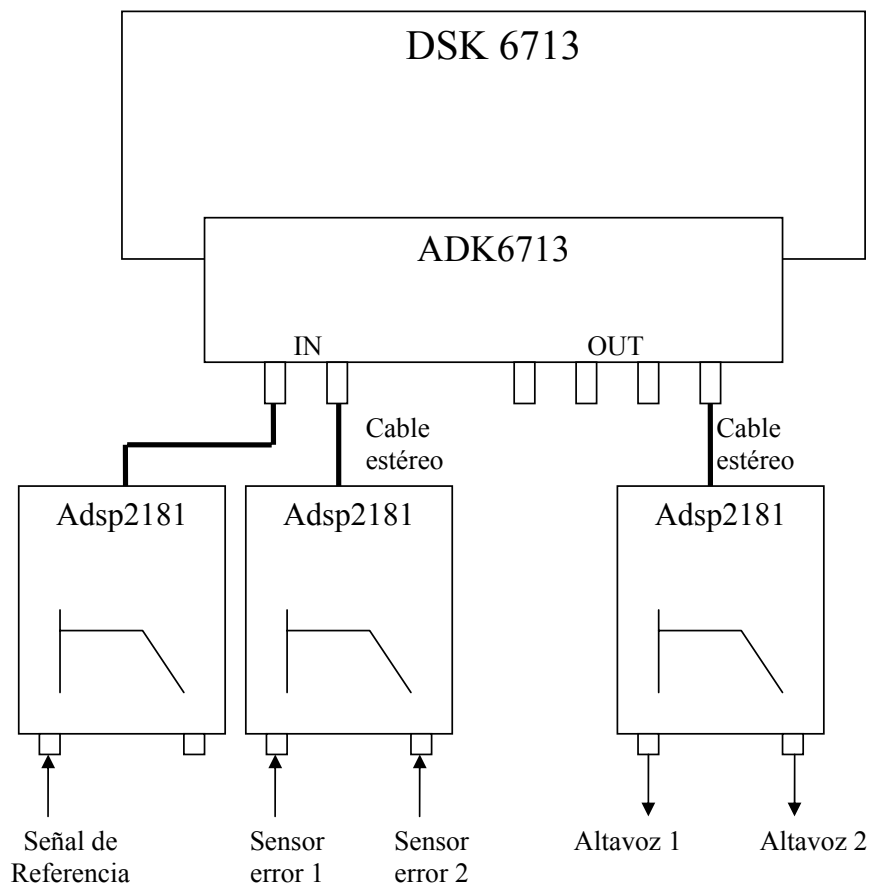


Figura 7.3: Esquema del controlador del sistema experimental de CAR.

Modos de la sala y primeras frecuencias de resonancia

l	m	r	frecuencia(Hz)	l	m	r	frecuencia(Hz)
0	1	0	18.4295	1	0	0	36.0879
0	2	0	36.8590	1	1	0	40.5214
1	2	0	51.5841	0	3	0	55.2885
0	0	1	65.5894	1	3	0	66.0238
0	1	1	68.1294	2	0	0	72.1757
0	4	0	73.7179	2	1	0	74.4915
1	0	1	74.8619	0	2	1	75.2366
1	1	1	77.0970	2	2	0	81.0427
1	4	0	82.0772	1	2	1	83.4439
0	3	1	85.7833	2	3	0	90.9184
1	3	1	93.0651	2	0	1	97.5259
0	4	1	98.6727	2	1	1	99.2519
2	4	0	103.1682	2	2	1	104.2587
1	4	1	105.0649	3	0	0	108.2636
3	1	0	109.8210	2	3	1	112.1076
3	2	0	114.3660	3	3	0	121.5641
2	4	1	122.2523	3	0	1	126.5819
3	1	1	127.9164	3	4	0	130.9784
0	0	2	131.1787	3	2	1	131.8391
0	1	2	132.4670	1	0	2	136.0521
0	2	2	136.2587	1	1	2	137.2947
3	3	1	138.1296	1	2	2	140.9566
0	3	2	142.3540	4	0	0	144.3515
4	1	0	145.5232	3	4	1	146.4831
1	3	2	146.8571	4	2	0	148.9830
2	0	2	149.7237	0	4	2	150.4732
2	1	2	150.8537	2	2	2	154.1939
4	3	0	154.5774	1	4	2	154.7402
4	0	1	158.5538	2	3	2	159.6213
4	1	1	159.6213	4	4	0	162.0854

Tabla 7.1: Primeras frecuencias de resonancia

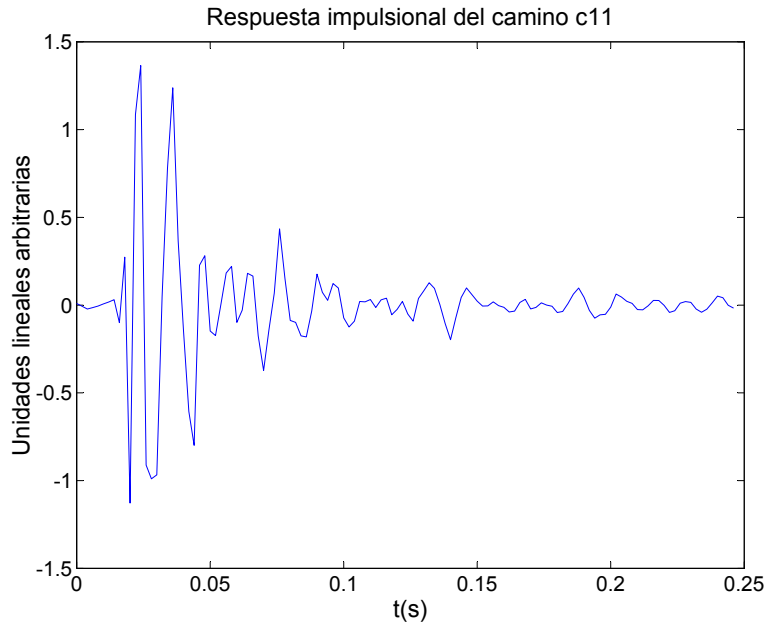
7.3. Análisis de las medidas

El algoritmo analizado en las medidas experimentales ha sido el algoritmo de proyección afín con estructura de filtrado-x (FXAP) donde se usa el método recursivo descrito en 3.4.1 para calcular la inversión matricial. Este algoritmo ha sido programado también con la variante eficiente usando los coeficientes auxiliares (FXAP-CA)(descritos en 3.5.2). Además, para contrastar los resultados obtenidos por estos algoritmos, se ha programado el algoritmo de filtrado-x LMS (FXLMS) normalizado, usado siempre como referencia en aplicaciones de control activo de ruido.

Todos estos algoritmos, han sido desarrollados y testeados en su versión multicanal. Las estimaciones de los caminos secundarios ente cada sensor de error y cada actuador secundario introducidas en los algoritmos se calcularon mediante un identificador de canal LMS, usando 124 coeficientes para su representación. En las figuras 7.4,7.5,7.6 y 7.7 se pueden apreciar gráficamente las características de dichos canales acústicos.

El análisis frecuencial realizado excluye las bajas frecuencias (de 0 a 40 Hz) donde los altavoces no presentaban un comportamiento lineal.

Se usaron diferentes señales de ruido para contrastar el buen funcionamiento de los diferentes algoritmos programados (con diferentes órdenes de proyección para los algoritmos de proyección afín), midiendo los resultados obtenidos en régimen permanente en los sensores de error del sistema de CAR, así como la evolución del transitorio medida en la cabeza acústica.



(a)



(b)

Figura 7.4: Características del camino acústico c11 entre el sensor ES1 y el actuador SS1 (a) Respuesta al impulso, (b) Respuesta en frecuencia.

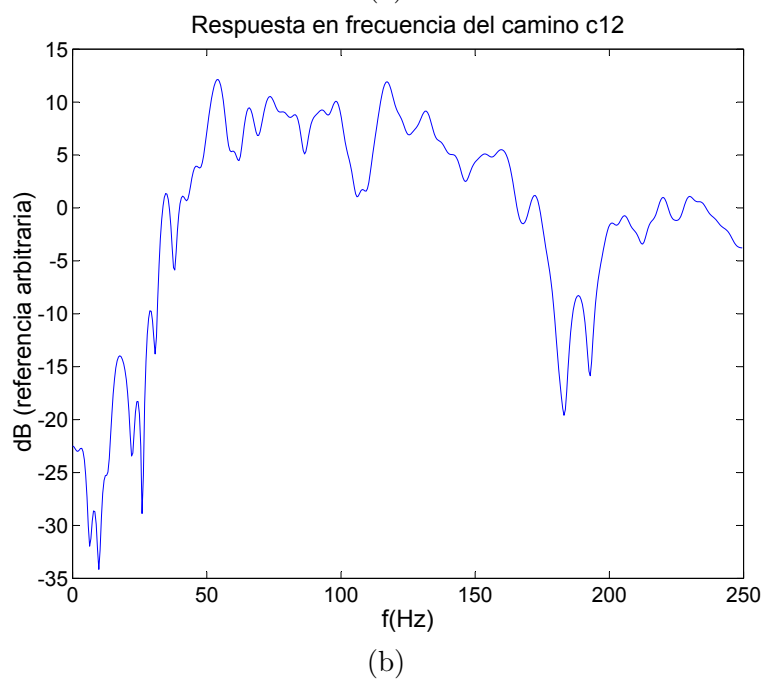
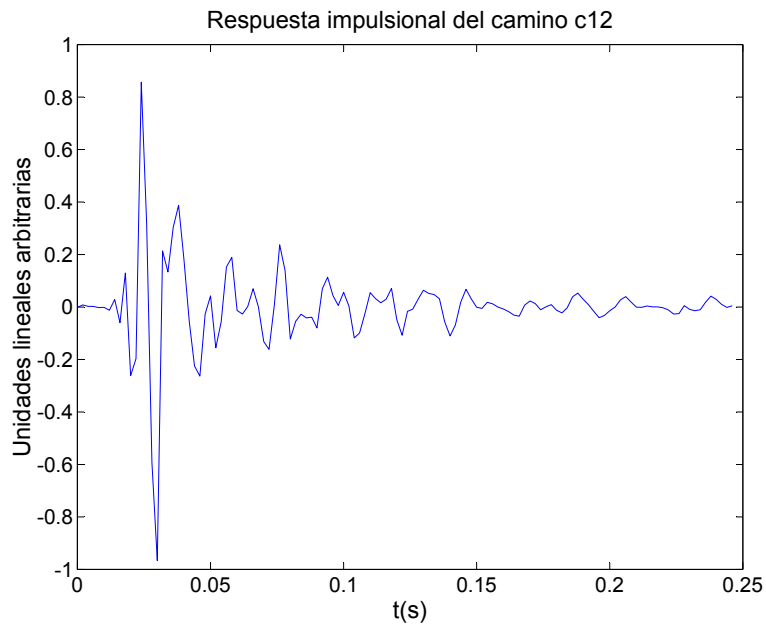


Figura 7.5: Características del camino acústico c12 entre el sensor ES2 y el actuador SS1 (a) Respuesta al impulso, (b) Respuesta en frecuencia.

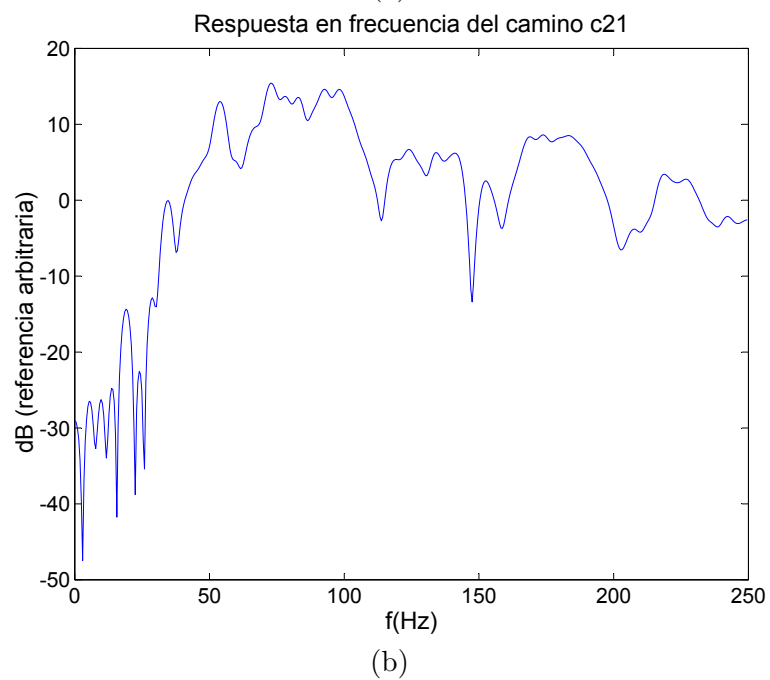
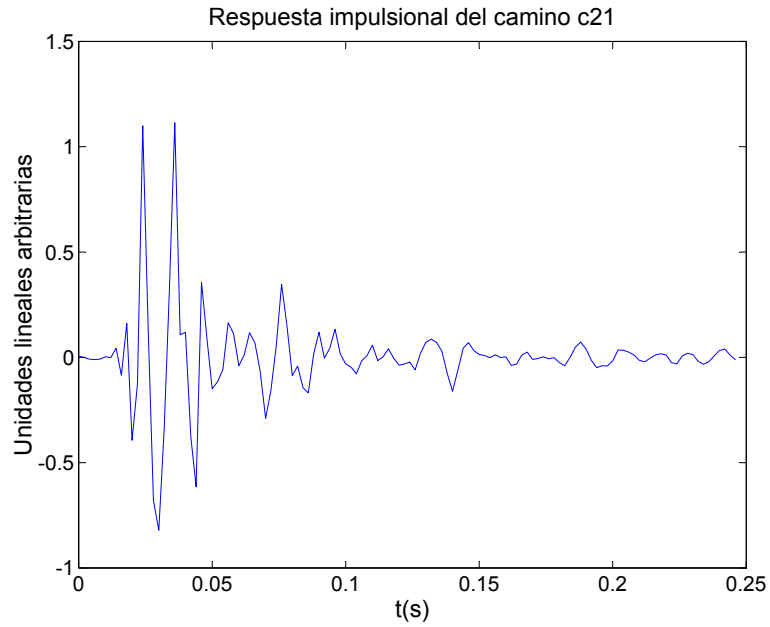


Figura 7.6: Características del camino acústico c21 entre el sensor ES1 y el actuador SS2 (a) Respuesta al impulso, (b) Respuesta en frecuencia.

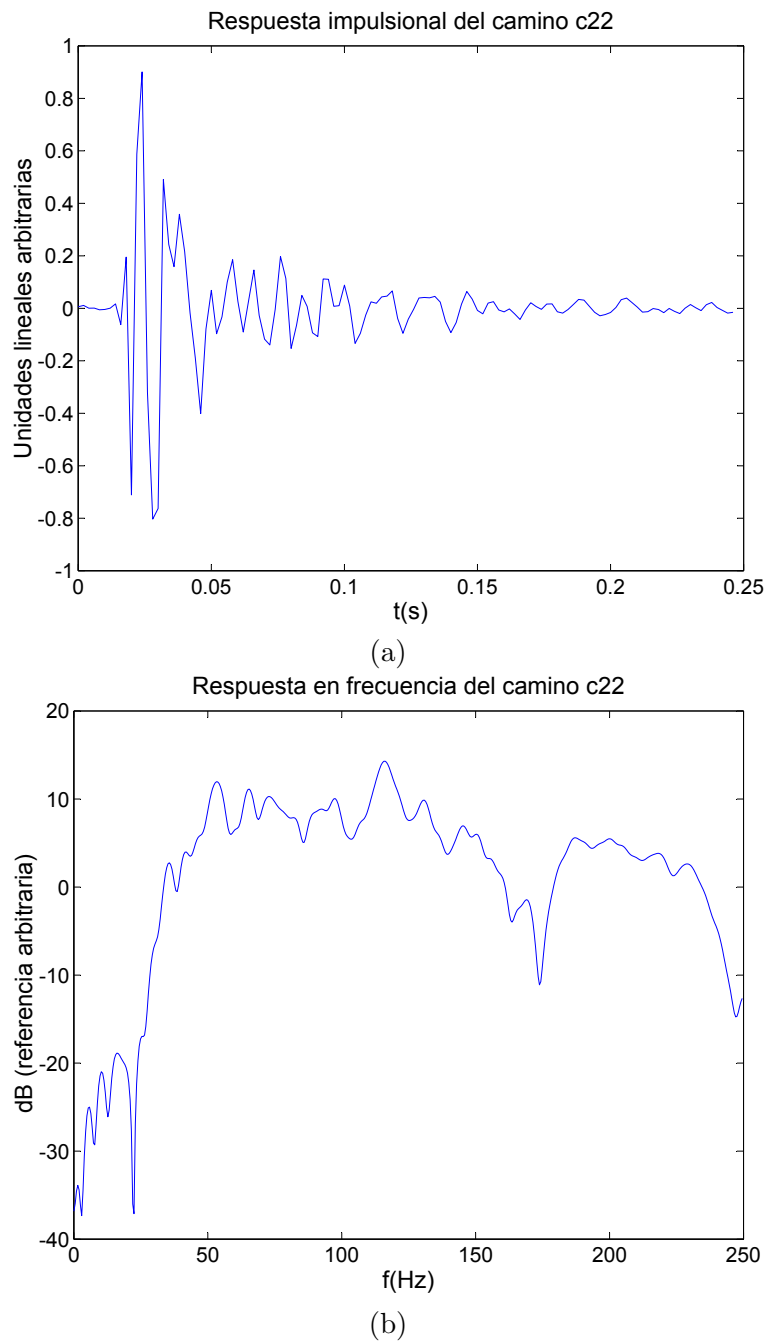


Figura 7.7: Características del camino acústico c22 entre el sensor ES2 y el actuador SS2 (a) Respuesta al impulso, (b) Respuesta en frecuencia.

7.3.1. Ruido aleatorio

Para comprobar el buen funcionamiento de los algoritmos cuando la señal de ruido acústico es de banda ancha, se generó, como señal a cancelar ruido blanco de función de densidad de probabilidad uniforme en la banda de trabajo. Los filtros adaptativos del sistema CAR eran de 120 coeficientes y se usó una constante de convergencia (apropiada para garantizar la convergencia en todos los casos) idéntica para todos los algoritmos ($\mu = 0,002$). La atenuación alcanzada en los sensores de error fue del orden de 12 dB, pudiéndose observar los niveles de las señales en el tiempo y sus espectros, una vez alcanzado el régimen permanente tanto cuando se ejerce control (color verde) como cuando no (color azul) en las figuras 7.8 a 7.17.

Se puede comprobar experimentalmente como los nuevos algoritmos programados funcionan correctamente, y consiguen atenuaciones en los sensores de error similares a las logradas para este tipo de ruido con el algoritmo de filtrado-x normalizado. En la figura 7.18 se pueden comprobar comparativamente los espectros de las señales en régimen permanente para todos los algoritmos estudiados en el experimento.

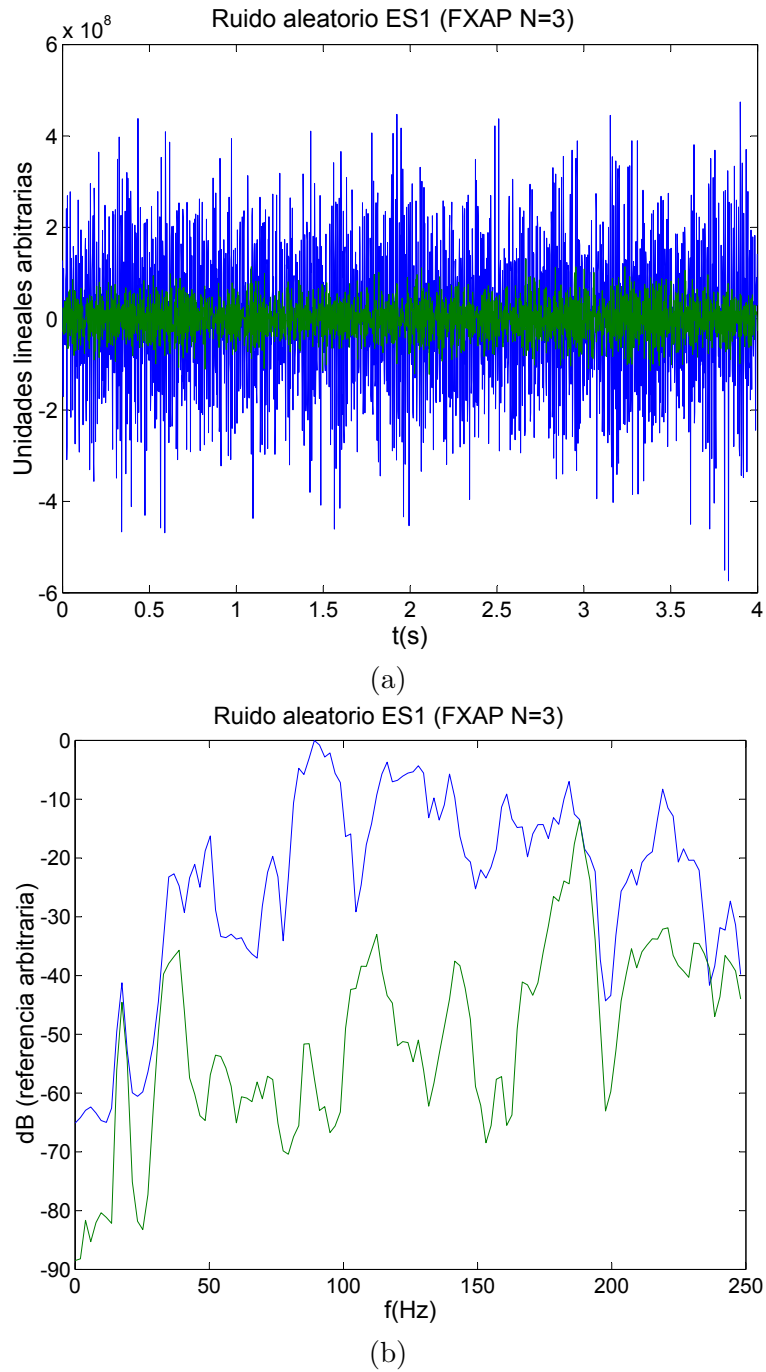
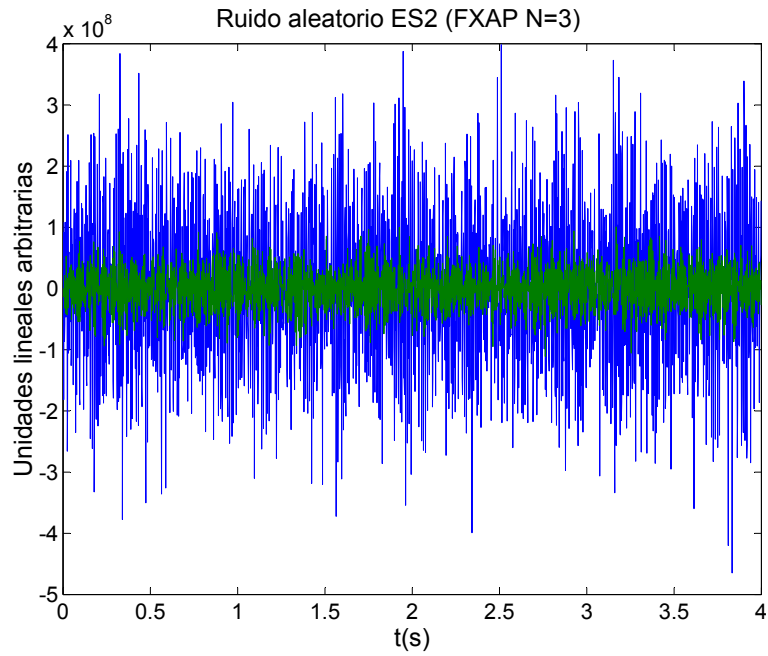
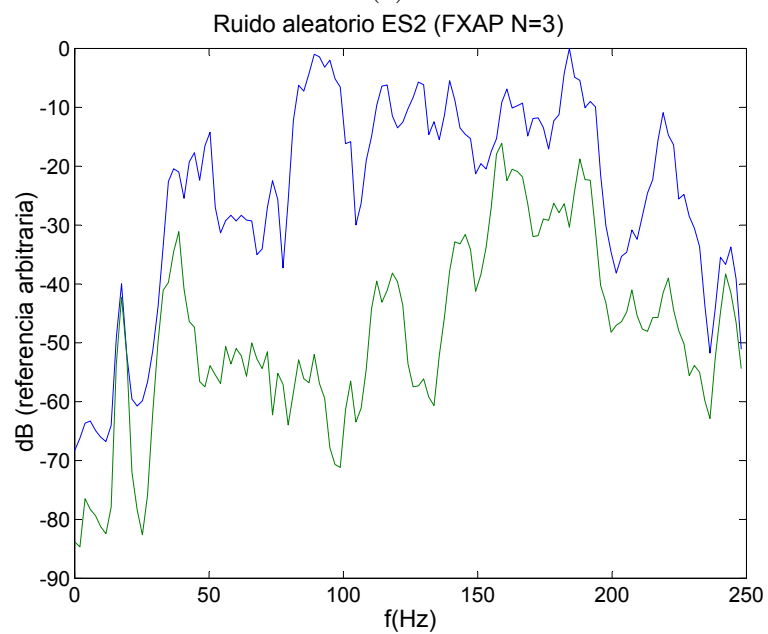


Figura 7.8: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.



(a)



(b)

Figura 7.9: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

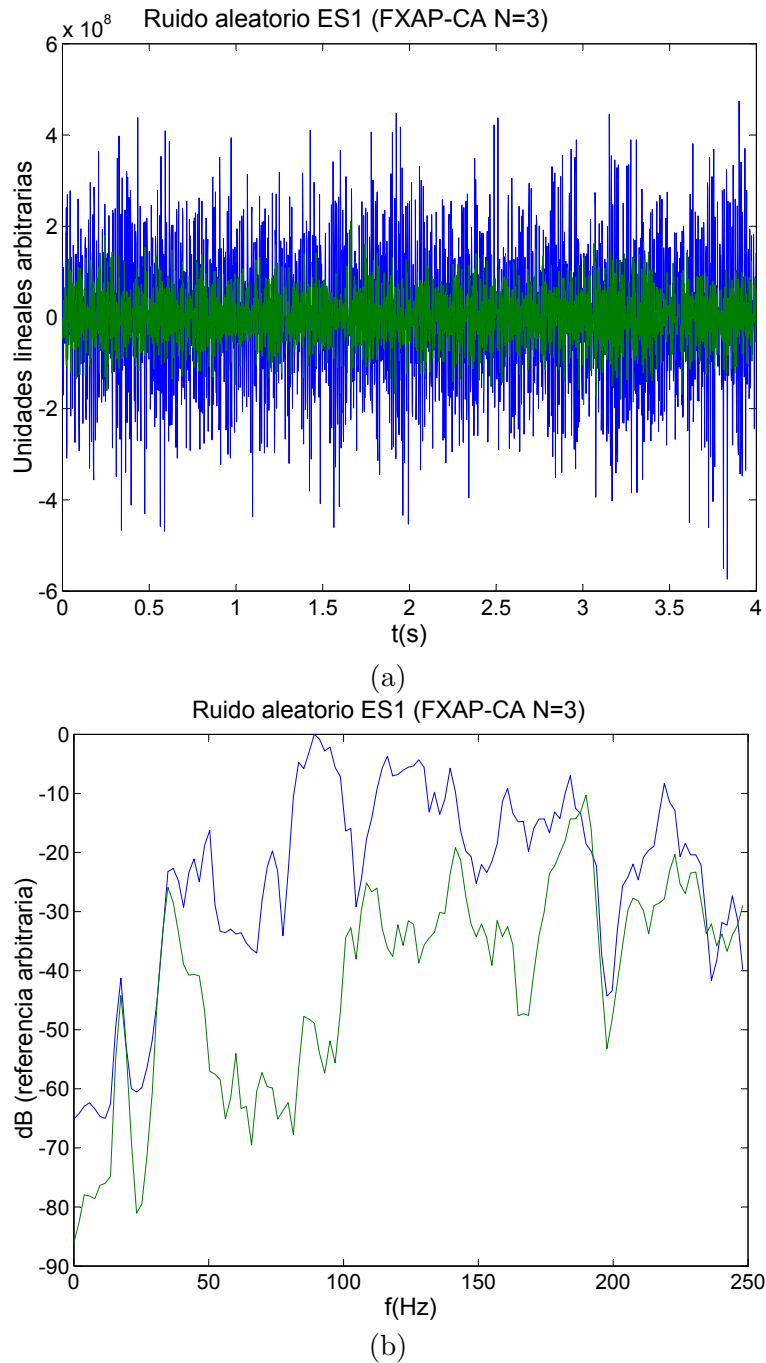


Figura 7.10: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

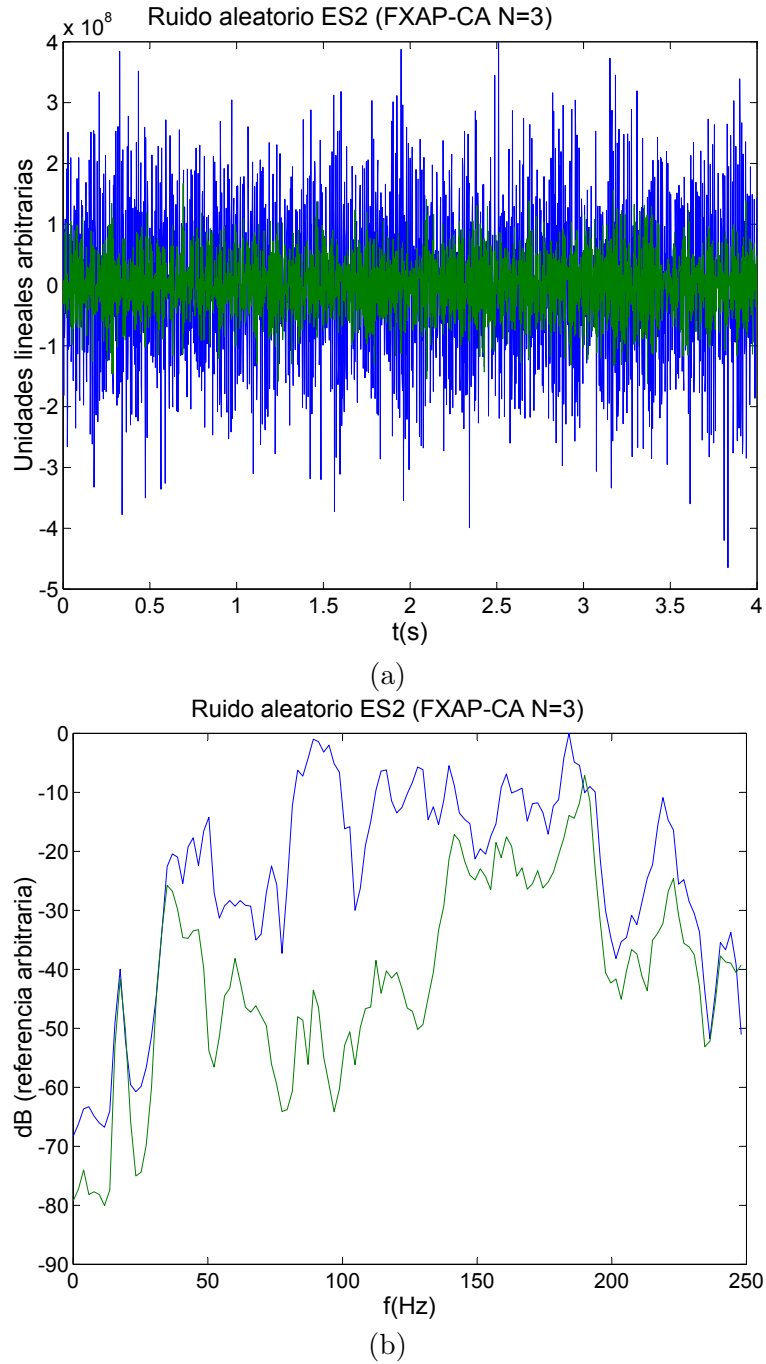


Figura 7.11: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares $N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

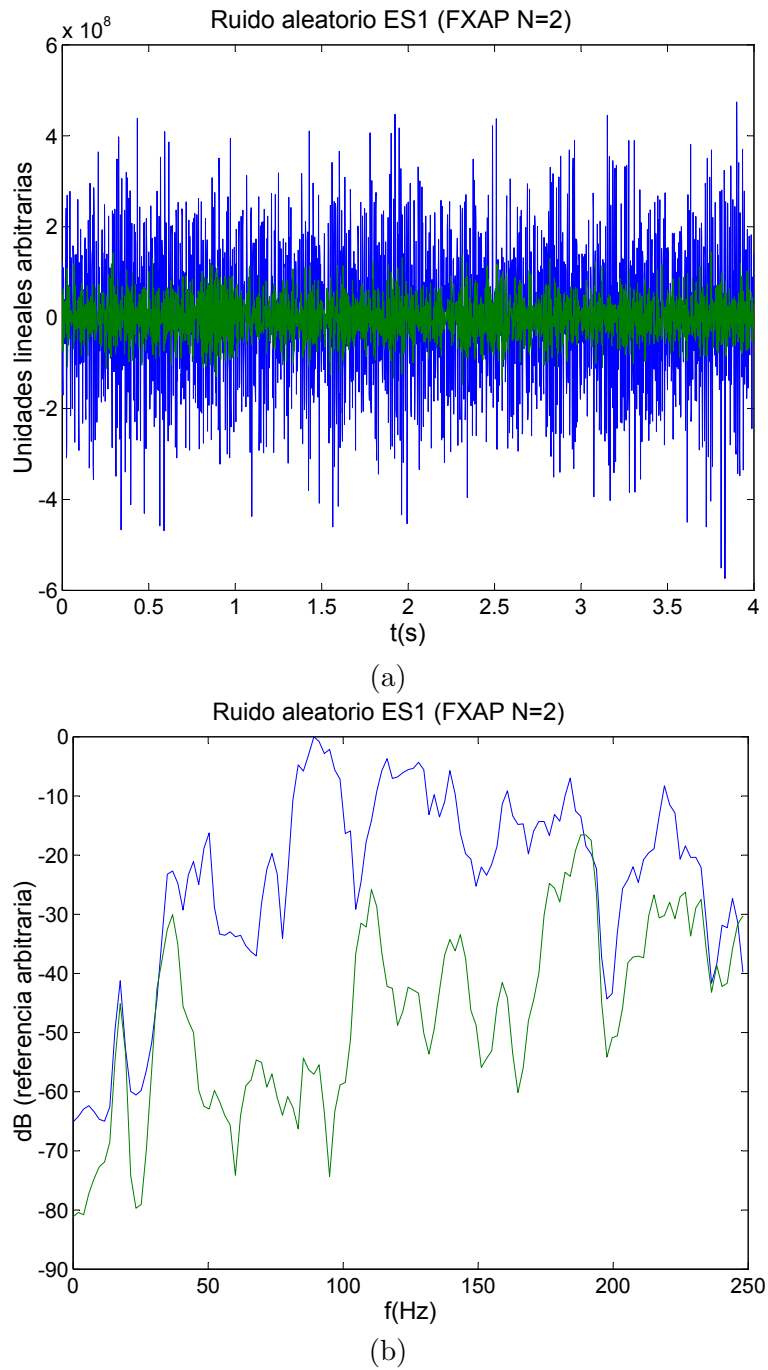


Figura 7.12: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

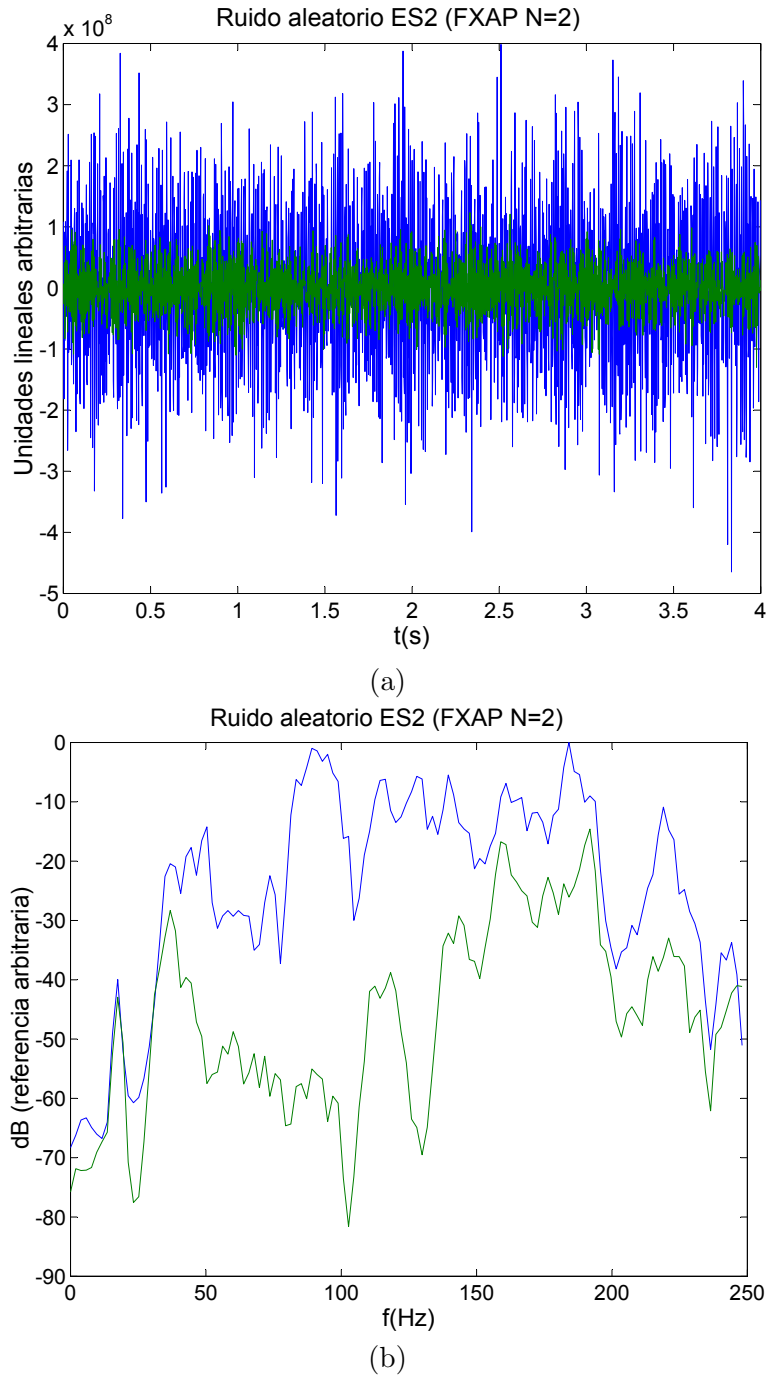


Figura 7.13: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

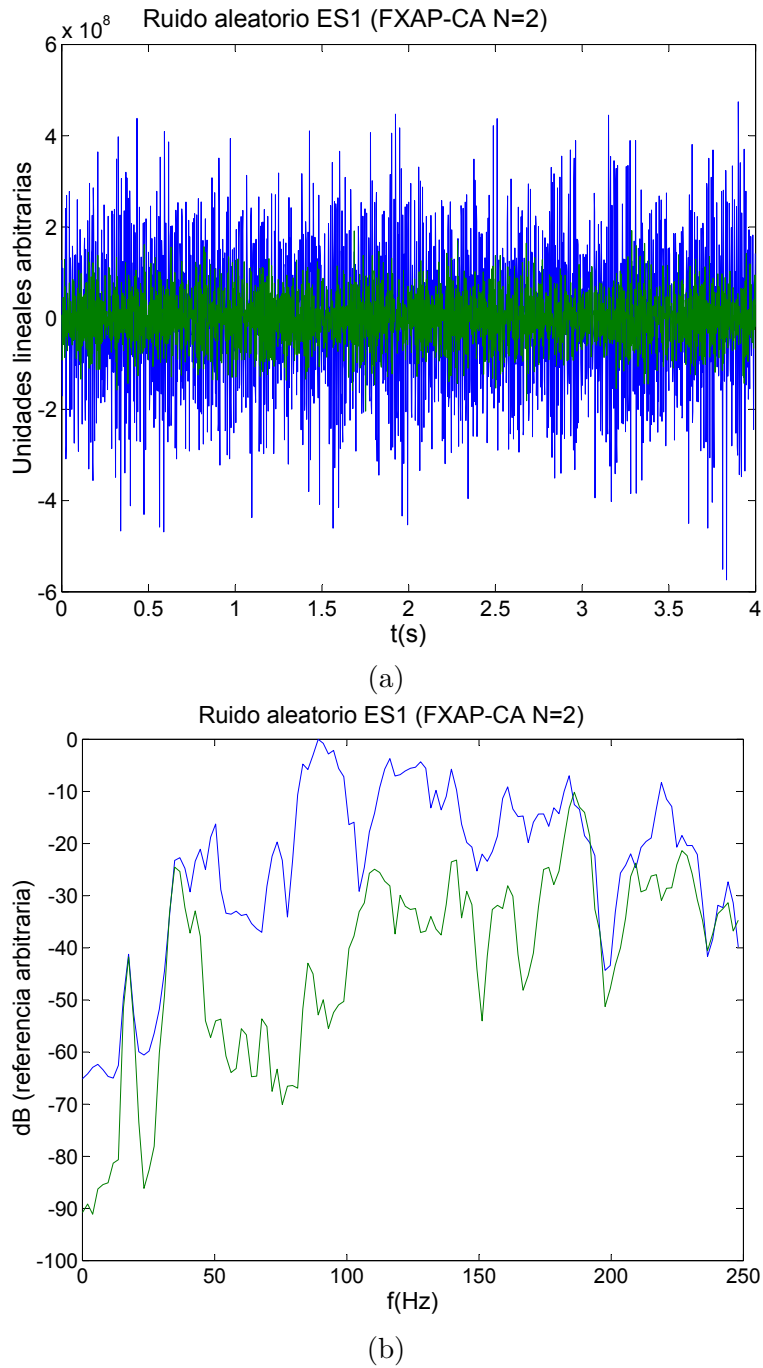


Figura 7.14: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

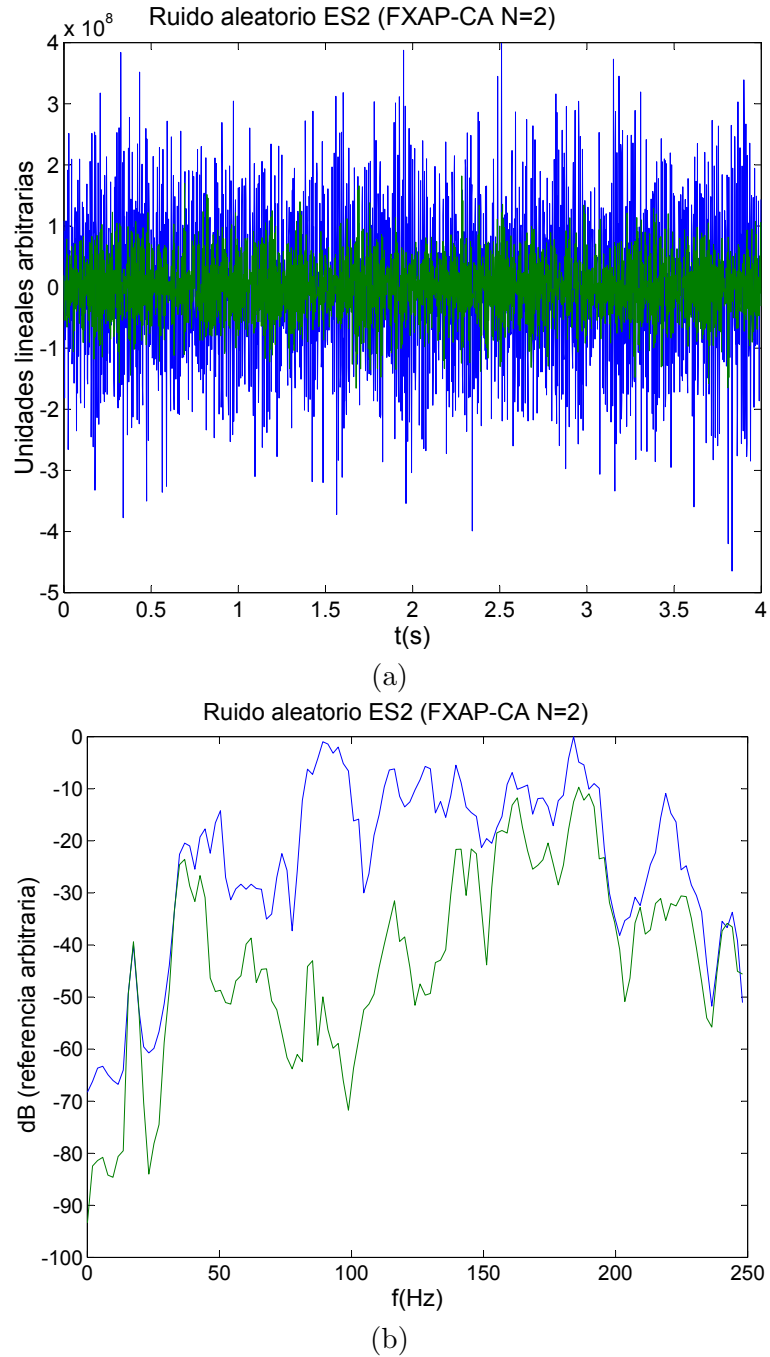


Figura 7.15: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares $N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

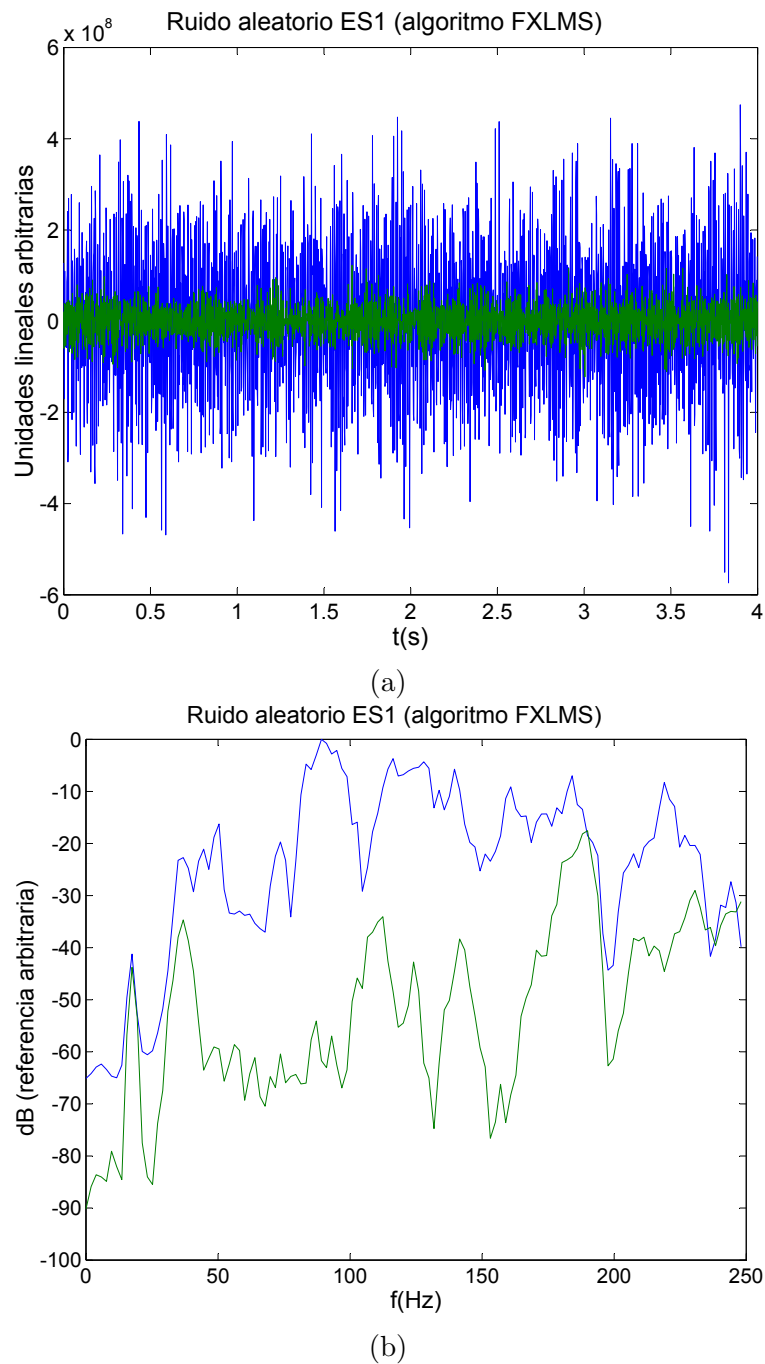
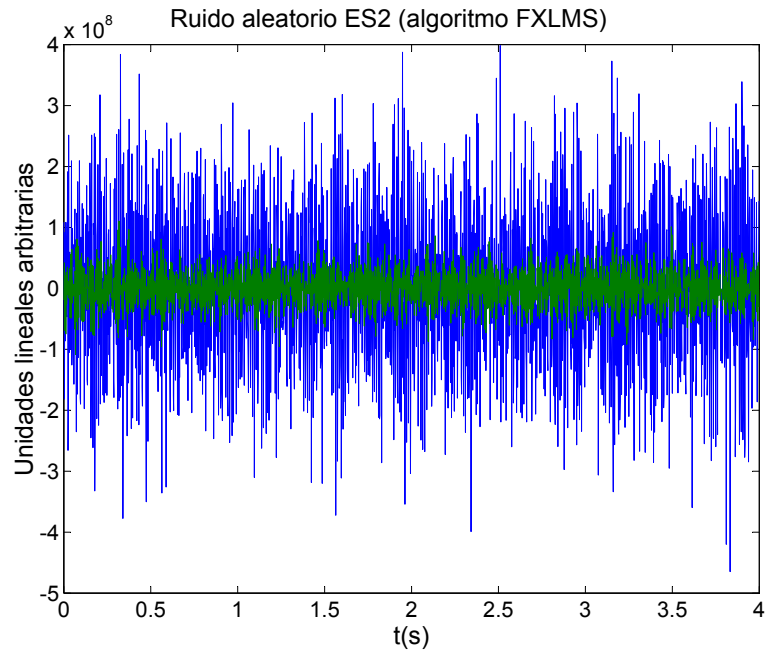
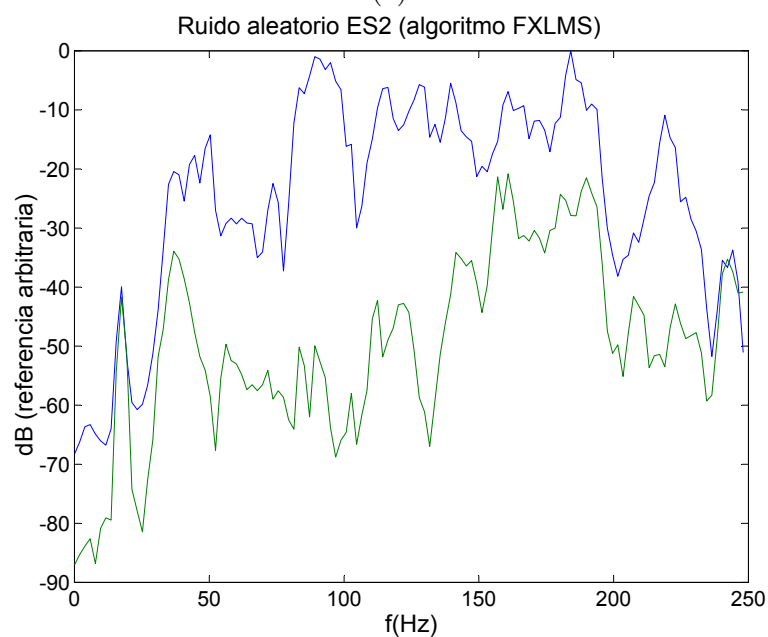


Figura 7.16: Señal captada en el sensor de error ES1 con el algoritmo FXLMS normalizado. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

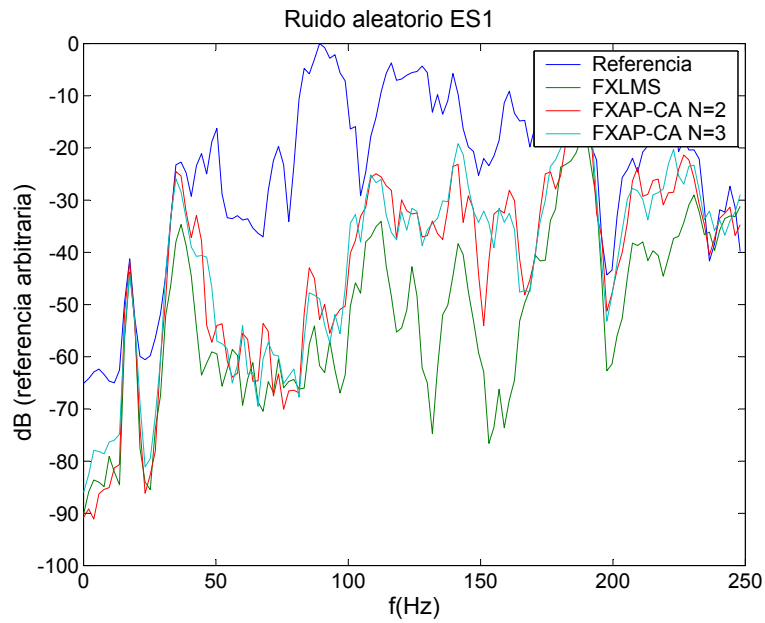


(a)

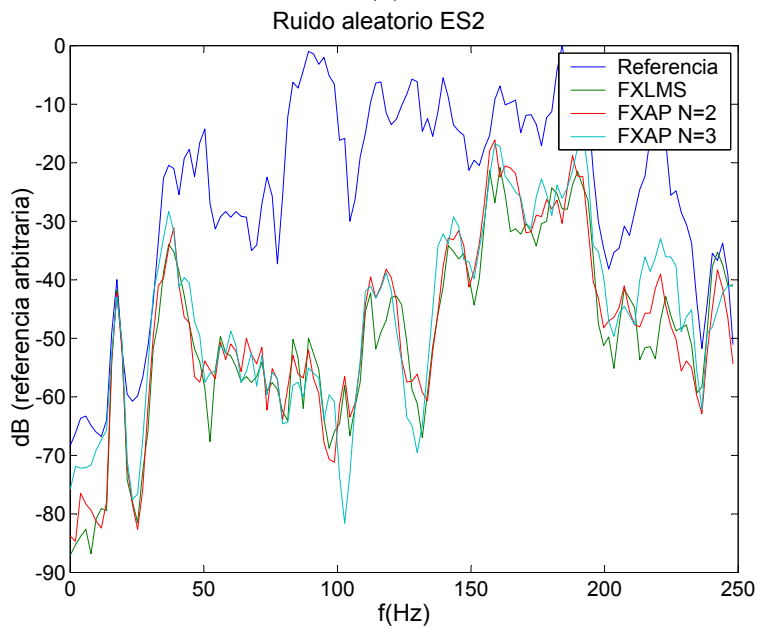


(b)

Figura 7.17: Señal captada en el sensor de error ES2 con el algoritmo FXLMS normalizado. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.



(a)



(b)

Figura 7.18: Espectros comparativos de las señales en los sensores de error usando ruido blanco (a) ES1 usando coeficientes auxiliares en los algoritmos de proyección afín, (b) ES2 sin usar coeficientes auxiliares en los algoritmos de proyección afín.

7.3.2. Tono 100 Hz

Veamos qué sucede cuando la señal de ruido es ahora un tono puro (de frecuencia 100 Hz). De nuevo, se usaron 120 coeficientes para los filtros adaptativos del sistema C.A.R. y una constante de convergencia igual al caso anterior ($\mu = 0,002$). En este caso, dada la simplicidad del ruido, las atenuaciones alcanzadas eran del orden de 38 dB. Los niveles de las señales en el tiempo y sus espectros tanto cuando se ejerce control (color verde) como cuando no (color azul) quedan reflejados en las figuras 7.19 a 7.28.

Como puede apreciarse en las gráficas de la densidad espectral de potencia, parece que hay algún problema de cancelación en las proximidades de 200 Hz. Si nos fijamos en las respuestas en frecuencia de los caminos secundarios, podemos encontrar mínimos de amplitud en el entorno de dicha frecuencia tanto en c11 como en c21 (lo que indica que el sensor ES1 puede estar colocado cerca de un nulo de presión de dicha frecuencia, que coincide con la frecuencia de resonancia de los modos (1 0 3), (0 2 3) y (1 1 3), y por lo tanto apenas recibiría información de señales acústicas en el entorno de 200 Hz) como en los caminos c11 y c12 (que nos sugiere que el actuador SS1 podría estar cerca de un nulo de velocidad de dicha frecuencia, por lo que no podría generar señales acústicas en el entorno de dicha frecuencia). En ambos casos, el sistema estaría físicamente mal distribuido para poder controlar eficientemente dicha frecuencia, lo que nos permite entender el pobre funcionamiento del controlador activo de ruido (independientemente del algoritmo usado) en el entorno de 200 Hz.

En bajas frecuencias (por debajo de 40 Hz), también se aprecia un comportamiento anómalo de los algoritmos que puede ser debido a la pobre respuesta de los equipos electroacústicos (principalmente de los altavoces) en dichas frecuencias, o también a un problema físico por la colocación de algún elemento próximo a un nulo del primer modo (en torno a los 36 Hz).

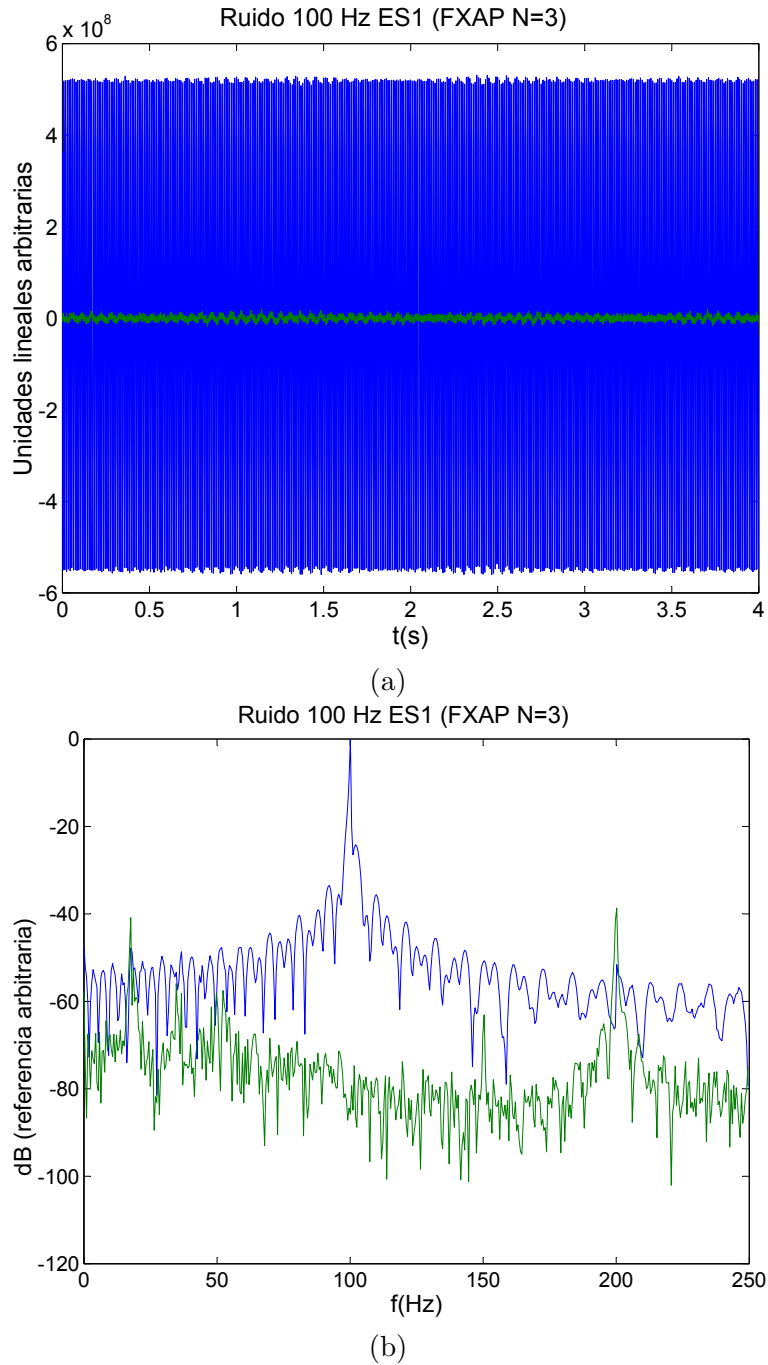


Figura 7.19: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

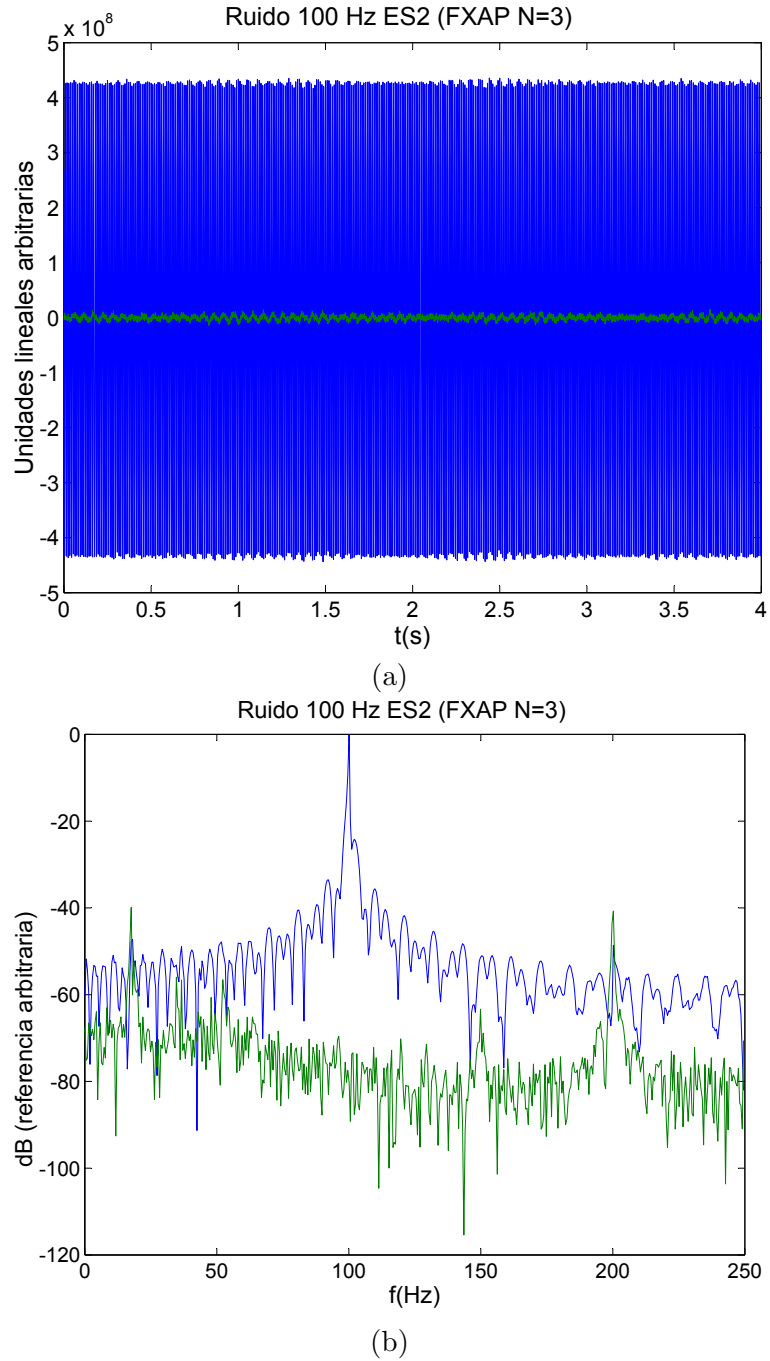


Figura 7.20: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

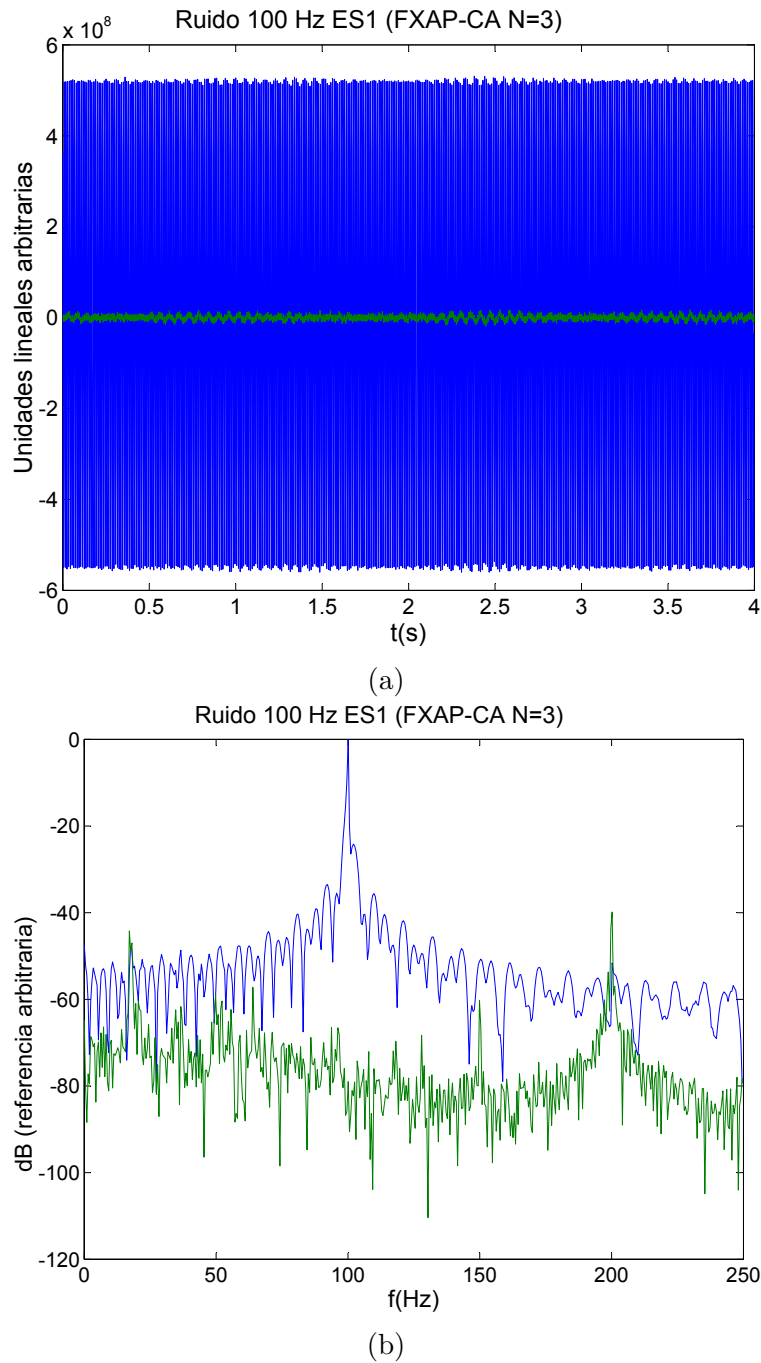


Figura 7.21: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

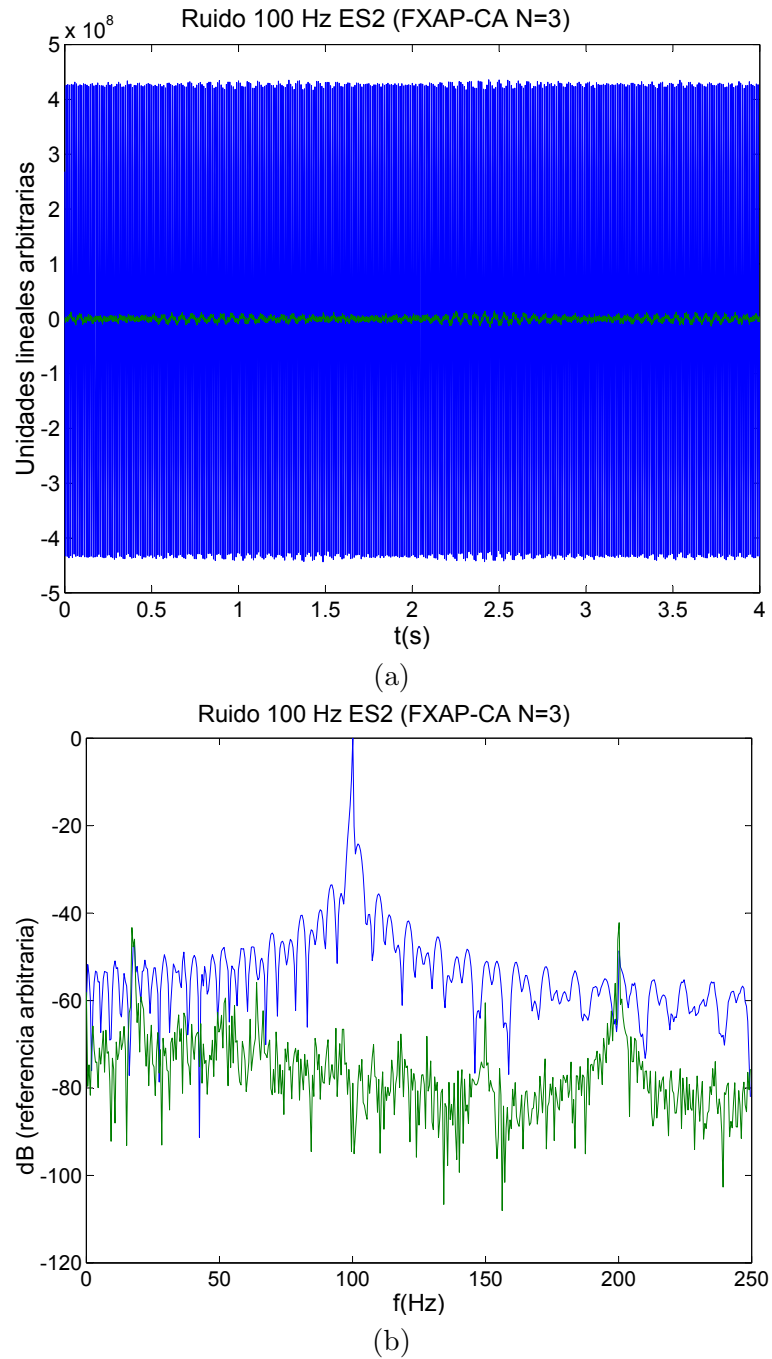


Figura 7.22: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

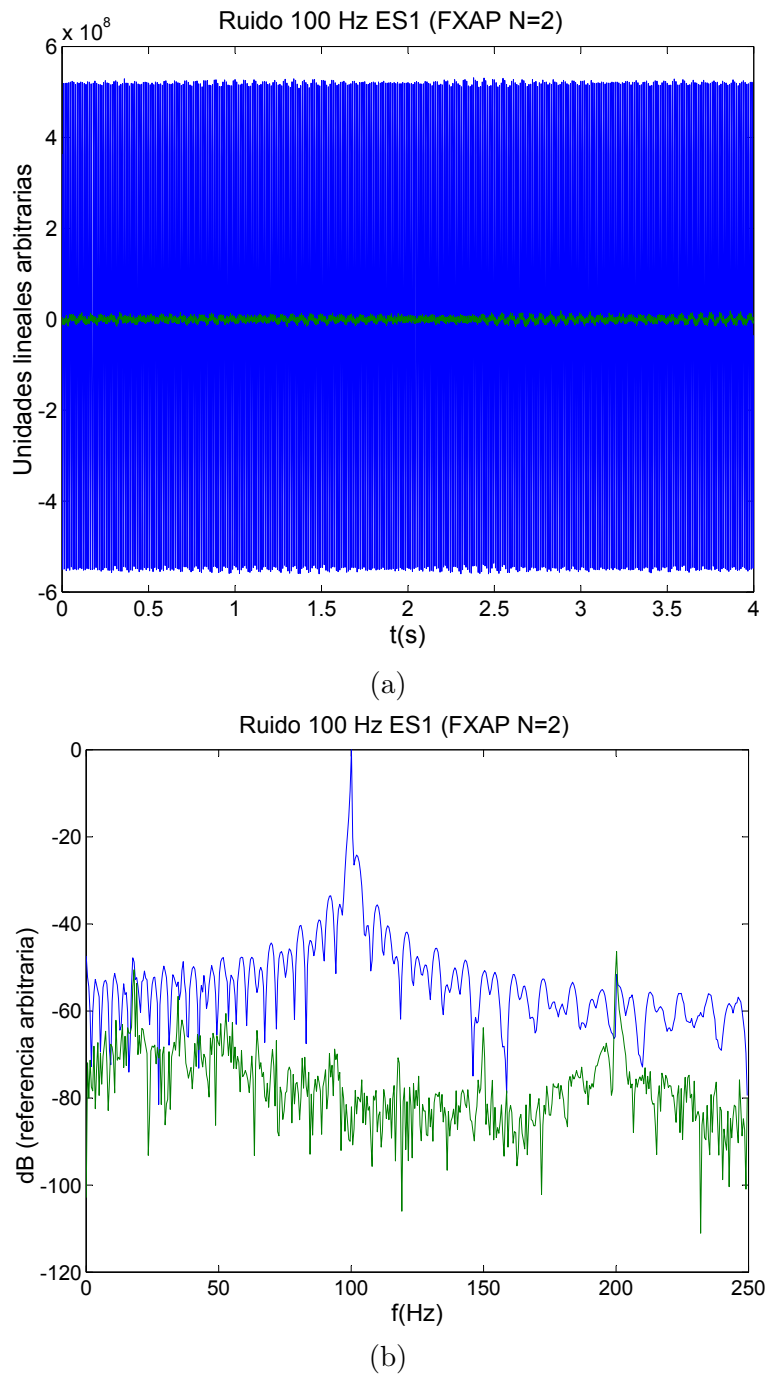
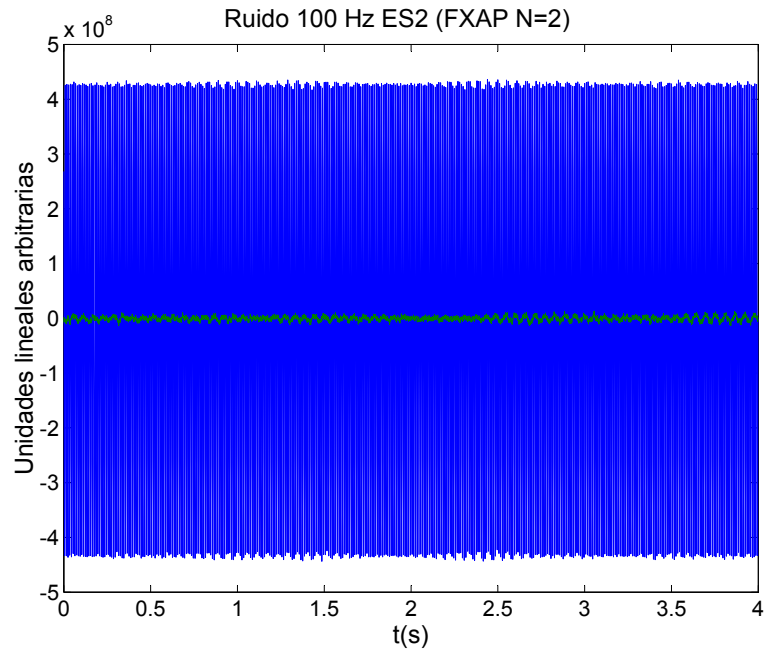
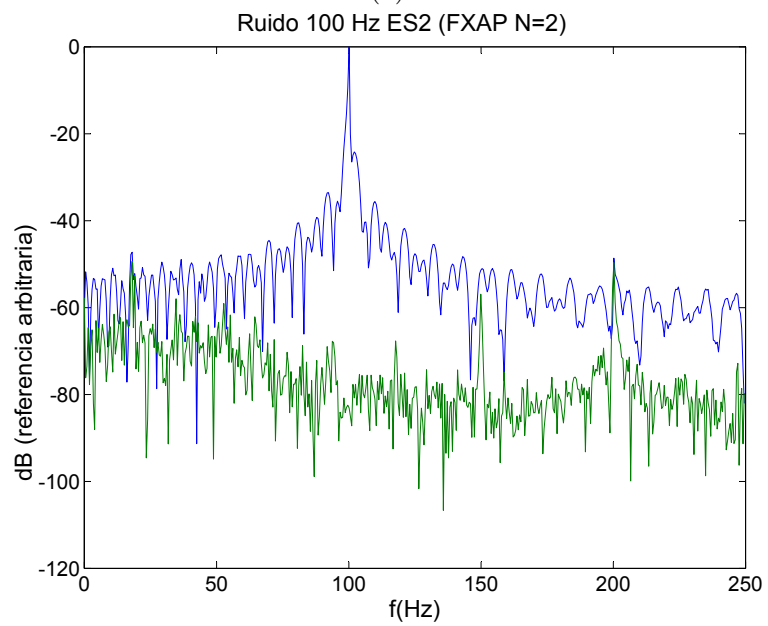


Figura 7.23: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.



(a)



(b)

Figura 7.24: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

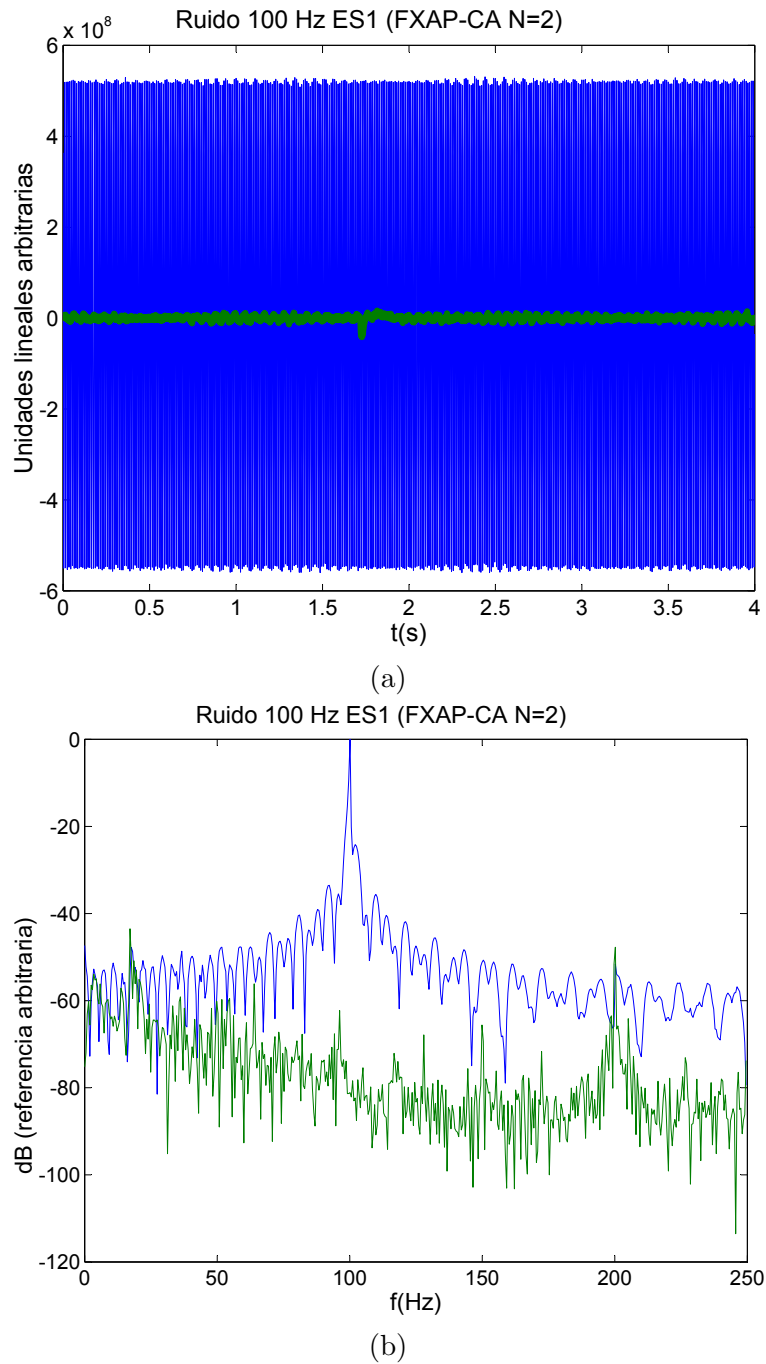


Figura 7.25: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

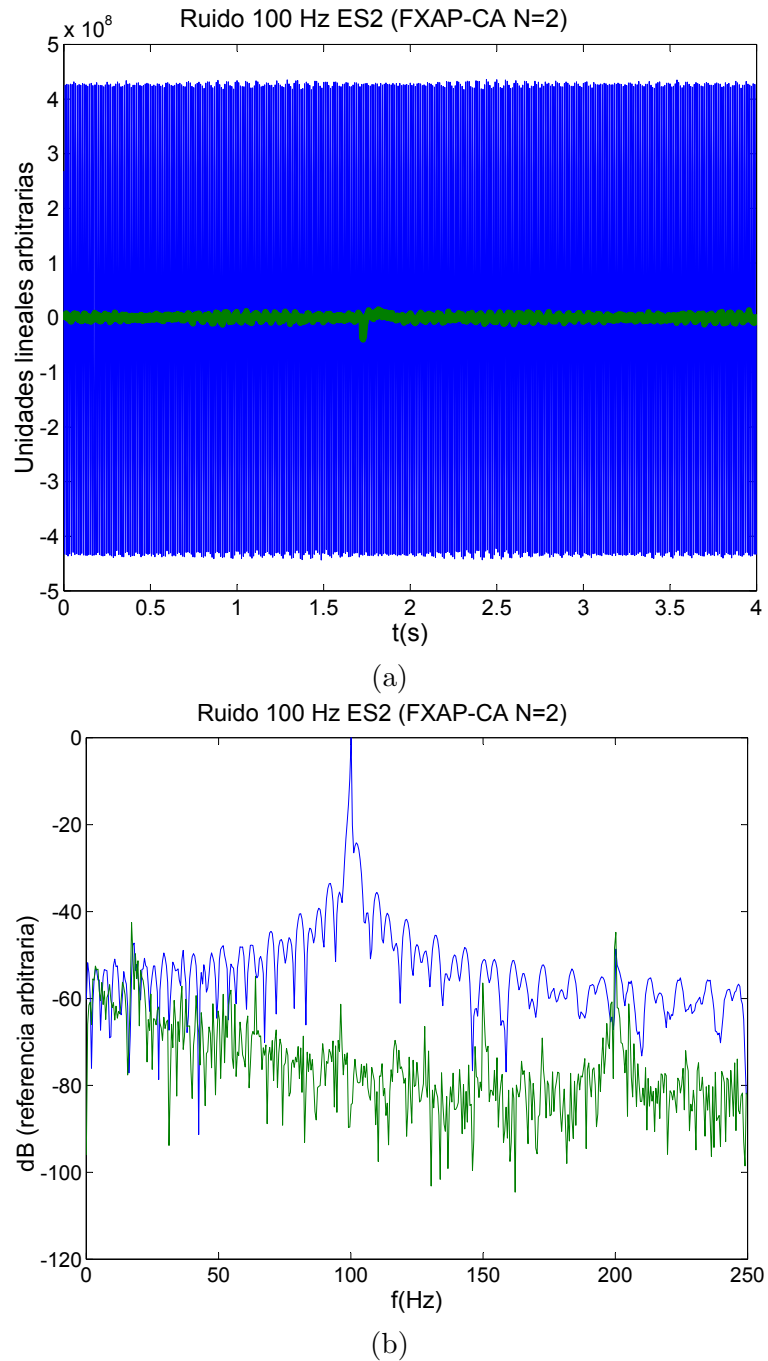


Figura 7.26: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

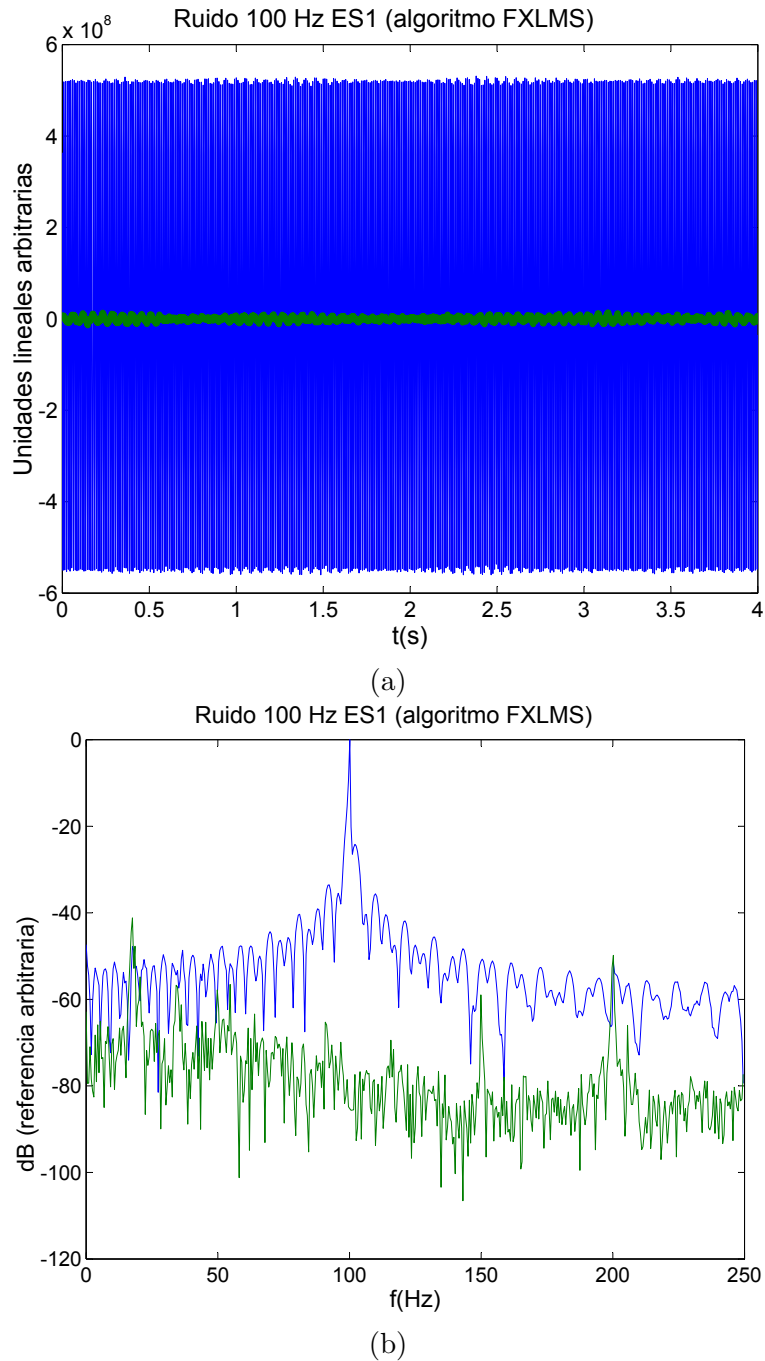
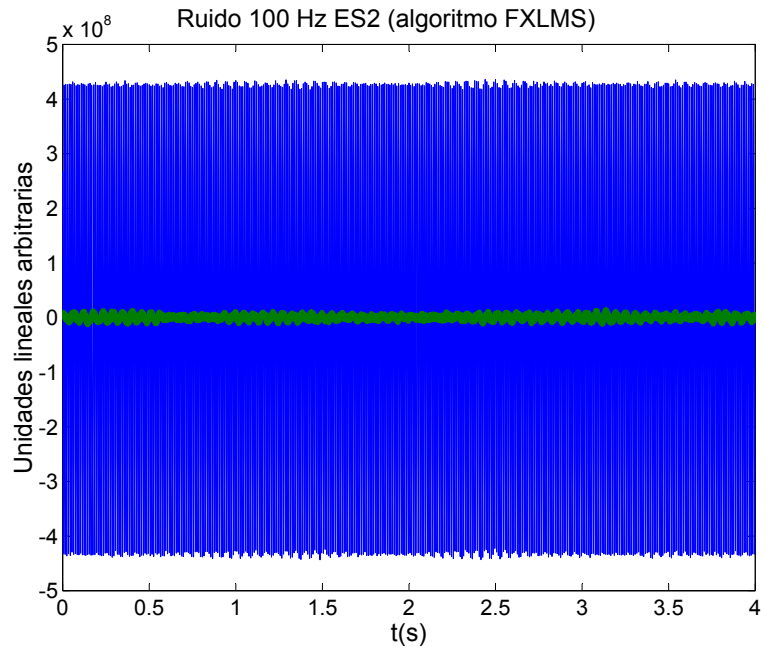
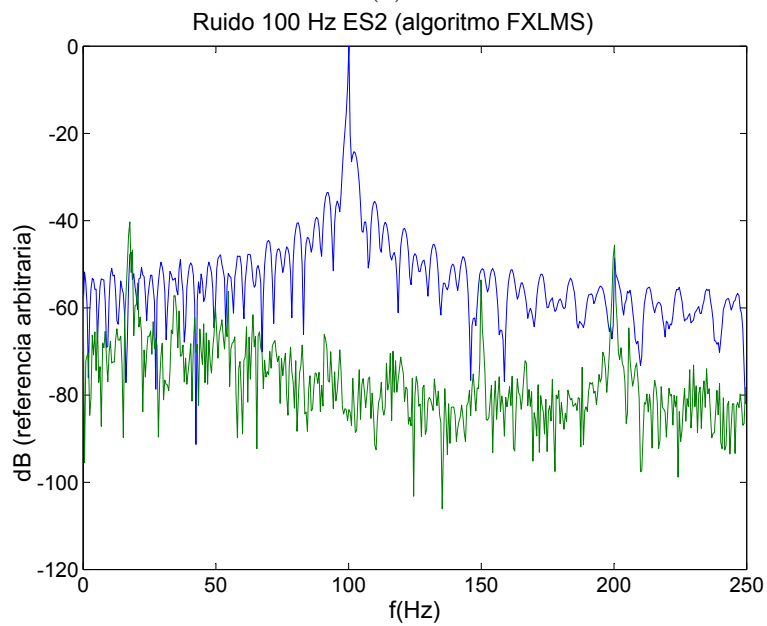


Figura 7.27: Señal captada en el sensor de error ES1 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.



(a)



(b)

Figura 7.28: Señal captada en el sensor de error ES2 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

7.3.3. Ruido periódico

Por último, se ha estudiado un caso intermedio entre los dos anteriores, y es cuando la señal de ruido es una composición de tonos (en este caso un tono de 10 Hz y sus diferentes armónicos hasta 250 Hz). De nuevo, se usaron los mismos algoritmos con los mismos parámetros de funcionamiento que en los casos anteriores. Las atenuaciones alcanzadas han sido en este caso del orden de 20 dB y los resultados, pueden observarse en las figuras 7.29 a 7.38.

En la figura 7.39 se puede apreciar de forma comparativa la evolución de la presión sonora (dB_{SPL}) en el oído derecho del maniquí (MS.D) durante los primeros segundos de medida, de forma que se aprecie con detalle el régimen transitorio de los diferentes algoritmos. Como se podía esperar, la velocidad de convergencia de los algoritmos de proyección afín es algo mayor que la del FXLMS normalizado, e incluso puede apreciarse que las velocidades de los algoritmos de proyección afín de orden 3 son ligeramente mayores que las de orden 2. En cuanto al uso de los coeficientes auxiliares o no, aunque el transitorio no es idéntico, puede concluirse que el comportamiento de ambos algoritmos es bastante similar. Hay que notar que en todos ellos se usó la misma constante de convergencia (para comparar sus prestaciones con los mismos parámetros de configuración), aunque ésta no era la que mayor velocidad de convergencia proporcionaba en cada caso. De hecho, como se desprende de los análisis teóricos y resultados de simulación, la constante de convergencia puede ser mayor cuando usamos la estructura de filtrado-x modificada proporcionando una velocidad de convergencia más rápida.

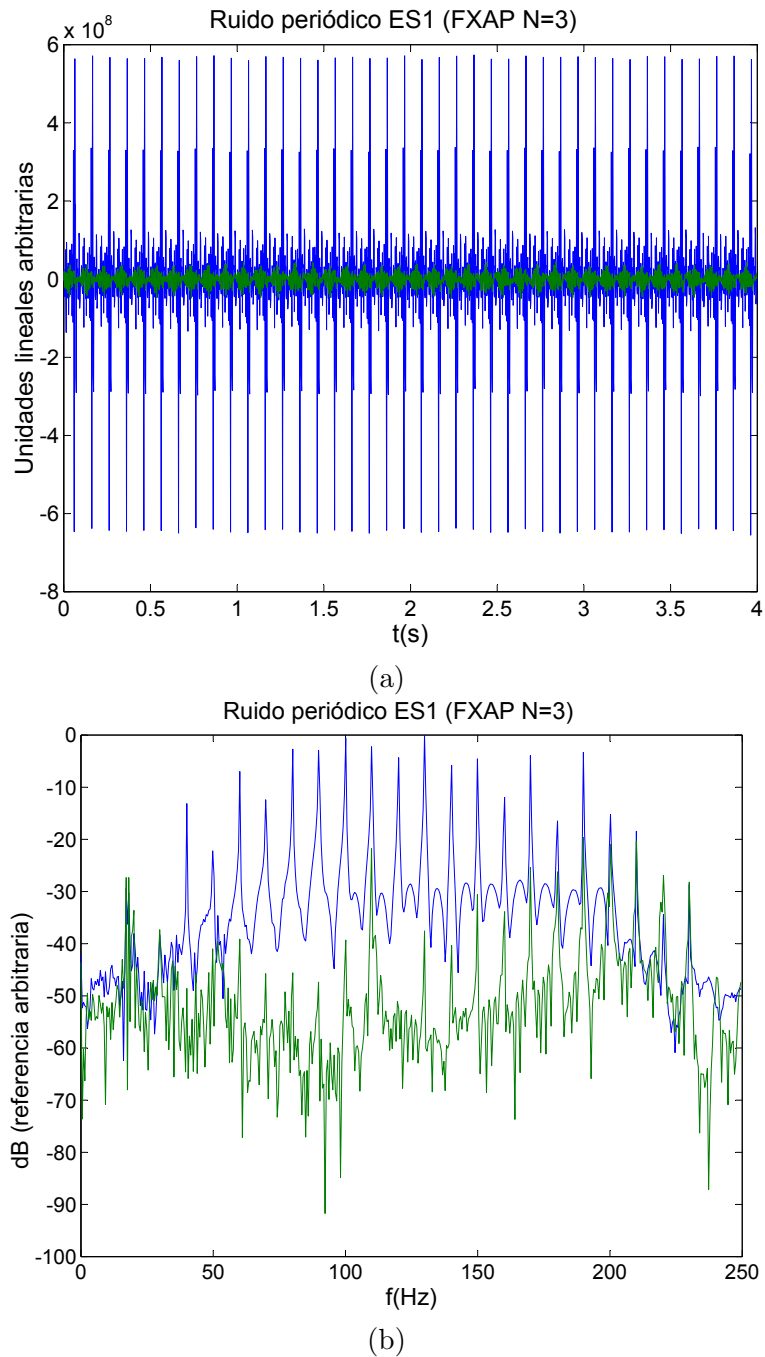


Figura 7.29: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

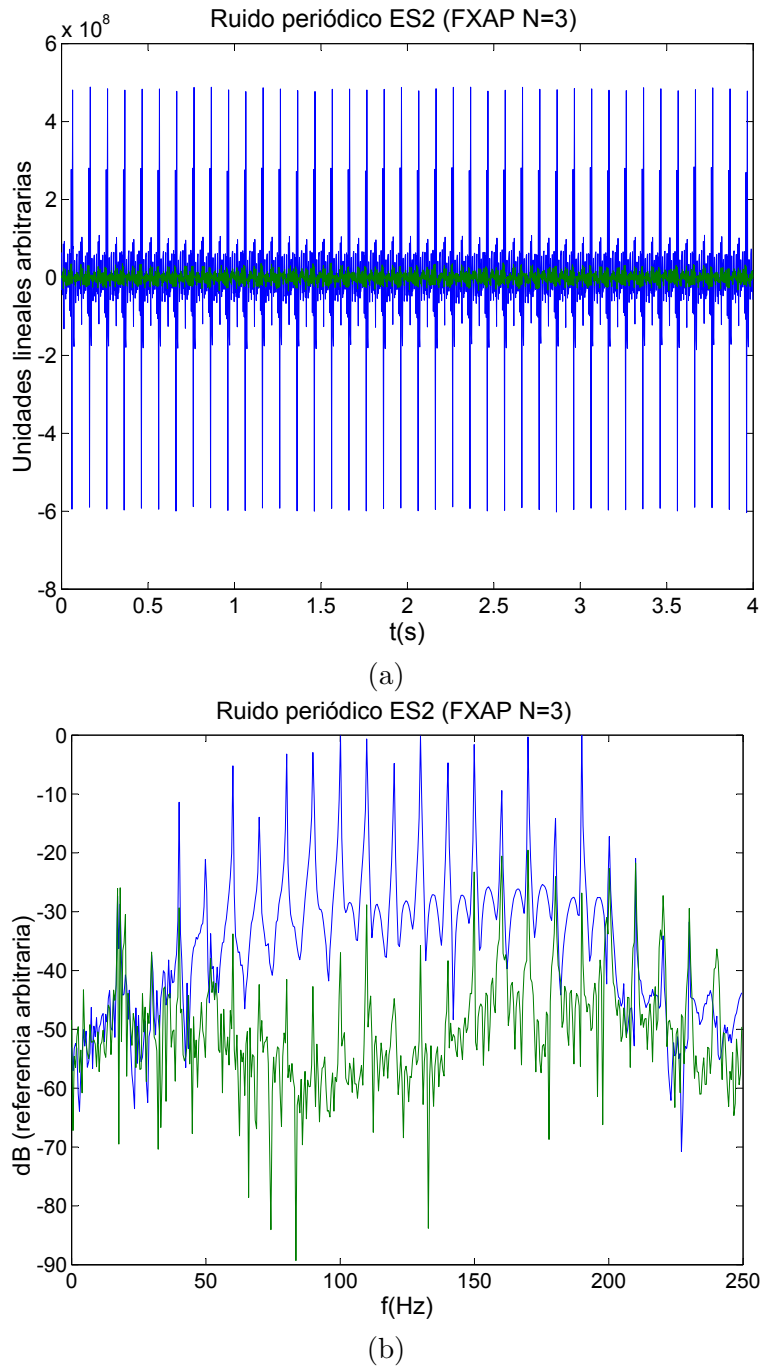


Figura 7.30: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

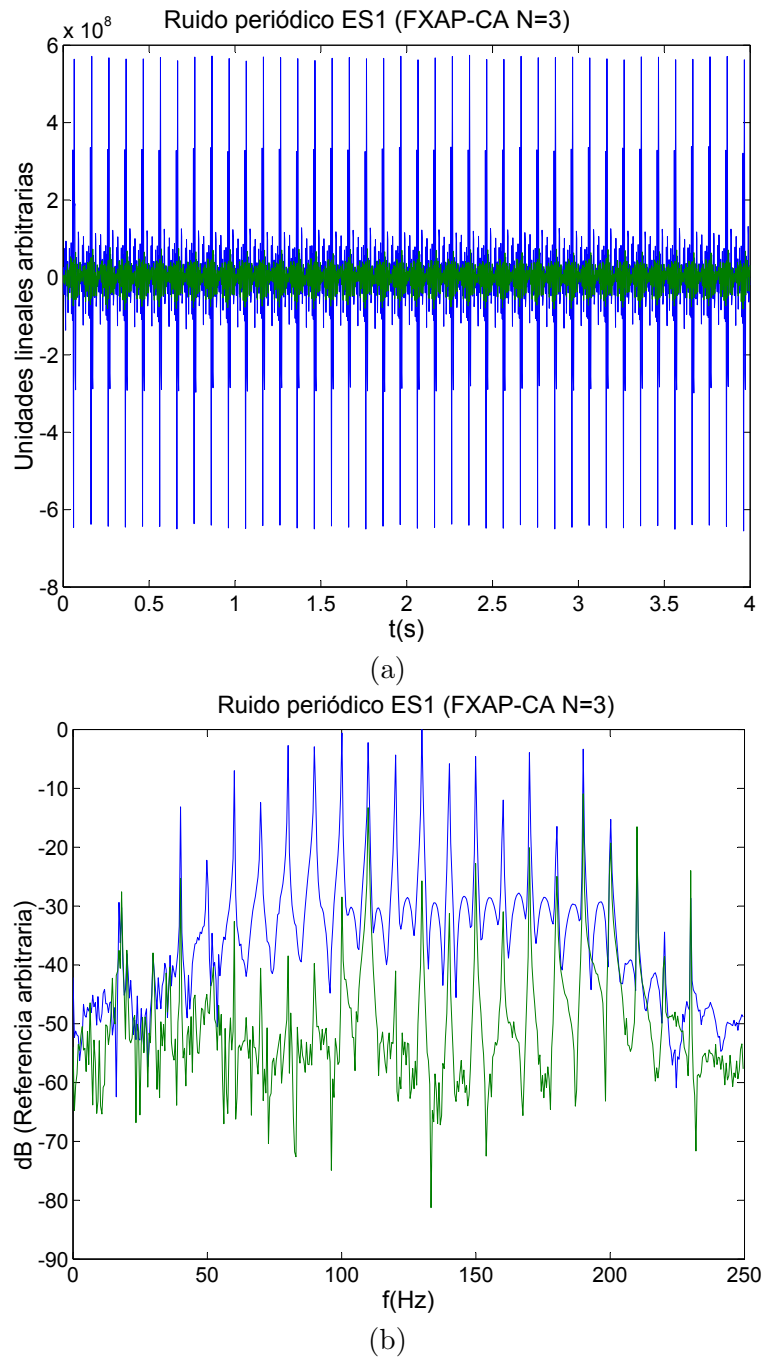


Figura 7.31: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

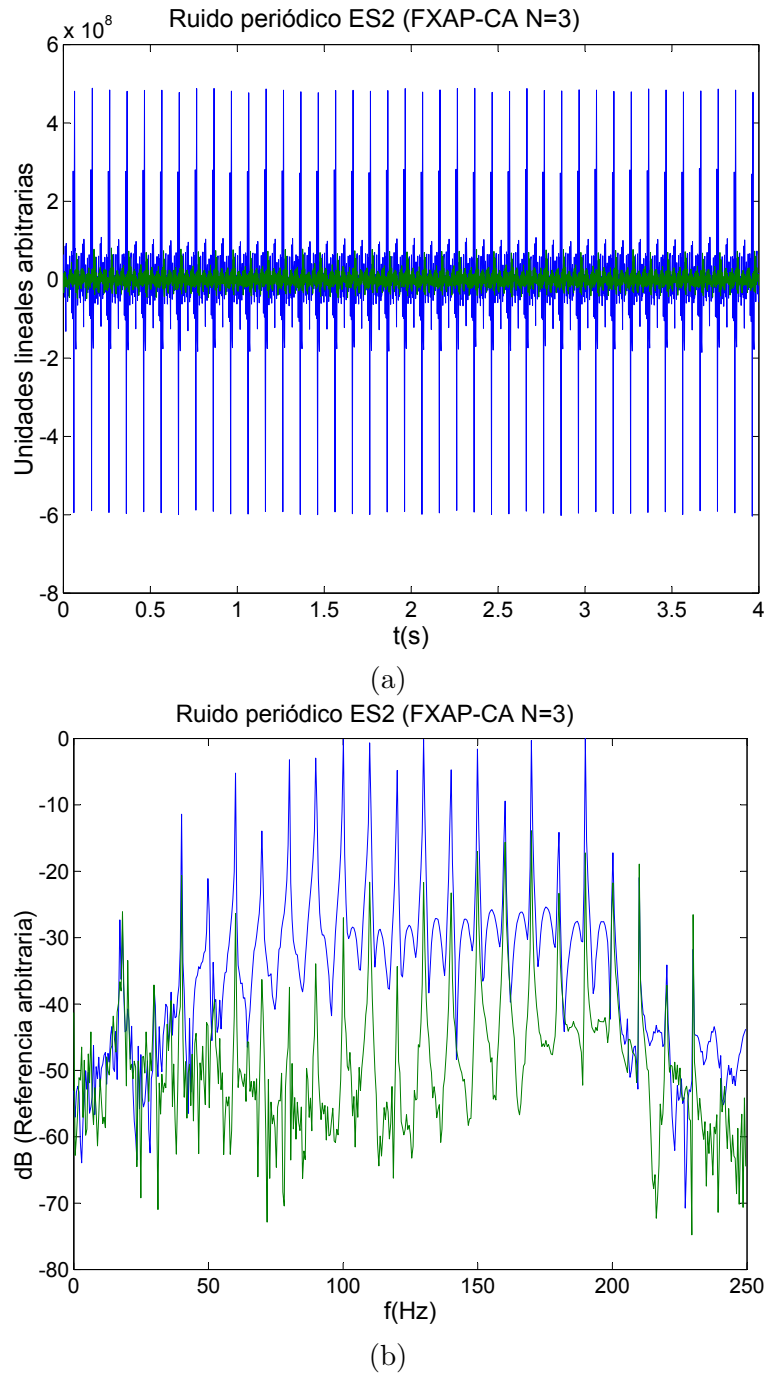
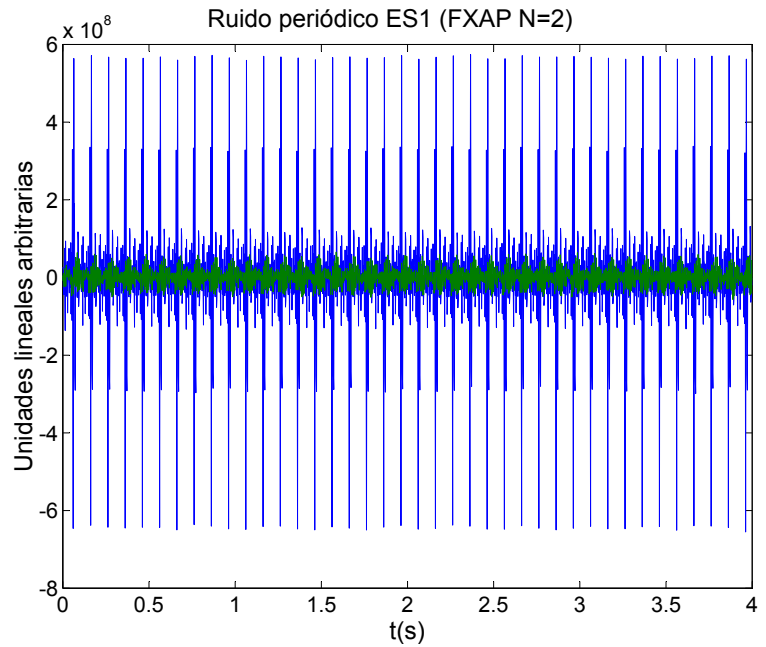
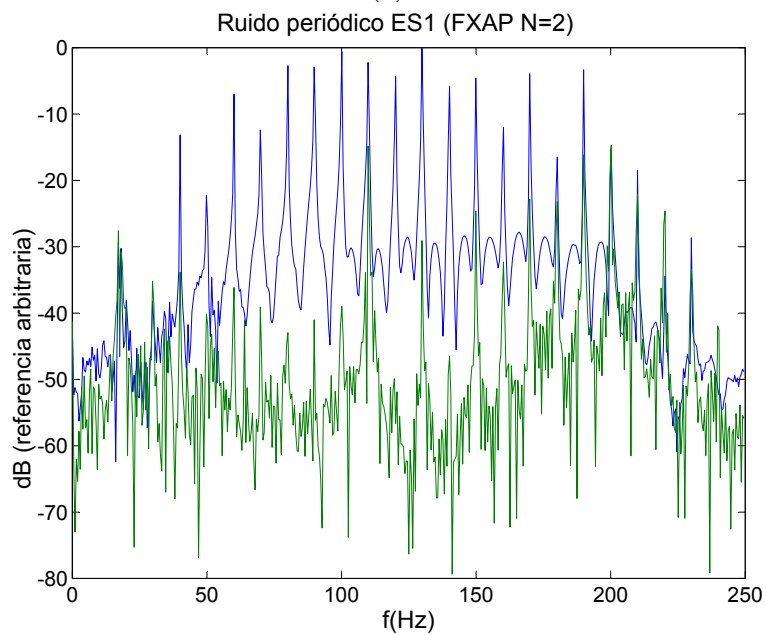


Figura 7.32: Señal captada en el sensor de error ES2 con el algoritmo de proyección sín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.



(a)



(b)

Figura 7.33: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

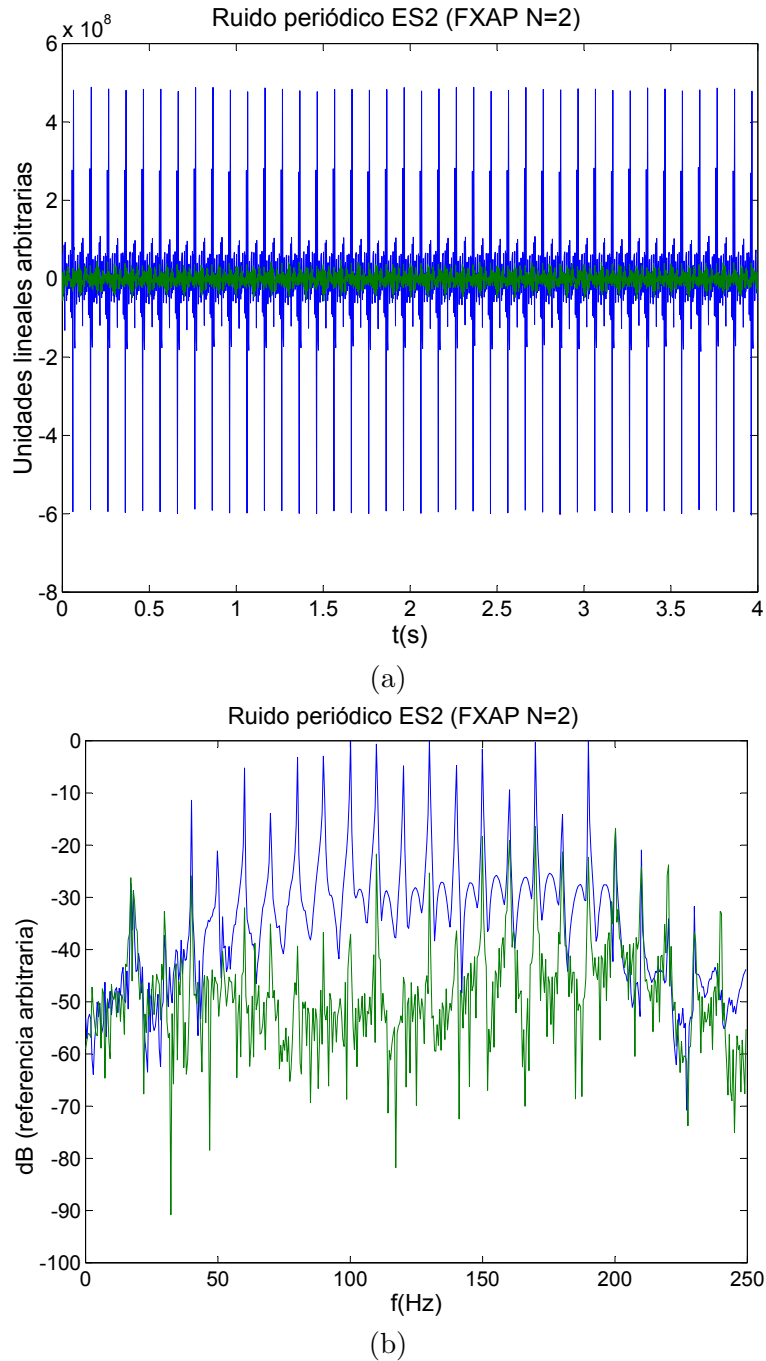
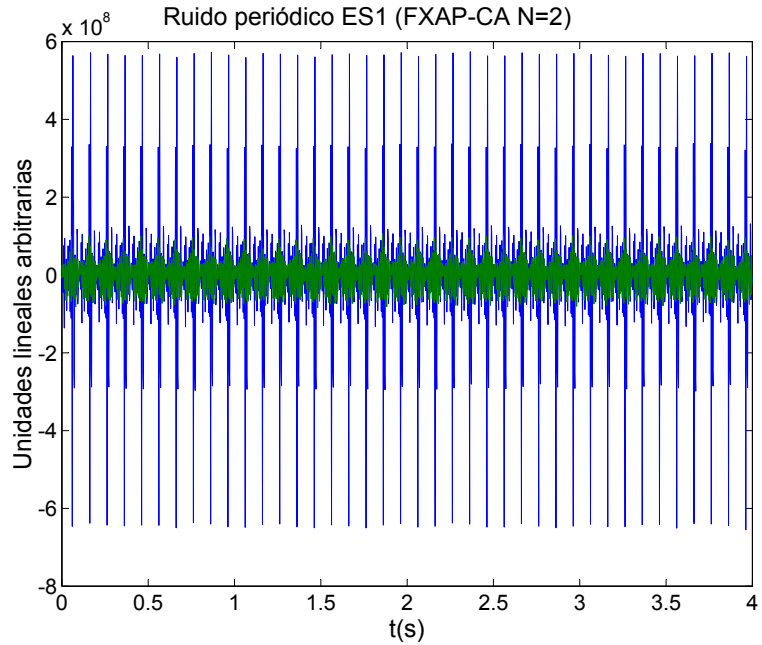
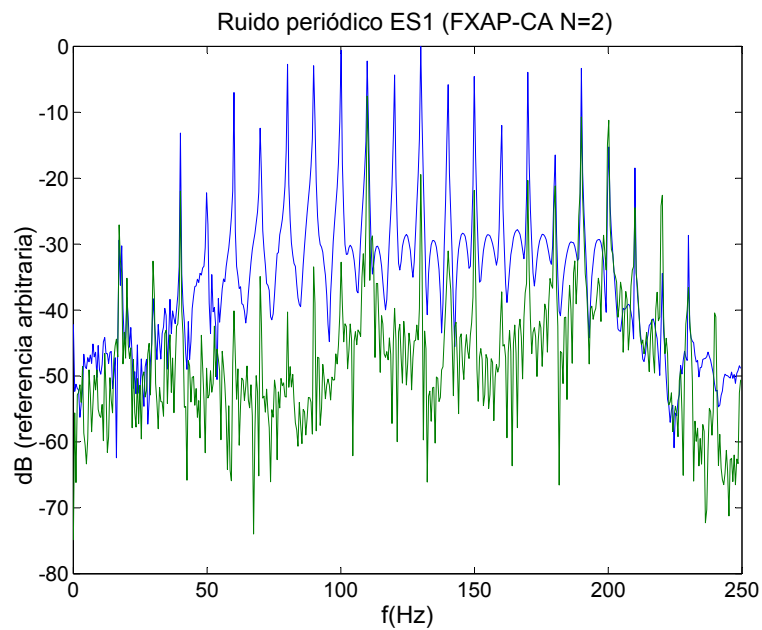


Figura 7.34: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.



(a)



(b)

Figura 7.35: Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

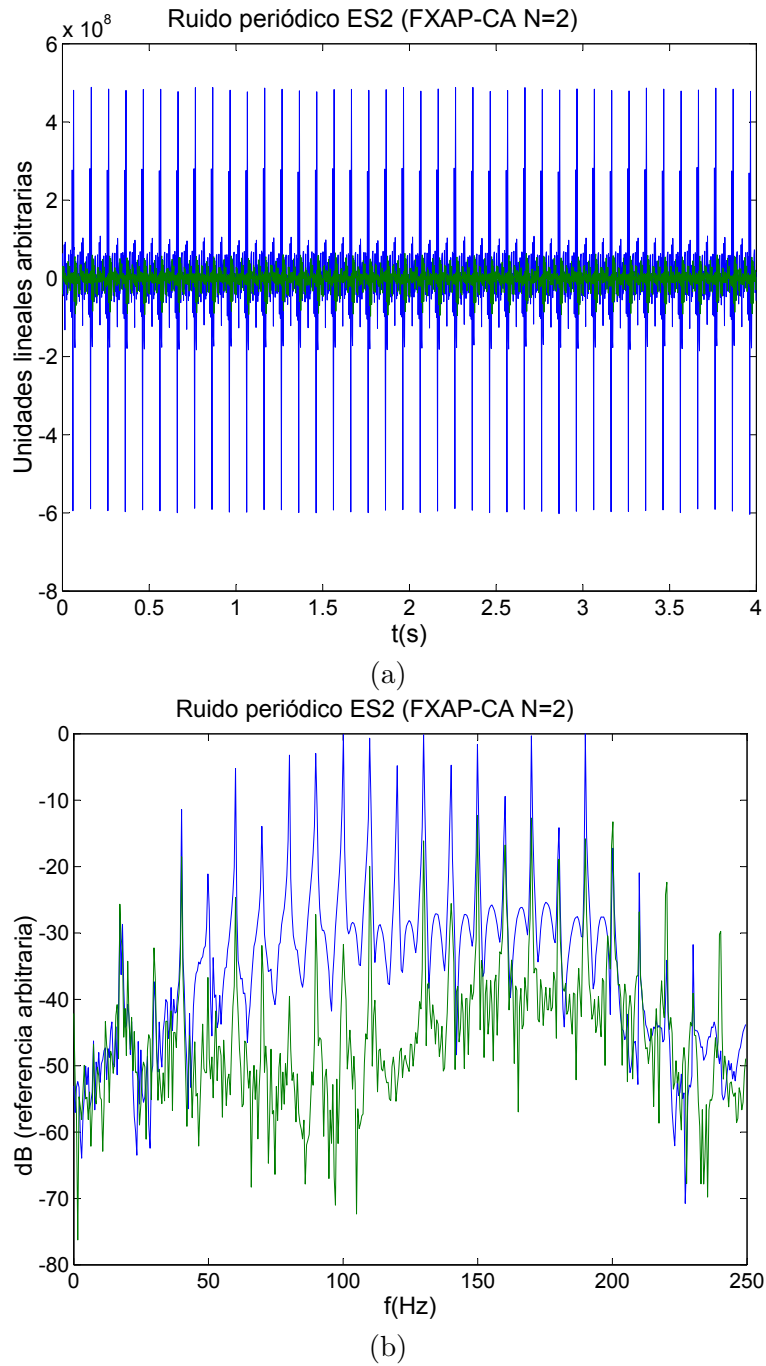
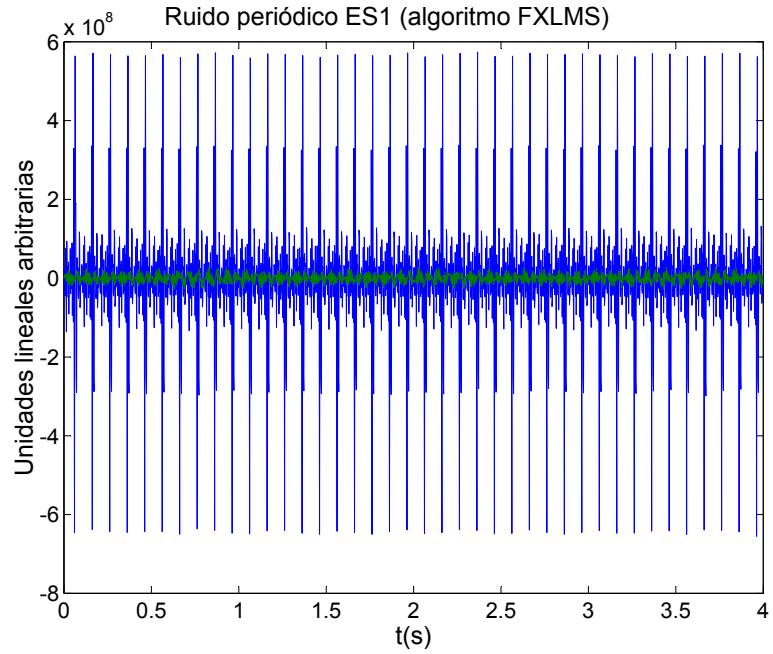
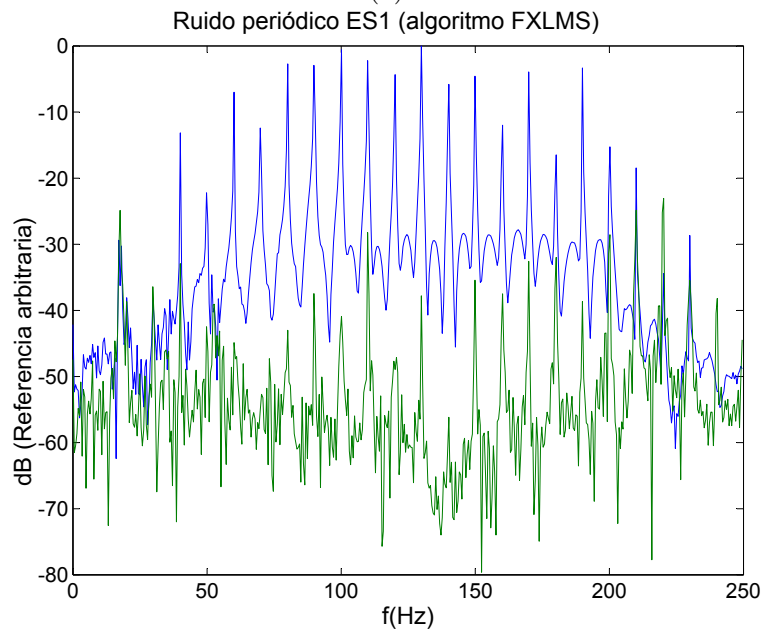


Figura 7.36: Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.



(a)



(b)

Figura 7.37: Señal captada en el sensor de error ES1 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

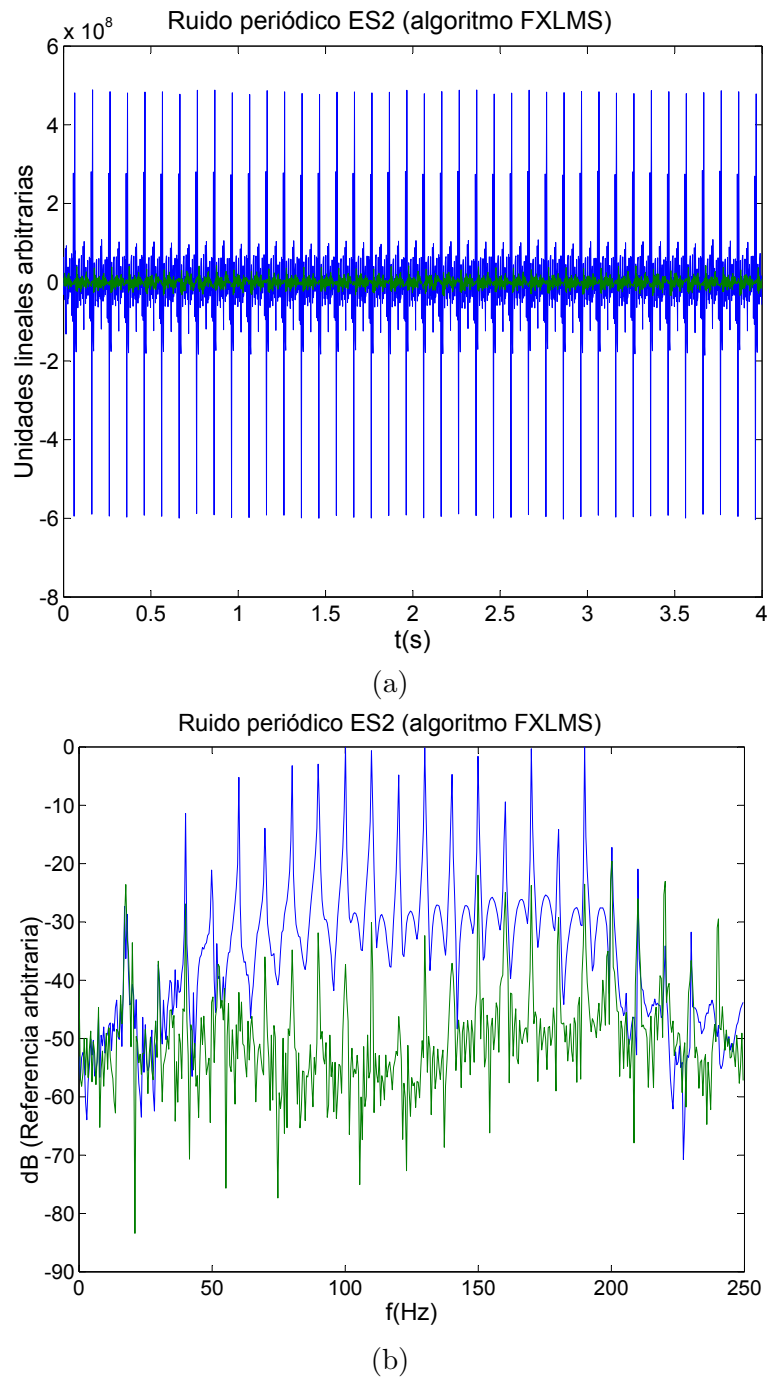


Figura 7.38: Señal captada en el sensor de error ES2 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

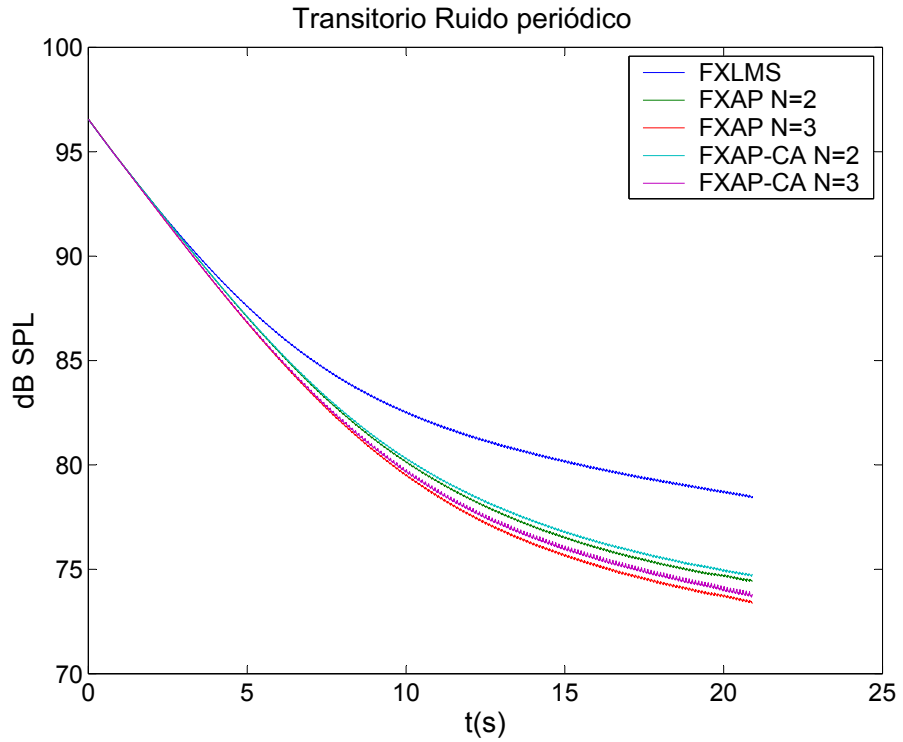


Figura 7.39: Transitorio del nivel de presión sonora (en dB_{SPL}) aplicando los diferentes algoritmos para una señal de ruido periódico.

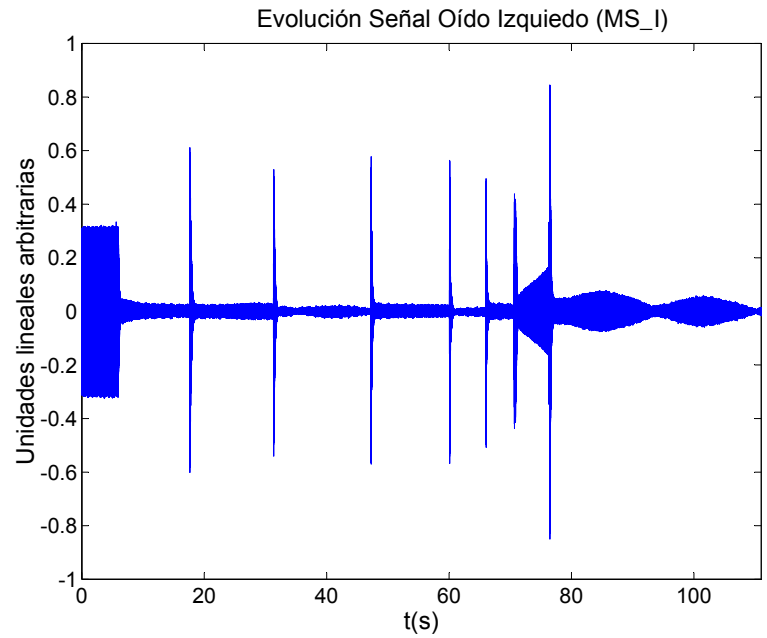
7.3.4. Robustez

Para el estudio de la robustez de estos algoritmos, se han realizado dos experimentos: el primero de ellos ha consistido en usar un tono cuya frecuencia variaba de forma brusca a lo largo del tiempo (observando la capacidad de los algoritmos para readaptarse) y el segundo, añadir a un tono de ruido (señal que se desea cancelar) otro ruido incorrelado (en este caso una señal con música) observando el 'tracking' o cómo responde el sistema a la adicción de esta nueva

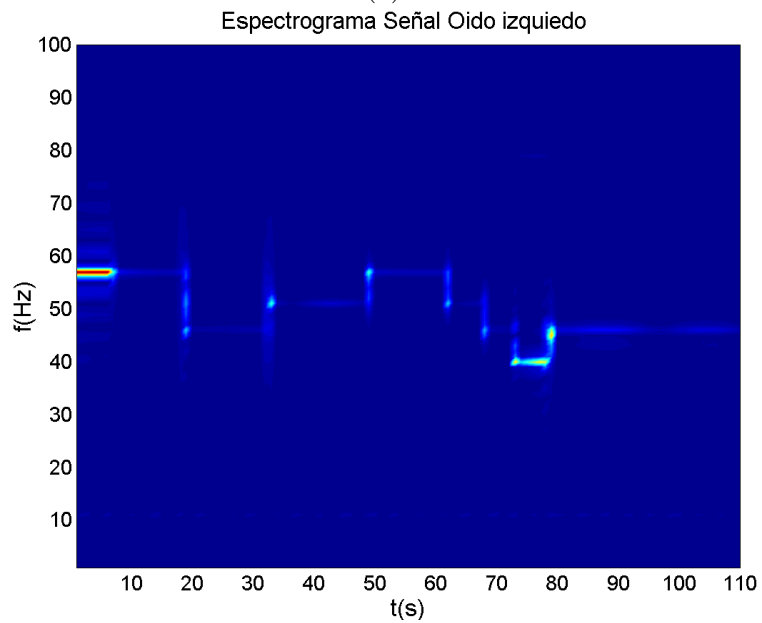
señal que no se puede cancelar por estar incorrelada con la referencia. En las figuras 7.40 y 7.41 podemos observar cómo el sistema es capaz de reengancharse perfectamente ante cualquier variación en la frecuencia de la señal a cancelar. El algoritmo usado fue el de proyección afín con una constante de convergencia $\mu = 0.001$ y orden de proyección $N = 2$.

Debe hacerse notar que los niveles de la señal captada por el oído derecho del maniquí son mayores que los del oído izquierdo. Esto es debido a que el sistema no era perfectamente simétrico y la fuente de ruido estaba en una posición ligeramente más cercana al oído izquierdo del maniquí.

En cuanto al funcionamiento de los algoritmos cuando existe un ruido adicional incorrelado, se ha comprobado cómo el sistema sigue siendo robusto y cancela sólo la parte del ruido total de la cual tiene referencia. En este caso, era un tono de 80 Hz al que se le añadía música como señal de ruido incorrelada. Cuando el sistema de control funcionaba, el ruido de 80 Hz se minimizaba, quedando solo el ruido incorrelado (la señal de música). En la figura 7.42 se puede apreciar tanto la evolución en el tiempo de la señal cuando se ejerce control de ruido y cuando no, como el espectro de la misma.



(a)



(b)

Figura 7.40: Señal captada en el oído izquierdo (MS_I) del maniquí usando el algoritmo de proyección afín ($N = 2$) cuando la señal a cancelar es un tono variante con la frecuencia. (a) Evolución en el tiempo. (b) Espectrograma.

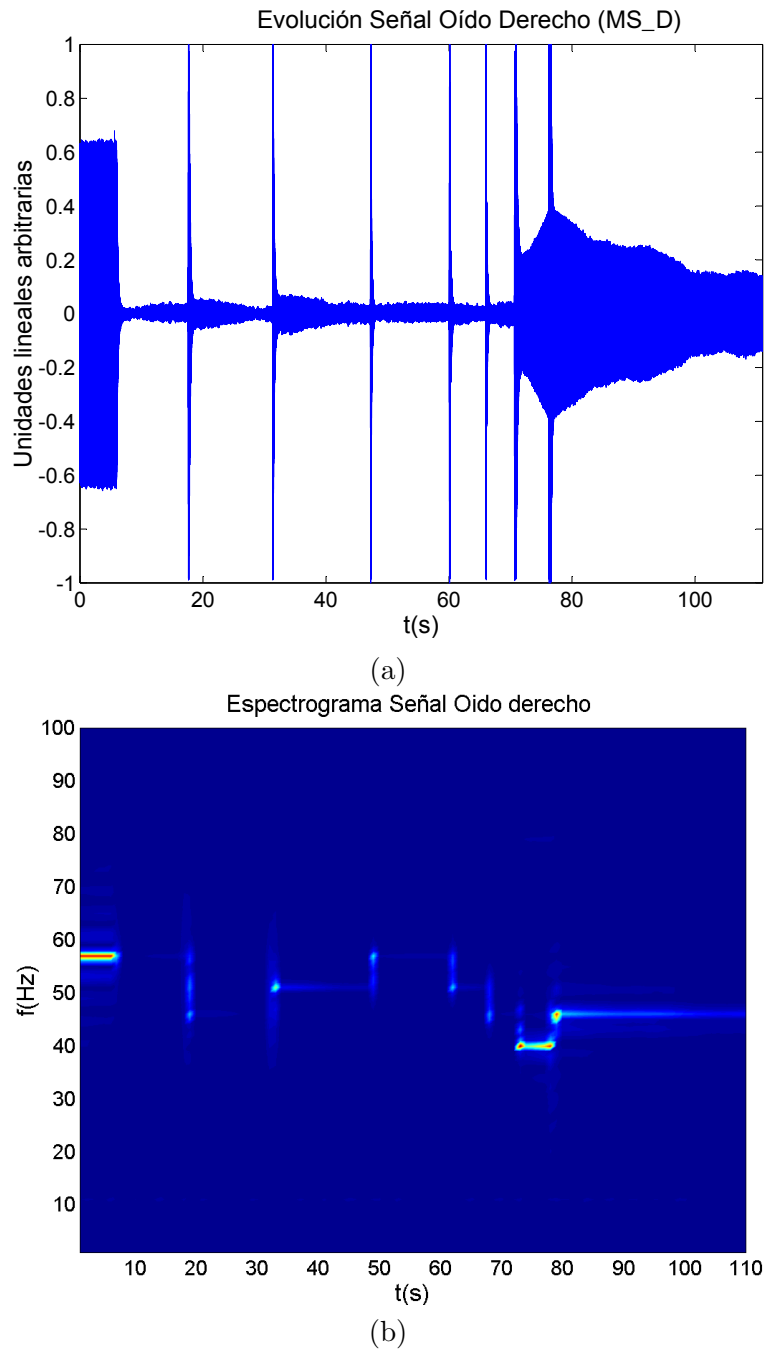


Figura 7.41: Señal captada en el oído derecho (MS_D) del maniquí usando el algoritmo de proyección afín ($N = 2$) cuando la señal a cancelar es un tono variante con la frecuencia. (a) Evolución en el tiempo. (b) Espectrograma.

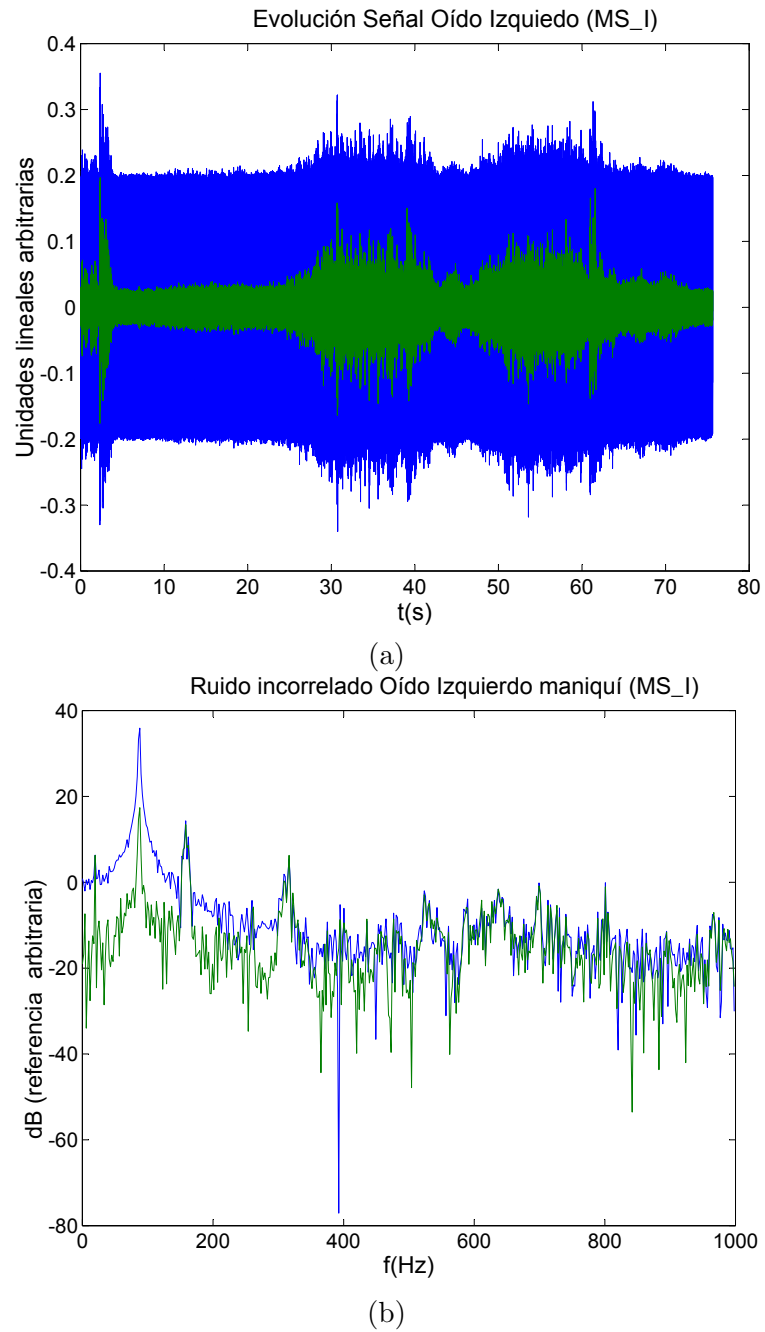


Figura 7.42: Señal captada en el oído izquierdo maniquí usando el algoritmo de proyección afín ($N = 2$) cuando la señal a cancelar es un tono al que se le añade ruido incorrelado. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.

7.4. Coste computacional real

Para estimar la carga computacional real del DSP al ejecutar las diferentes versiones de los algoritmos de control programados, se ha hecho uso de las utilidades que ofrece el Software Code Composer Studio [88] (de Texas Instrument), midiendo el número de ciclos de reloj que el DSP necesita para ejecutar un determinado código.

En la tabla 7.2, se resume el número de ciclos y frecuencia máxima del sonido que teóricamente podríamos cancelar para diferentes órdenes de proyección en un sistema 1:2:2 de CAR como el usado para nuestro experimento.

Hay que considerar que estas limitaciones vienen dadas por la carga computacional del algoritmo, que como se ha demostrado anteriormente, para los algoritmos de proyección afín, es creciente con el orden de proyección. El periodo de muestreo mínimo (frecuencia de muestreo máxima) al que podremos trabajar vendría dado por el tiempo de ejecución máximo de una iteración de nuestro algoritmo, lo que se puede calcular a partir del número de ciclos empleados para ejecutar el código de la iteración más lenta (peor de los casos) de nuestro algoritmo. Es decir, el DSP debe tener suficiente tiempo entre dos interrupciones sucesivas como para ejecutar todo el código de atención a la interrupción que ejecuta las instrucciones pertinentes de nuestro algoritmo. Fijada esa frecuencia de muestreo, hemos de tener cuidado con que las señales de audio a cancelar no superen la frecuencia de Nyquist (mitad de la frecuencia de muestreo) si queremos que no se produzca aliasing ni efectos de no linealidades.

Como podíamos imaginar, el cálculo óptimo de la actualización de los coeficientes mediante el uso de los coeficientes auxiliares, ahorra algunos ciclos de reloj, lo que permitiría al algoritmo trabajar a frecuencias de muestreo algo superiores, controlando señales acústicas de frecuencias más altas.

N	Número de ciclos max. por iter. (FXAP-CA)	Max. f. señal a cancelar
2	16279	863.8 Hz
3	16482	853.2 Hz
4	17444	806.2 Hz
5	17623	798.0 Hz

N	Número de ciclos max. por iter. (FXAP)	Max. f. señal a cancelar
2	16485	853.0 Hz
3	17610	798.6 Hz
4	18024	780.2 Hz
5	18152	774.7 Hz

Tabla 7.2: Número de ciclos por iteración y frecuencia máxima posible de la señal a cancelar para los algoritmos de proyección afín en un sistema multicanal de control activo de ruido 1:2:2 usando diferentes órdenes de proyección.

Capítulo 8

Conclusiones y líneas futuras

8.1. Conclusiones

El objetivo inicial de esta tesis doctoral era profundizar en los algoritmos adaptativos existentes que se podían usar en aplicaciones de control local multicanal de ruido acústico, prestando especial atención a su coste computacional, ya que el fin último de estos algoritmos ha de ser una implementación práctica en aplicaciones de tiempo real. Puesto que el coste computacional suele ser inversamente proporcional a las prestaciones que ofrecen los algoritmos, conviene encontrar un factor de compromiso entre dichas prestaciones y la complejidad de los algoritmos, intentando encontrar los algoritmos que ofrecen los mejores resultados con un coste computacional menor. Considerando los algoritmos existentes, en un extremo (en el de la sencillez y simplicidad con bajo coste computacional) se encuentran los algoritmos derivados del LMS, mientras que en el otro (alta demanda computacional y velocidades de convergencia rápidas) los algoritmos derivados del RLS. Los algoritmos de proyección afín parecen responder a ese factor de compromiso buscado, puesto que pueden poseer velocidades de convergencia elevadas (como los RLS) manteniendo las buenas propiedades de estabilidad y sencillez de los LMS, con un coste computacional moderado. Por eso estos algoritmos se han convertido en el objetivo de nuestro estudio.

Aunque los algoritmos de proyección afín llevan estudiándose y aplicándose a problemas similares a los planteados en esta tesis doctoral desde los años 90 (véase [63], [62] o [89]), en este trabajo hemos querido hacer una reflexión sobre las estrategias que se han venido usando para la optimización del coste computacional de estos algoritmos, particularizados en una aplicación concreta: la de control activo de ruido. Esta aplicación particular, no quita generalidad a los resultados alcanzados, puesto que en todo caso es una aplicación que añade

dificultades adicionales (como la presencia de caminos acústicos) al resto de aplicaciones de control adaptativo. Como resultado de esta reflexión, en este trabajo se presenta una novedosa estrategia donde se usa como algoritmo de control el de proyección afín, pero basado en la estructura convencional de filtrado-x, en vez de en la estructura modificada, que había sido la estructura natural usada por dichos algoritmos para aplicaciones de CAR.

En el capítulo 4 se ha podido comprobar como la estructura convencional de filtrado-x es una alternativa válida a las estrategias eficientes usadas con la estructura modificada pero con un coste computacional menor. Por tanto, el uso de la estructura convencional de filtrado-x junto con los algoritmos de proyección afín empleados para control activo de ruido, es una alternativa eficiente a las estrategias *fast* desarrolladas para reducir el coste computacional de los algoritmos de proyección afín. Aun cuando este algoritmo ha sido presentado en la literatura reciente como uno de los que mejores propiedades presentan, dada su estabilidad y su velocidad de convergencia, el hecho de tener que ser implementados con la estructura modificada hace que para el caso de sistemas multicanal y órdenes de proyección altos, el coste computacional sea también demasiado elevado, por lo que son necesarias las técnicas *fast* para reducirlo. El empleo de estas técnicas se basa en manipulaciones algebraicas y de datos para realizar las mismas operaciones con un coste computacional menor. Sin embargo, el cálculo eficiente del vector de error, introduce ligeras aproximaciones que empeoran el rendimiento de este algoritmo. Aunque la estructura convencional de filtrado-x es menos eficiente que la modificada en cuanto a la velocidad de convergencia, resulta más atractiva desde el punto de vista computacional, y podría entenderse como otra forma de aproximar dicho vector de error. Los resultados obtenidos mediante simulación así lo reflejan. Indudablemente, el uso de la estructura modificada consigue los mejores resultados en cuanto a convergencia pero si buscamos la implementación óptima de estos algoritmos al problema del control activo de ruido, observamos como al reducir el coste computacional con las estrategias eficientes también se empeora la velocidad de convergencia. Por eso resulta interesante cuestionar si usando la estructura convencional de filtrado-x se obtienen resultados comparables a los obtenidos con los algoritmos de proyección afín (aun cuando es menos eficiente que la modificada y no es posible obtener el vector de error a minimizar $\mathbf{e}_N[n]$). Según se ha comprobado mediante simulación, los resultados obtenidos, sin llegar a ser tan buenos como los de la estructura modificada, sí que son mejores que los que se logran cuando se aplican las estrategias *fast* a dicha estructura con un coste computacional inferior. Además, también es posible optimizar aún más el coste computacional aplicando las estrategias *fast* del cálculo del vector de error normalizado y del uso de los coeficientes auxiliares junto con la estructura de filtrado-x sin que esto afecte a la velocidad de convergencia.

Por tanto, el algoritmo de proyección afín basado en la estructura convencional de filtrado-x presenta un coste computacional menor que su homónimo basado en la estructura modificada aunque claro está que su velocidad de convergencia será menor, sobre todo si tratamos de controlar sistemas cuyos caminos acústicos introducen retardos largos. Sin embargo, si lo comparamos con el algoritmo AP basado en la estructura modificada y con estrategias eficientes, con costes computacionales menores que el original pero nunca inferiores a nuestra versión basada en la estructura convencional de filtrado-x, nuestra propuesta ofrece prestaciones al menos comparables a las ofrecidas por dichas versiones óptimas. Tan solo en el caso extremo de caminos secundarios con elevadísimos retardos, la mejora del coste computacional se ve penalizada por un decremento significativo en la velocidad de convergencia. No obstante, tales retardos no suelen encontrarse en los sistemas prácticos reales.

Puesto que el algoritmo propuesto mejora la complejidad computacional de los existentes sin introducir pérdidas significativas en las propiedades de convergencia, el siguiente paso ha sido analizar sus propiedades, centrándonos en el estado estacionario, concluyendo que con el algoritmo de proyección afín basado en la estructura convencional de filtrado-x se obtienen valores del error cuadrático medio en régimen estacionario tan buenos o mejores que los ofrecidos por los algoritmos basados en la estructura modificada. Además, se ha desarrollado un modelo matemático que nos proporciona una estima de este valor del error cuadrático medio en régimen estacionario.

La extensión multicanal de este algoritmo también era uno de nuestros objetivos iniciales puesto que se pretendía obtener algoritmos eficientes multicanal de control local. Para dicha extensión, nos hemos basado en la misma estrategia que nos permite pasar del algoritmo LMS de filtrado-x al MELMS de filtrado-x [90]. Precisamente en las versiones multicanal del algoritmo propuesto es donde el ahorro computacional alcanzado es más significativo. Esta versión multicanal ha sido analizada igualmente en estado estacionario, elaborando un modelo que proporciona el valor del error cuadrático medio multicanal.

Por último, el algoritmo ha sido implementado en una aplicación de control activo de ruido multicanal local real, basada en un DSP comercial, comprobando que los resultados procedentes del análisis previo (tanto mediante simulación como mediante estudio teórico) también se alcanzan en un sistema práctico, y sobre todo, que el coste computacional del algoritmo es abordable por los sistemas hardware existentes.

8.2. Líneas futuras

El inicio de una línea de investigación da lugar a diferentes frentes abiertos imposibles de abordar en una única tesis doctoral. Este trabajo no ha sido una excepción, y otras alternativas de investigación han quedado pendientes como posibles estudios futuros, de entre las que se pueden destacar:

- Tal y como se expone en [79], el algoritmo de proyección afín puede ser considerado como uno más dentro de una familia de algoritmos cuya característica común es que siguen una regla de actualización para los coeficientes de los filtros adaptativos como la definida en la ecuación (3.34). Aunque durante esta tesis doctoral nos hemos centrado en el estudio del algoritmo de proyección afín (caso particular de dicha expresión genérica cuya ecuación de actualización de los coeficientes viene dada por la ecuación (3.33)), el esfuerzo por optimizar las otras variantes de la familia de algoritmos que se desprenden de (3.34) podría ser abordado en trabajos futuros, extendiendo los resultados obtenidos en la presente tesis doctoral, y comparando las prestaciones de cada una de las diferentes variantes.
- Por otro lado, cuando los algoritmos de proyección afín se aplican al CAR, hemos considerado que la estimación de los caminos acústicos secundarios necesarios para el buen funcionamiento de dichos algoritmos era sin error (los caminos estimados eran iguales a los acústicos). En las aplicaciones prácticas podemos usar sistemas de identificación muy precisos que nos proporcionen la respuesta de una planta acústica con un error pequeño y acotado, por lo que ciertamente pueden considerarse aproximadamente iguales. Sería interesante estudiar como afecta ese error en las estimaciones de los caminos acústicos al comportamiento y prestaciones de estos algoritmos.
- Los algoritmos eficientes propuestos consiguen optimizar la eficiencia computacional, aunque como se desprende de los resultados de simulación, es a costa de la velocidad de convergencia. El transitorio de estos algoritmos basados en la estructura de filtrado-x convencional es más largo (parámetro de convergencia menor para que no se produzca divergencia), sobre todo cuando el retardo de los caminos acústicos es grande. Una posible línea futura podría ser intentar modelizar el transitorio de estos algoritmos y acotar el parámetro de convergencia óptimo que maximice la velocidad de convergencia de los mismo, a partir de los estadísticos de las señales que intervienen en ellos y de las características físicas del sistema donde se aplican estos algoritmos (retardo de los cami-

nos acústicos). Una primera aproximación del modelado del transitorio ya se presenta en el anexo A.

- Para la actualización de los coeficientes en los algoritmos multicanal desarrollados en este trabajo se ha usado la expresión (5.83), que resulta de la adaptación natural del algoritmo Multiple Error LMS (MELMS) descrito en [90], cuando el algoritmo de control es el de proyección afín. Sin embargo, dicha actualización multicanal admite otras estrategias como las definidas en [83]:

$$\mathbf{w}[n] = \mathbf{w}[n-1] - \mu \mathbf{U}[n](\mathbf{U}^T[n]\mathbf{U}[n] + \delta \mathbf{I})^{-1} \mathbf{e}[n], \quad (8.1)$$

$$\mathbf{w}[n] = \mathbf{w}[n-1] - \mu \sum_{k=1}^K \mathbf{V}_k[n](\mathbf{V}_k^T[n]\mathbf{V}_k[n] + \delta \mathbf{I})^{-1} \mathbf{e}_k[n], \quad (8.2)$$

dando lugar a diferentes algoritmos multicanal basados en los de proyección afín. Se podrían definir otras muchas formas de actualizar los coeficientes en sistemas multicanal, como por ejemplo, usando un vector de error normalizado común con todas las aportaciones de los sensores a cada filtro adaptativo:

$$\epsilon_{ij}[n] = \sum_{k=1}^K (\mathbf{V}_{ijk}^T[n]\mathbf{V}_{ijk}[n] + \delta \mathbf{I})^{-1} \mathbf{e}_k[n] \quad (8.3)$$

$$\mathbf{w}_{ij}[n] = \mathbf{w}_{ij}[n-1] - \mu \sum_{k=1}^K \mathbf{V}_{ijk}[n] \epsilon_{ij}[n] \quad (8.4)$$

Si usamos estrategias eficientes (como la de cálculo recursivo) para calcular el vector de error normalizado, con esta estrategia para la actualización de los coeficientes, optimizaríamos aún más el coste computacional, puesto que pasaríamos de tener que calcular IJK vectores de error normalizados, a solo IJ .

Así, una posible línea futura de investigación puede consistir en analizar el comportamiento de cada una de las diferentes versiones de actualización multicanal, en los algoritmos de proyección afín desarrollados en este trabajo.

- Además, respecto a la extensión multicanal de los algoritmos estudiados, existen otras estrategias que pueden optimizar aún más el coste computacional de los algoritmos usando en cada iteración la información de un único sensor y no la de todos como se propone en las estrategias comentadas en el punto anterior. De entre estas estrategias se pueden mencionar

las denominadas *Scanning* [91] (se van usando secuencialmente la información de cada sensor de error en cada iteración) y *Minimax* [92] (se escoge en cada iteración, el sensor cuyo error sea mayor). El estudio de los algoritmos multicanal descritos en esta tesis, podría también abordarse cuando las estrategias de actualización usadas sean éstas.

- Los algoritmos descritos son susceptibles de mejoras, como por ejemplo, el uso de un parámetro de convergencia variable. Es decir se podría adaptar el algoritmo LMS de paso variable [24], donde el parámetro de paso se autoajusta para optimizar la velocidad de convergencia o el *MSE* final dependiendo de las necesidades en cada momento. Algunos trabajos sobre este aspecto han sido propuestos en [93], [94], [95] ó [96] donde además se tiene en cuenta el incremento del coste computacional debido a los mecanismos de control para la variación del parámetro de convergencia, proponiendo alternativas para mejorar la eficiencia computacional junto con el uso del parámetro de convergencia variable. El estudio y análisis de los algoritmos de proyección afín con estructura de filtrado-x convencional y paso variable podría ser también objeto de investigación en el futuro.
- En la misma línea se podría desarrollar un algoritmo de proyección afín multi-orden, donde el orden del algoritmo (que controla la velocidad de convergencia, y determina en gran medida la complejidad computacional) puede ir variando atendiendo a las necesidades en cada instante (ahorro computacional, velocidad de convergencia, mínimo MSE,...).
- El uso de algoritmos funcionando en paralelo, en la línea de lo expuesto en [97] y [98] podría ser aplicado cuando uno o los dos algoritmos son del tipo de los desarrollados en este trabajo. De esta forma, se podrían conjuntar las mejores propiedades de los dos algoritmos usados con la finalidad de obtener un comportamiento global mejor que el que se obtendría con cada uno por separado. Una posible idea sería usar dos algoritmos de proyección afín de órdenes diferentes (uno de orden alto con rápida velocidad de convergencia y otro de orden bajo con mejores propiedades en régimen estacionario) y que conmuten el control total o el peso que tiene cada algoritmo en el comportamiento del algoritmo global en función de las necesidades de velocidad de convergencia o mínimo MSE.
- Finalmente, las dos últimas ideas podrían aplicarse a la vez teniendo algoritmos de proyección afín funcionando en paralelo cuyo orden de proyección pueda ser ajustable a lo largo del tiempo.

Apéndice A

Comportamiento en régimen transitorio de los algoritmos de proyección afín.

A.1. Introducción

Si en el capítulo 5 hemos obtenido información acerca del comportamiento de los algoritmos en su estado estacionario, en el presente capítulo nos centraremos en el transitorio. Aunque los transitorios de los algoritmos de proyección afín pueden analizarse mediante simulación, sería interesante obtener un modelo matemático que nos diera información de la evolución de los mismos a partir de los datos estadísticos de las señales de referencia y las condiciones de funcionamiento del algoritmo. Además, el modelo podría facilitarnos información útil referente a la convergencia y estabilidad de dichos algoritmos, lo que permitirá comparar teóricamente sus prestaciones (sobre todo en cuanto a sus velocidades de convergencia).

El estimador que mejor información nos proporcionará del transitorio será la evolución temporal de la varianza del error a priori $E\{|e_v^a[n]|^2\}$. Dicha evolución viene descrita por lo que se conoce como curvas de aprendizaje y matemáticamente (considerando que los vectores de datos $\mathbf{v}_L[n]$ son independientes e idénticamente distribuidos) vendrían dadas por la expresión:

$$E\{|e_v^a[n]|^2\} = E\{|\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]|^2\} = E\{|\tilde{\mathbf{w}}_L^T[n-1]\mathbf{v}_L[n]\mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1]|\}, \quad (\text{A.1})$$

donde $\mathbf{v}_L[n]$ sería el vector de tamaño L con las últimas L muestras de la señal de referencia filtrada por la estimación del camino secundario, que ya

ha aparecido anteriormente en la definición de $\mathbf{V}[n]$ en la ecuación (5.48), y $\tilde{\mathbf{w}}_L[n]$ se corresponde con el vector de pesos definido en (5.4). Para operar con la expresión descrita en A.1, conviene introducir la definición del cálculo de la norma euclídea ponderada de un vector que, dada una matriz Σ (simétrica y de tamaño adecuado), viene descrita según:

$$\|\mathbf{w}\|_{\Sigma}^2 = \mathbf{w}^T \Sigma \mathbf{w}, \quad (\text{A.2})$$

por lo que la expresión definida en (A.1), puede reescribirse usando la definición expresada en (A.2)

$$E\{|e_v^a[n]|^2\} = E\{\|\tilde{\mathbf{w}}_L^T[n-1]\|_{E\{\mathbf{v}_L[n]\mathbf{v}_L^T[n]\}}^2\} = E\{\|\tilde{\mathbf{w}}_L^T[n-1]\|_{\mathbf{R}_v}^2\}, \quad (\text{A.3})$$

siendo \mathbf{R}_v la matriz de covarianza (definida positiva) de la variable aleatoria $\mathbf{v}_L[n]$.

Veamos ahora cómo conseguir una relación iterativa para la expresión definida en (A.3).

A.2. Análisis del transitorio del algoritmo de proyección afín con la estructura de filtrado-x modificada aplicado al CAR

Para el estudio del transitorio de los algoritmos vamos a usar, de nuevo, la relación de la conservación de la energía que ya se expuso para el análisis del estado estacionario, pero en este caso, con una ligera variación puesto que introduciremos la norma ponderada definida en (A.2). Partiremos, otra vez, de la ecuación de actualización de los pesos para nuestro algoritmo (ecuación (5.51)), de la que se puede derivar la expresión:

$$\tilde{\mathbf{w}}_L[n] = \tilde{\mathbf{w}}_L[n-1] - \mu \mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta I)^{-1} \mathbf{e}_N[n]. \quad (\text{A.4})$$

Si definimos unos nuevos vectores de error a priori y posteriori ponderados (de tamaño igual al orden de proyección N) como:

$$\mathbf{e}_{\Sigma}^a[n] = \mathbf{V}[n]\Sigma\tilde{\mathbf{w}}_L[n-1] \quad (\text{A.5})$$

$$\mathbf{e}_{\Sigma}^p[n] = \mathbf{V}[n]\Sigma\tilde{\mathbf{w}}_L[n], \quad (\text{A.6})$$

al multiplicar la expresión (A.4) por la izquierda por $\mathbf{V}[n]\Sigma$, se obtiene:

$$\mathbf{e}_\Sigma^p[n] = \mathbf{e}_\Sigma^a[n] + \mu \mathbf{V}[n]\Sigma \mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1} \mathbf{e}_N[n]. \quad (\text{A.7})$$

De aquí, podemos despejar $(\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1} \mathbf{e}_N[n]$ e introducirlo en la expresión (A.4), obteniendo la siguiente igualdad:

$$\tilde{\mathbf{w}}_L[n] = \tilde{\mathbf{w}}_L[n-1] + \mathbf{V}^T[n](\mathbf{V}[n]\Sigma \mathbf{V}^T[n])^{-1}(\mathbf{e}_\Sigma^p[n] - \mathbf{e}_\Sigma^a[n]) \quad (\text{A.8})$$

que reordenada adecuadamente, puede escribirse como:

$$\tilde{\mathbf{w}}_L[n] + \mathbf{V}^T[n](\mathbf{V}[n]\Sigma \mathbf{V}^T[n])^{-1} \mathbf{e}_\Sigma^a[n] = \tilde{\mathbf{w}}_L[n-1] + \mathbf{V}^T[n](\mathbf{V}[n]\Sigma \mathbf{V}^T[n])^{-1} \mathbf{e}_\Sigma^p[n]. \quad (\text{A.9})$$

Si calculamos, ahora, la norma ponderada de ambos lados de la igualdad, obtendremos:

$$\begin{aligned} & \|\tilde{\mathbf{w}}_L[n]\|_\Sigma^2 + (\mathbf{e}_\Sigma^a[n])^T (\mathbf{V}[n]\Sigma \mathbf{V}^T[n])^{-1} \mathbf{e}_\Sigma^a[n] \\ &= \|\tilde{\mathbf{w}}_L[n-1]\|_\Sigma^2 + (\mathbf{e}_\Sigma^p[n])^T (\mathbf{V}[n]\Sigma \mathbf{V}^T[n])^{-1} \mathbf{e}_\Sigma^p[n], \end{aligned} \quad (\text{A.10})$$

que es lo que se conoce como ecuación de conservación de la energía ponderada y no es más que una extensión de la ecuación de conservación de la energía definida en (5.12), que se cumpliría para el caso en que Σ fuera la matriz identidad. Al igual que se realizó en el capítulo anterior, continuamos desarrollando la expresión (A.10) usando la definición del error a posteriori ponderado descrita en (A.7). Así, obtendremos:

$$\begin{aligned} \|\tilde{\mathbf{w}}_L[n]\|_\Sigma^2 &= \|\tilde{\mathbf{w}}_L[n-1]\|_\Sigma^2 + \mu^2 \mathbf{e}_N^T[n] \mathbf{A}_\Sigma[n] \mathbf{e}_N[n] \\ &+ \mu \mathbf{e}_N^T[n] (\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1} \mathbf{e}_\Sigma^a[n] + \mu (\mathbf{e}_\Sigma^a[n])^T (\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1} \mathbf{e}_N[n], \end{aligned} \quad (\text{A.11})$$

donde $\mathbf{A}_\Sigma[n] = (\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1} \mathbf{V}[n]\Sigma \mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1}$.

Considerando que $\mathbf{e}_N[n] = -\mathbf{e}^a[n] + \mathbf{r}_N[n]$, con $\mathbf{e}^a[n] = \mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1]$, y teniendo en cuenta la independencia estadística de las señales de ruido y de datos, al tomar esperanzas matemáticas obtenemos:

$$\begin{aligned} E\{\|\tilde{\mathbf{w}}_L[n]\|_\Sigma^2\} &= E\{\|\tilde{\mathbf{w}}_L[n-1]\|_\Sigma^2\} \\ &- \mu E\{(\mathbf{e}^a[n])^T (\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1} \mathbf{e}_\Sigma^a[n]\} \\ &- \mu E\{(\mathbf{e}_\Sigma^a[n])^T (\mathbf{V}[n]\mathbf{V}^T[n] + \delta \mathbf{I})^{-1} \mathbf{e}^a[n]\} \\ &+ \mu^2 E\{(\mathbf{e}^a[n])^T \mathbf{A}_\Sigma[n] \mathbf{e}^a[n]\} + \mu^2 E\{\mathbf{r}_N^T[n] \mathbf{A}_\Sigma[n] \mathbf{r}_N[n]\}, \end{aligned} \quad (\text{A.12})$$

que introduciendo las definiciones de $\mathbf{e}^a[n]$ y $\mathbf{e}_\Sigma^a[n]$ nos conduce a la expresión:

$$\begin{aligned} E\{\|\tilde{\mathbf{w}}_L[n]\|_\Sigma^2\} &= E\{\|\tilde{\mathbf{w}}_L[n-1]\|_\Sigma^2\} \\ &- \mu E\{\tilde{\mathbf{w}}_L^T[n-1]\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\Sigma\tilde{\mathbf{w}}_L[n-1]\} \\ &- \mu E\{\tilde{\mathbf{w}}_L^T[n-1]\Sigma\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1]\} \\ &+ \mu^2 E\{\tilde{\mathbf{w}}_L^T[n-1]\mathbf{V}^T[n]\mathbf{A}_\Sigma[n]\mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1]\} + \mu^2 E\{\mathbf{r}_N^T[n]\mathbf{A}_\Sigma[n]\mathbf{r}_N[n]\}. \end{aligned} \quad (\text{A.13})$$

Dicha expresión, por la propiedad de superposición de la norma ponderada (que como puede comprobarse en [80]), garantiza que $a_1\|\mathbf{w}\|_{\Sigma_1}^2 + a_2\|\mathbf{w}\|_{\Sigma_2}^2 = \|\mathbf{w}\|_{a_1\Sigma_1+a_2\Sigma_2}^2$, puede escribirse como:

$$E\{\|\tilde{\mathbf{w}}_L[n]\|_\Sigma^2\} = E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\Sigma'}^2\} + \mu^2 E\{\mathbf{r}_N^T[n]\mathbf{A}_\Sigma[n]\mathbf{r}_N[n]\}, \quad (\text{A.14})$$

con

$$\begin{aligned} \Sigma' &= \Sigma - \mu E\{\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\}\Sigma \\ &- \mu\Sigma E\{\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\} + \mu^2 E\{\mathbf{V}^T[n]\mathbf{A}_\Sigma[n]\mathbf{V}[n]\}. \end{aligned} \quad (\text{A.15})$$

En (A.15), se ha supuesto independencia estadística entre $\|\tilde{\mathbf{w}}_L[n-1]\|$ y $\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]$, para poder introducir el valor medio en la expresión de Σ' . Precisamente llamando $\mathbf{P}[n]$ a esta última expresión, se puede reescribir de una forma más compacta la expresión (A.15), resultando:

$$\Sigma' = \Sigma - \mu E\{\mathbf{P}[n]\}\Sigma - \mu\Sigma E\{\mathbf{P}[n]\} + \mu^2 E\{\mathbf{P}[n]\Sigma\mathbf{P}[n]\}. \quad (\text{A.16})$$

Para poder seguir manipulando dicha expresión, introducimos adecuadamente la matriz identidad en la misma:

$$\Sigma' = \mathbf{I}\Sigma\mathbf{I} - \mu E\{\mathbf{P}[n]\}\Sigma\mathbf{I} - \mu\mathbf{I}\Sigma E\{\mathbf{P}[n]\} + \mu^2 E\{\mathbf{P}[n]\Sigma\mathbf{P}[n]\}. \quad (\text{A.17})$$

Si la igualdad anterior se cumple matricialmente, también se cumplirá para cada elemento de las matrices. Así, si relocalamos los elementos de las matrices de forma que el operador $\text{vec}(\cdot)$ aplicado a una matriz $M \times M$, la transforme en un vector columna de dimensiones $M^2 \times 1$, tendremos

$$\begin{aligned} &\text{vec}(\Sigma') \\ &= \text{vec}(\mathbf{I}\Sigma\mathbf{I}) - \mu\text{vec}(E\{\mathbf{P}[n]\}\Sigma\mathbf{I}) - \mu\text{vec}(\mathbf{I}\Sigma E\{\mathbf{P}[n]\}) + \mu^2\text{vec}(E\{\mathbf{P}[n]\Sigma\mathbf{P}[n]\}). \end{aligned} \quad (\text{A.18})$$

además, para cualquier conjunto de matrices $\{A, B \text{ y } C\}$ de dimensiones adecuadas, se cumple que:

$$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B}), \quad (\text{A.19})$$

designando el operador \otimes , como al producto de Kronecker. De esta forma, podemos reescribir la ecuación (A.18), según:

$$\begin{aligned} \text{vec}(\Sigma') = & (\mathbf{I}^T \otimes \mathbf{I})\text{vec}(\Sigma) - \mu(\mathbf{I} \otimes E\{\mathbf{P}[n]\})\text{vec}(\Sigma) \\ & - \mu(E\{\mathbf{P}[n]^T\} \otimes \mathbf{I})\text{vec}(\Sigma) + \mu^2(E\{\mathbf{P}[n]^T \otimes \mathbf{P}[n]\})\text{vec}(\Sigma), \end{aligned} \quad (\text{A.20})$$

donde sacando factor común $\text{vec}(\Sigma)$ en el segundo término de la igualdad, y agrupando al resto de términos en la matriz \mathbf{F} , descrita en (A.21),

$$\mathbf{F} = \mathbf{I}^T \otimes \mathbf{I} - \mu\mathbf{I} \otimes E\{\mathbf{P}[n]\} - \mu E\{\mathbf{P}[n]^T\} \otimes \mathbf{I} + \mu^2 E\{\mathbf{P}[n]^T \otimes \mathbf{P}[n]\}, \quad (\text{A.21})$$

se obtiene la igualdad:

$$\text{vec}(\Sigma') = \mathbf{F}\text{vec}(\Sigma). \quad (\text{A.22})$$

Llamando $\sigma = \text{vec}(\Sigma)$ y $\sigma' = \text{vec}(\Sigma')$, la igualdad anterior se transforma en

$$\sigma' = \mathbf{F}\sigma. \quad (\text{A.23})$$

Considerando ahora, que el operador $\text{vec}^{-1}(\cdot)$, aplicado a un vector de tamaño $M^2 \times 1$, lo transforma en una matriz $M \times M$, podemos reescribir la ecuación (A.14)

$$E\{\|\tilde{\mathbf{w}}_L[n]\|_{\text{vec}^{-1}(\sigma')}^2\} = E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\text{vec}^{-1}(\sigma')}^2\} + \mu^2 E\{\mathbf{r}_N^T[n]\mathbf{A}_\Sigma[n]\mathbf{r}_N[n]\}, \quad (\text{A.24})$$

donde el último elemento del segundo término puede simplificarse, de igual modo que se hizo en (5.37), como:

$$E\{\mathbf{r}_N^T[n]\mathbf{A}_\Sigma[n]\mathbf{r}_N[n]\} = E\{\text{Tr}(\mathbf{r}_N[n]\mathbf{r}_N[n]^T\mathbf{A}_\Sigma[n])\} = \sigma_r^2 \text{Tr}(E\{\mathbf{A}_\Sigma[n]\}), \quad (\text{A.25})$$

que teniendo en cuenta el valor de $\mathbf{A}_\Sigma[n]$ que se desprende de la ecuación (A.11), podemos simplificar más, puesto que:

$$\begin{aligned} & \text{Tr}(E\{(\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\Sigma\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\}) \\ &= \text{Tr}(E\{\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}(\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\Sigma\}) \\ &= \text{vec}(E\{\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}(\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\})^T \text{vec}(\Sigma) = \gamma^T \sigma, \end{aligned} \quad (\text{A.26})$$

donde $\gamma = \text{vec}(E\{\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-2}\mathbf{V}[n]\})$.

De esta forma, podemos reescribir la ecuación (A.24)

$$E\{\|\tilde{\mathbf{w}}_L[n]\|_{\text{vec}^{-1}(\sigma)}^2\} = E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\text{vec}^{-1}(\mathbf{F}\sigma)}^2\} + \mu^2\sigma_r^2(\gamma^T\sigma). \quad (\text{A.27})$$

Para tratar de encontrar una expresión recursiva en (A.27), si sustituimos el vector σ , por $\mathbf{F}\sigma$, obtendremos:

$$E\{\|\tilde{\mathbf{w}}_L[n]\|_{\text{vec}^{-1}(\mathbf{F}\sigma)}^2\} = E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\text{vec}^{-1}(\mathbf{F}^2\sigma)}^2\} + \mu^2\sigma_r^2(\gamma^T\mathbf{F}\sigma), \quad (\text{A.28})$$

por lo que la ecuación (A.27) se puede reescribir como:

$$E\{\|\tilde{\mathbf{w}}_L[n]\|_{\text{vec}^{-1}(\sigma)}^2\} = E\{\|\tilde{\mathbf{w}}_L[n-2]\|_{\text{vec}^{-1}(\mathbf{F}^2\sigma)}^2\} + \mu^2\sigma_r^2(\gamma^T\sigma) + \mu^2\sigma_r^2(\gamma^T\mathbf{F}\sigma). \quad (\text{A.29})$$

Y así, operando iterativamente, podemos llegar a la expresión:

$$\begin{aligned} & E\{\|\tilde{\mathbf{w}}_L[n]\|_{\text{vec}^{-1}(\sigma)}^2\} \\ &= E\{\|\tilde{\mathbf{w}}_L[0]\|_{\text{vec}^{-1}(\mathbf{F}^n\sigma)}^2\} + \mu^2\sigma_r^2\gamma^T(\mathbf{I}\sigma + \mathbf{F}\sigma + \mathbf{F}^2\sigma + \dots + \mathbf{F}^{n-1}\sigma) \end{aligned} \quad (\text{A.30})$$

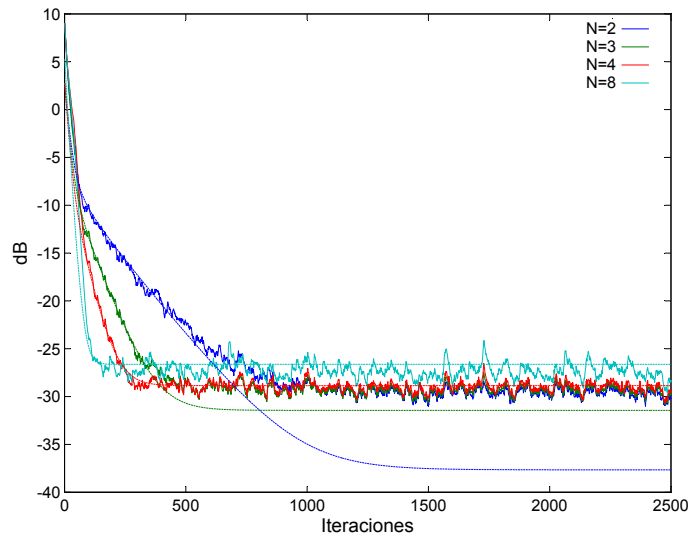
que se puede reescribir de una forma recursiva como

$$E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\text{vec}^{-1}(\sigma)}^2\} = E\{\|\tilde{\mathbf{w}}_L[0]\|_{\text{vec}^{-1}(\mathbf{f}[n])}^2\} + \mu^2\sigma_r^2g[n] \quad (\text{A.31})$$

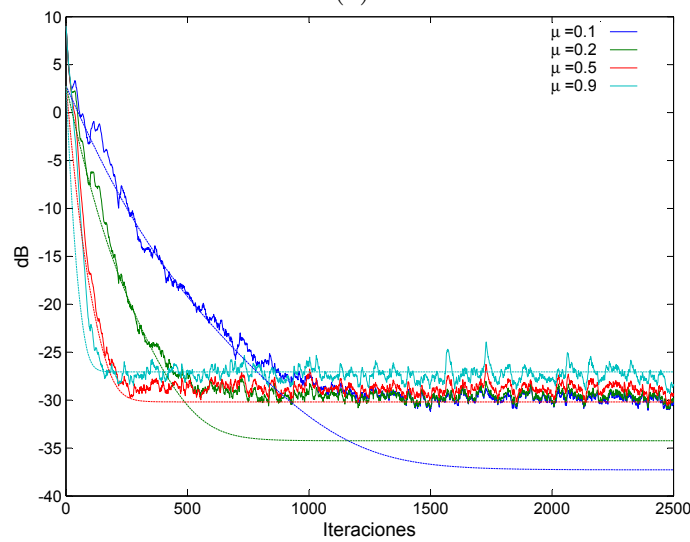
con $\mathbf{f}[n] = \mathbf{F}\mathbf{f}[n-1]$, $\mathbf{f}[0] = \sigma$, $g[n] = g[n-1] + \gamma^T\mathbf{f}[n]$ y $g[0] = 0$.

Particularizando la expresión (A.31) para $\Sigma = \mathbf{R}_v$, o lo que es lo mismo, $\sigma = \text{vec}(E\{\mathbf{v}_L[n]\mathbf{v}_L^T[n]\})$, obtenemos la expresión recursiva buscada para $E\{|e_v^a[n]|^2\}$, según se desprende de (A.3).

De esta forma, tendríamos resuelto el cálculo de las curvas de aprendizaje y por tanto, la evolución teórica del transitorio, para el algoritmo de proyección afín basado en la estructura modificada. Para validar el modelo, se han realizado simulaciones para un sistema con un filtro adaptativo de 16 coeficientes, 8 coeficientes para el camino secundario, y diferentes valores de μ y N . La constante de regularización se fijó en 10^{-5} . De nuevo se ha usado como señal de referencia ruido Gaussiano (de varianza unidad y media cero) coloreado tras ser filtrado por un filtro autorregresivo de primer orden con función de transferencia $\sqrt{1-a^2}/(1-az^{-1})$ con $a=0.9$. La señal deseada $\mathbf{d}[n]$ se ha generado siguiendo el modelo descrito en (5.98) usando ruido Gaussiano de varianza $\sigma_r^2 = 10^{-3}$.



(a)



(b)

Figura A.1: Curvas de la evolución del transitorio para (a) $\mu = 0.07$ y diferentes órdenes de proyección, (b) $N = 6$ y diferentes valores de μ . En trazo continuo la curva obtenida mediante simulaciones y en trazo discontinuo la obtenida mediante el modelo matemático.

En la figura A.1 se muestran comparativamente las curvas de aprendizaje teóricas y las obtenidas mediante simulación, para diferentes condiciones de funcionamiento del algoritmo. Mientras que en A.1 (a) se observan dichas curvas para diferentes órdenes de proyección manteniendo μ constante ($\mu = 0,07$), en A.1 (b), se representan para $N = 6$, variando la constante μ .

Podemos observar como el transitorio se ajusta bastante bien en todos los casos, existiendo diferencias solo en los valores finales de convergencia obtenidos por el modelo respecto a los proporcionados mediante simulación, algo mayores estas diferencias conforme la convergencia es más lenta (μ y N pequeños). Es probable que las imprecisiones cometidas en las simulaciones no permitan alcanzar las atenuaciones que teóricamente nos ofrece el modelo matemático, por lo que cuando estas atenuaciones dejan de ser tan grandes (μ o N mayores) los resultados mediante simulación se asemejan más a los proporcionados por el modelo.

A.3. Análisis del transitorio del algoritmo de proyección afín con la estructura de filtrado-x convencional aplicado al CAR

El análisis del transitorio cuando usamos la estructura de filtrado-x convencional se puede realizar de forma similar a lo mostrado con la estructura de filtrado-x modificada. Solo hay que tener en cuenta que con esta estructura el vector de error se forma con los datos de la señal en el sensor de error. Es decir:

$$\begin{aligned} \mathbf{e}_N[n] &= \mathbf{d}_N[n] + \begin{pmatrix} \mathbf{v}_L^T[n]\mathbf{w}_L[n-1] \\ \mathbf{v}_L^T[n-1]\mathbf{w}_L[n-2] \\ \vdots \\ \mathbf{v}_L^T[n-N+1]\mathbf{w}_L[n-N] \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{v}_L^T[n](\mathbf{w}_L[n-1] - \mathbf{w}_L^0) \\ \mathbf{v}_L^T[n-1](\mathbf{w}_L[n-2] - \mathbf{w}_L^0) \\ \vdots \\ \mathbf{v}_L^T[n-N+1](\mathbf{w}_L[n-N] - \mathbf{w}_L^0) \end{pmatrix} + \mathbf{r}_N[n], \end{aligned} \quad (\text{A.32})$$

siendo $\mathbf{d}_N[n] = -\mathbf{V}[n]\mathbf{w}_L^0 + \mathbf{r}_N[n]$. El vector $\mathbf{e}_N[n]$ puede escribirse, igual que en el caso anterior como $\mathbf{e}_N[n] = -\mathbf{e}^{a'}[n] + \mathbf{r}_N[n]$, siendo $\mathbf{e}^{a'}[n]$

$$\mathbf{e}^{a'}[n] = \begin{pmatrix} \mathbf{v}_L^T[n]\tilde{\mathbf{w}}_L[n-1] \\ \mathbf{v}_L^T[n-1]\tilde{\mathbf{w}}_L[n-2] \\ \vdots \\ \mathbf{v}_L^T[n-N+1]\tilde{\mathbf{w}}_L[n-N] \end{pmatrix} = \begin{pmatrix} e_v^a[n] \\ e_v^a[n-1] \\ \vdots \\ e_v^a[n-N+1] \end{pmatrix}. \quad (\text{A.33})$$

La ecuación (A.33) puede descomponerse como sigue:

$$\begin{aligned} \mathbf{e}^{a'}[n] &= \mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1] + \mathbf{V}_1[n](\tilde{\mathbf{w}}_L[n-2] - \tilde{\mathbf{w}}_L[n-1]) \\ &+ \mathbf{V}_2[n](\tilde{\mathbf{w}}_L[n-3] - \tilde{\mathbf{w}}_L[n-2]) + \dots + \mathbf{V}_{N-1}[n](\tilde{\mathbf{w}}_L[n-N] - \tilde{\mathbf{w}}_L[n-N+1]) \\ &= \mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1] + \sum_{i=1}^{N-1} \mathbf{V}_i[n](\tilde{\mathbf{w}}_L[n-(i+1)] - \tilde{\mathbf{w}}_L[n-i]) = \mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1] + \mathbf{R}[n], \end{aligned} \quad (\text{A.34})$$

siendo

$$\begin{aligned} \mathbf{V}_1[n] &= \begin{pmatrix} \mathbf{0} \\ \mathbf{v}_L^T[n-1] \\ \mathbf{v}_L^T[n-2] \\ \vdots \\ \mathbf{v}_L^T[n-N+1] \end{pmatrix} & \mathbf{V}_2[n] &= \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_L^T[n-2] \\ \vdots \\ \mathbf{v}_L^T[n-N+1] \end{pmatrix} \\ \mathbf{V}_i[n] &= \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{v}_L^T[n-i] \\ \vdots \\ \mathbf{v}_L^T[n-N+1] \end{pmatrix} \end{aligned} \quad (\text{A.35})$$

donde $\mathbf{0}$ representa un vector de ceros.

Para simplificar el análisis posterior, consideramos que $\mathbf{R}[n]$ está formada por datos de una variable aleatoria de media cero e independiente de $\tilde{\mathbf{w}}_L[n-1]$.

Teniendo en cuenta (A.34), la ecuación (A.12), sería ahora:

$$\begin{aligned} E\{\|\tilde{\mathbf{w}}_L[n]\|_{\Sigma}^2\} &= E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\Sigma}^2\} \\ &- \mu E\{(\mathbf{e}^{a'}[n])^T(\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{e}_{\Sigma}^a[n]\} \\ &- \mu E\{(\mathbf{e}_{\Sigma}^a[n])^T(\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{e}^{a'}[n]\} \\ &+ \mu^2 E\{(\mathbf{e}^{a'}[n])^T\mathbf{A}_{\Sigma}[n]\mathbf{e}^{a'}[n]\} + \mu^2 E\{\mathbf{r}_N^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{r}_N[n]\}, \end{aligned} \quad (\text{A.36})$$

y análogamente a la expresión alcanzada en (A.13) se obtiene

$$\begin{aligned}
 E\{\|\tilde{\mathbf{w}}_L[n]\|_{\Sigma}^2\} &= E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\Sigma}^2\} \\
 &- \mu E\{(\tilde{\mathbf{w}}_L^T[n-1]\mathbf{V}^T[n] + \mathbf{R}^T[n])(\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}\mathbf{V}[n]\Sigma\tilde{\mathbf{w}}_L[n-1]\} \\
 &- \mu E\{\tilde{\mathbf{w}}_L^T[n-1]\Sigma\mathbf{V}^T[n](\mathbf{V}[n]\mathbf{V}^T[n] + \delta\mathbf{I})^{-1}(\mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1] + \mathbf{R}[n])\} \\
 &+ \mu^2 E\{\tilde{\mathbf{w}}_L^T[n-1]\mathbf{V}^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{V}[n]\tilde{\mathbf{w}}_L[n-1]\} + \mu^2 E\{\mathbf{r}_N^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{r}_N[n]\} \\
 &+ \mu^2 E\{\mathbf{R}^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{R}[n]\}.
 \end{aligned} \tag{A.37}$$

Con la hipótesis considerada sobre $\mathbf{R}[n]$, podemos escribir (A.37) como:

$$\begin{aligned}
 &E\{\|\tilde{\mathbf{w}}_L[n]\|_{\Sigma}^2\} \\
 &= E\{\|\tilde{\mathbf{w}}_L[n-1]\|_{\Sigma'}^2\} + \mu^2 E\{\mathbf{r}_N^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{r}_N[n]\} + \mu^2 E\{\mathbf{R}^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{R}[n]\},
 \end{aligned} \tag{A.38}$$

donde la primera parte de la ecuación coincide con la expresión del transitorio para la estructura de filtrado-x modificada, y la última puede considerarse como la ralentización que se introduce en el transitorio debido al uso de la estructura de filtrado-x convencional. Para estudiar esta diferencia consideramos, de acuerdo con (A.34), que $\mathbf{R}[n] = \mathbf{e}^{a'}[n] - \mathbf{e}^a[n]$, por lo que:

$$\begin{aligned}
 E\{\mathbf{R}^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{R}[n]\} &= E\{(\mathbf{e}^{a'}[n])^T\mathbf{A}_{\Sigma}[n]\mathbf{e}^{a'}[n]\} - E\{(\mathbf{e}^a[n])^T\mathbf{A}_{\Sigma}[n]\mathbf{e}^a[n]\} \\
 &= E\{Tr(\mathbf{e}^{a'}[n](\mathbf{e}^{a'}[n])^T\mathbf{A}_{\Sigma}[n])\} - E\{Tr(\mathbf{e}^a[n](\mathbf{e}^a[n])^T\mathbf{A}_{\Sigma}[n])\}
 \end{aligned} \tag{A.39}$$

que considerando independencia estadística entre los datos de los vectores de error $\mathbf{e}^{a'}[n]$ y $\mathbf{e}^a[n]$ y los de la matriz $\mathbf{A}_{\Sigma}[n]$, se puede reescribir como:

$$\begin{aligned}
 E\{\mathbf{R}^T[n]\mathbf{A}_{\Sigma}[n]\mathbf{R}[n]\} &= Tr(E\{\mathbf{e}^{a'}[n](\mathbf{e}^{a'}[n])^T\}E\{\mathbf{A}_{\Sigma}[n]\}) \\
 &- Tr(E\{\mathbf{e}^a[n](\mathbf{e}^a[n])^T\}E\{\mathbf{A}_{\Sigma}[n]\}).
 \end{aligned} \tag{A.40}$$

Igual que se ha descrito en el capítulo 5, se puede demostrar que:

$$\begin{aligned}
 &E\{\mathbf{e}^{a'}[n](\mathbf{e}^{a'}[n])^T\} \approx \\
 &\left(\begin{array}{cccc}
 E\{|e_v^a[n]|^2\} & 0 & \dots & 0 \\
 0 & E\{|e_v^a[n-1]|^2\} & 0 & \dots \\
 \vdots & 0 & E\{|e_v^a[n-2]|^2\} & \vdots \\
 0 & & & E\{|e_v^a[n-N+1]|^2\}
 \end{array} \right)
 \end{aligned} \tag{A.41}$$

y

$$E\{\mathbf{e}_N^a[n](\mathbf{e}_N^a[n])^T\} \approx E\{\mathbf{e}_N^{a'}[n](\mathbf{e}_N^{a'}[n])^T\}D_{11} + \mu^2\sigma_r^2D_{12}. \quad (\text{A.42})$$

Siendo D_{11} y D_{12} las siguientes matrices diagonales:

$$D_{11} = \begin{pmatrix} 1 & 0 & \cdots & & 0 \\ 0 & (1-\mu)^2 & 0 & \cdots & \vdots \\ \vdots & 0 & (1-\mu)^4 & & \\ & & & \ddots & 0 \\ 0 & & & & (1-\mu)^{2(N-1)} \end{pmatrix} \quad (\text{A.43})$$

y

$$D_{12} = \begin{pmatrix} 0 & \cdots & & & 0 \\ 0 & 1 & 0 & \cdots & \vdots \\ \vdots & 0 & 1+(1-\mu)^2 & 0 & \\ & & & \ddots & 0 \\ 0 & \cdots & & 0 & 1+(1-\mu)^2+\dots+(1-\mu)^{2(N-2)} \end{pmatrix}. \quad (\text{A.44})$$

Por tanto, en la ecuación (A.40) aparece el valor buscado de la evolución temporal del transitorio en el instante n -ésimo, así como los $N-1$ anteriores, ya que el primer elemento de la traza depende de $E\{|e_v^a[n]|^2\}$ y el resto de los valores anteriores. Así, (A.39), se podría escribir como:

$$E\{\mathbf{R}^T[n]\mathbf{A}_\Sigma[n]\mathbf{R}[n]\} = \sum_{i=1}^N E\{|e_v^a[n-i+1]|^2\}\mathbf{M}(i,i) \quad (\text{A.45})$$

siendo \mathbf{M} la matriz $(\mathbf{I} - (D_{11} + \mu^2\sigma_r^2D_{12})E\{\mathbf{A}_\Sigma[n]\})$.

Finalmente, la evolución del transitorio ($E\{|e_v^a[n]|^2\}$) se puede escribir de forma recursiva según:

$$\begin{aligned} & E\{|e_v^a[n]|^2\} \\ &= \frac{E\{\|\tilde{\mathbf{w}}_L[0]\|_{\text{vec}^{-1}(\mathbf{f}[n])}^2\} + \mu^2\sigma_r^2g[n] + \mu^2\sum_{i=2}^N E\{|e_v^a[n-i+1]|^2\}\mathbf{M}(i,i)}{1 - \mu^2\mathbf{M}(1,1)} \\ &= E\{\|\tilde{\mathbf{w}}_L[0]\|_{\text{vec}^{-1}(\mathbf{f}[n])}^2\} + \mu^2\sigma_r^2g[n] + \mu^2\sum_{i=2}^N E\{|e_v^a[n-i+1]|^2\}\mathbf{M}(i,i) \end{aligned} \quad (\text{A.46})$$

con $\mathbf{f}[n] = \mathbf{F}\mathbf{f}[n - 1]$, $\mathbf{f}[0] = \sigma$, $g[n] = g[n - 1] + \gamma^T \mathbf{f}[n]$ y $g[0] = 0$.

Aunque, como se desprende de la expresión (A.46), el comportamiento del transitorio para la estructura de filtrado-x convencional es ligeramente diferente al obtenido para la de filtrado-x modificada, para los resultados de simulación, puesto que los caminos secundarios son muy cortos (no resulta viable realizar simulaciones con caminos secundarios mucho más largos debido a limitaciones computacionales) los resultados que se obtienen son bastante similares a los ya expuestos anteriormente.

La extrapolación de este estudio a un sistema genérico multicanal no es evidente, pero intuitivamente se puede deducir que el comportamiento de los sistemas multicanal seguirán la misma tendencia que el monocal en cuanto a la evolución del transitorio en cada uno de los sensores de error.

Apéndice B

Estrategias eficientes aplicadas a los algoritmos LS

B.1. Introducción

Aunque el algoritmo RLS presentado en el capítulo 2 reduce la complejidad computacional del algoritmo de mínimos cuadrados y consigue actualizar los pesos de una forma recursiva, todavía el coste computacional es bastante elevado en comparación con el LMS, por lo que se han desarrollado versiones más eficientes basadas en el algoritmo de mínimos cuadrados llamadas *fast RLS*. La más ampliamente usada de estas versiones *fast* es el algoritmo de *fast Kalman* [27] conocido también como *fast transversal filter* (FTF) puesto que usa filtros transversales para calcular los coeficientes de los filtros predictores que se usan en dichos algoritmos. Este tipo de estrategias comparten metodología con algunas de las estrategias eficientes empleadas en los algoritmos de proyección afín, por lo que resulta interesante, al menos, describirlas brevemente. En este anexo veremos algunas de estas estrategias *fast* para la resolución del algoritmo de mínimos cuadrados recursivos. En primer lugar se presentarán los algoritmos RLS de ventana deslizante a los que pueden aplicarse las estrategias eficientes que a continuación se describirán y que son extrapolables a los algoritmos de proyección afín.

B.2. Algoritmo *fast Kalman* o *fast transversal filter*

Antes de introducirnos en la base de los FTF, es conveniente definir las herramientas que nos van a servir para definir estos algoritmos *fast*, como

son los filtros de predicción lineal hacia adelante (*forward linear prediction* (FLP)) y hacia atrás (*backward linear prediction* (BLP)).

El problema del filtrado de predicción lineal hacia adelante, pretende encontrar los coeficientes del filtro que minimizan la señal de error expresada en (B.1) o lo que es lo mismo, trata de predecir con el mínimo error a partir de las últimas L muestras de una señal de referencia, el valor de la siguiente muestra:

$$e^f[n] = x[n] - \hat{x}[n] = x[n] - \mathbf{a}_L^T[n] \mathbf{x}_L[n-1], \quad (\text{B.1})$$

donde $\mathbf{a}_L[n]$ es un vector columna que contiene los L coeficientes del filtro predictor y $\mathbf{x}_L[n-1]$ es el *buffer* de datos de la señal de entrada conteniendo las muestras desde el instante $n-1$ hasta el instante $n-L$. Puesto que los filtros predictores son un caso particular de filtrado óptimo donde la señal deseada $d[n]$ es la siguiente muestra de la señal de referencia $x[n]$, podemos hallar los coeficientes del filtro predictor óptimo del mismo modo que se resolvió el problema de filtrado de mínimos cuadrados (en 2.5.2), obteniendo:

$$\mathbf{R}_L[n-1] \mathbf{a}_L[n] = \mathbf{r}_L^f[n], \quad (\text{B.2})$$

siendo $\mathbf{r}_L^f[n]$

$$\mathbf{r}_L^f[n] = \sum_{i=0}^n \mathbf{x}_L[i-1] x[i]. \quad (\text{B.3})$$

Del mismo modo, podemos calcular los coeficientes del filtro predictor hacia atrás (BLP) que son los que tratan de minimizar la expresión (B.4):

$$e^b[n] = x[n-L] - \hat{x}[n-L] = x[n-L] - \mathbf{b}_L^T[n] \mathbf{x}_L[n]. \quad (\text{B.4})$$

Los coeficientes de dicho filtro vendrían dados resolviendo la expresión:

$$\mathbf{R}_L[n] \mathbf{b}_L[n] = \mathbf{r}_L^b[n] \quad (\text{B.5})$$

con

$$\mathbf{r}_L^b[n] = \sum_{i=0}^n \mathbf{x}_L[i] x[i-L]. \quad (\text{B.6})$$

De forma análoga a lo ya visto anteriormente en 2.5.2 para obtener una forma recursiva de actualización de los coeficientes de los filtros por mínimos

cuadrados, podemos obtener para los filtros predictores las siguientes expresiones:

$$\mathbf{a}_L[n+1] = \mathbf{a}_L[n] + \mathbf{G}_L^*[n]e_{pri}^f[n+1] \quad (FLP) \quad (\text{B.7})$$

$$\mathbf{b}_L[n+1] = \mathbf{b}_L[n] + \mathbf{G}_L^*[n+1]e_{pri}^b[n+1] \quad (BLP) \quad (\text{B.8})$$

donde

$$e_{pri}^f[n+1] = x[n+1] - \mathbf{x}_L^T[n]\mathbf{a}_L[n] \quad (\text{B.9})$$

y

$$e_{pri}^b[n+1] = x[n-L+1] - \mathbf{x}_L^T[n+1]\mathbf{b}_L[n]. \quad (\text{B.10})$$

Por otro lado, haciendo uso de la definición del vector de ganancia de *Kalman* dado en (2.37), podemos definir una nueva relación suponiendo que usamos filtros adaptativos de $L+1$ coeficientes, dada por:

$$\mathbf{R}_{L+1}[n+1]\mathbf{G}_{L+1}^*[n+1] = \mathbf{x}_{L+1}[n+1] \quad (\text{B.11})$$

donde $\mathbf{x}_{L+1}[n]$ es un vector columna que contiene los $L+1$ últimos elementos de la variable $x(n)$, $\mathbf{G}_{L+1}^*[n]$ es el vector de ganancia de *Kalman* de tamaño $(L+1) \times 1$ y la matriz \mathbf{R}_{L+1} de tamaño $(L+1) \times (L+1)$ puede escribirse de una forma sencilla mediante matrices de bloque de las siguientes formas:

$$\mathbf{R}_{L+1}[n] = \begin{pmatrix} r_{oL}^f[n] & \mathbf{r}_L^f[n]^T \\ \mathbf{r}_L^f[n] & R_L[n-1] \end{pmatrix} \quad (\text{B.12})$$

$$\mathbf{R}_{L+1}[n] = \begin{pmatrix} R_L[n] & \mathbf{r}_L^b[n] \\ \mathbf{r}_L^b[n]^T & r_{oL}^b[n] \end{pmatrix} \quad (\text{B.13})$$

donde:

$$r_{oL}^f[n] = \sum_{i=0}^n x^2[i] \quad (\text{B.14})$$

y

$$r_{oL}^b[n] = \sum_{i=0}^n x^2[i-L]. \quad (\text{B.15})$$

Los valores $\mathbf{r}_L^f[n]$ y $\mathbf{r}_L^b[n]$ son los ya definidos para los filtros predictores en las ecuaciones (B.3) y (B.6) respectivamente.

La idea del filtrado *fast Kalman* se basa en computar el valor de la ganancia de *Kalman* $G_L^*[n+1]$ a partir de $G_{L+1}^*[n+1]$. Si volvemos a la ecuación (B.11), podemos escribir que:

$$\mathbf{G}_{L+1}^*[n+1] = \mathbf{R}_{L+1}^{-1}[n+1]\mathbf{x}_{L+1}[n+1] \quad (\text{B.16})$$

donde vuelve a aparecer una indeseada inversión matricial. Sin embargo, haciendo uso de sencillas manipulaciones algebraicas podemos reducir la complejidad de dicha inversión. Para ello, nos valemos de las siguientes relaciones matriciales de inversión de matrices de bloques:

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{-1} = \begin{pmatrix} F_{11}^{-1} & -F_{11}^{-1}A_{12}A_{22}^{-1} \\ -A_{22}^{-1}A_{21}F_{11}^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}F_{11}^{-1}A_{12}A_{22}^{-1} \end{pmatrix} \quad (\text{B.17})$$

siendo

$$F_{11} = A_{11} - A_{12}A_{22}^{-1}A_{21}. \quad (\text{B.18})$$

De esta forma, haciendo uso de (B.12) la matriz $\mathbf{R}_{L+1}^{-1}[n+1]$ puede escribirse como:

$$\begin{aligned} & \mathbf{R}_{L+1}^{-1}[n+1] \\ &= \begin{pmatrix} 1/\alpha^f[n+1] & -\mathbf{a}_L^T[n+1]/\alpha^f[n+1] \\ -\mathbf{a}_L[n+1]/\alpha^f[n+1] & \mathbf{R}_L^{-1}[n] + \mathbf{a}_L[n+1]\mathbf{a}_L^T[n+1]/\alpha^f[n+1] \end{pmatrix} \end{aligned} \quad (\text{B.19})$$

donde:

$$\alpha^f[n+1] = r_{oL}^f[n+1] - \mathbf{r}_L^f[n+1]^T \mathbf{R}_L^{-1}[n] \mathbf{r}_L^f[n+1] = r_{oL}^f[n+1] - \mathbf{a}_L^T[n+1] \mathbf{r}_L^f[n+1]. \quad (\text{B.20})$$

Además, la expresión (B.17) puede reescribirse también como

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{-1} = \begin{pmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}F_{22}^{-1}A_{21}A_{12}^{-1} & -A_{11}^{-1}A_{12}F_{22}^{-1} \\ -F_{22}^{-1}A_{21}A_{11}^{-1} & F_{22}^{-1} \end{pmatrix} \quad (\text{B.21})$$

siendo

$$F_{22} = A_{22} - A_{21}A_{11}^{-1}A_{12} \quad (\text{B.22})$$

por lo que, haciendo uso ahora de (B.13) la matriz $\mathbf{R}_{L+1}^{-1}[n+1]$ puede escribirse como:

$$\begin{aligned} & \mathbf{R}_{L+1}^{-1}[n+1] \\ &= \begin{pmatrix} \mathbf{R}_L^{-1}[n+1] + \mathbf{b}_L[n+1]\mathbf{b}_L^T[n+1]/\alpha^b[n+1] & -\mathbf{b}_L[n+1]/\alpha^b[n+1] \\ -\mathbf{b}_L^T[n+1]/\alpha^b[n+1] & 1/\alpha^b[n+1] \end{pmatrix} \end{aligned} \quad (\text{B.23})$$

donde:

$$\alpha^b[n+1] = r_{oL}^b[n+1] - \mathbf{r}_L^b[n+1]^T \mathbf{R}_L^{-1}[n+1] \mathbf{r}_L^b[n+1] = r_{oL}^b[n+1] - \mathbf{b}_L^T[n+1] \mathbf{r}_L^b[n+1]. \quad (\text{B.24})$$

Operando con la expresión (B.16), introduciendo las relaciones expresadas en (B.23) y teniendo en cuenta que:

$$\mathbf{x}_{L+1}[n] = \begin{pmatrix} \mathbf{x}_L[n] \\ x[n-L] \end{pmatrix} = \begin{pmatrix} x[n] \\ \mathbf{x}_L[n-1] \end{pmatrix}, \quad (\text{B.25})$$

podemos escribir para el vector $\mathbf{G}_{L+1}^*[n+1]$

$$\begin{pmatrix} \mathbf{R}_L^{-1}[n+1] \mathbf{x}_L[n+1] + \frac{\mathbf{b}_L[n+1] \mathbf{b}_L^T[n+1] \mathbf{x}_L[n+1]}{\alpha^b[n+1]} - \frac{\mathbf{b}_L[n+1] x[n-L+1]}{\alpha^b[n+1]} \\ -\mathbf{b}_L^T[n+1] \mathbf{x}_L[n+1] / \alpha^b[n+1] + x[n-L+1] / \alpha^b[n+1] \end{pmatrix} \quad (\text{B.26})$$

$$\mathbf{G}_{L+1}^*[n+1] = \begin{pmatrix} \mathbf{G}_L^*[n+1] \\ 0 \end{pmatrix} + \frac{1}{\alpha^b[n+1]} \begin{pmatrix} \mathbf{b}_L[n+1] (\mathbf{b}_L^T[n+1] \mathbf{x}_L[n+1] - x[n-L+1]) \\ -(\mathbf{b}_L^T[n+1] \mathbf{x}_L[n+1] - x[n-L+1]) \end{pmatrix}. \quad (\text{B.27})$$

Haciendo uso de la ecuación (B.4), podemos escribir:

$$\mathbf{G}_{L+1}^*[n+1] = \begin{pmatrix} \mathbf{G}_L^*[n+1] \\ 0 \end{pmatrix} - \frac{e^b[n+1]}{\alpha^b[n+1]} \begin{pmatrix} \mathbf{b}_L[n+1] \\ -1 \end{pmatrix}. \quad (\text{B.28})$$

Por otra parte, si al operar con la expresión (B.16) nos valemos de la ecuación (B.19), obtendríamos de forma análoga la siguiente expresión:

$$\mathbf{G}_{L+1}^*[n+1] = \begin{pmatrix} 0 \\ \mathbf{G}_L^*[n] \end{pmatrix} - \frac{e^f[n+1]}{\alpha^f[n+1]} \begin{pmatrix} -1 \\ \mathbf{a}_L[n+1] \end{pmatrix}. \quad (\text{B.29})$$

Ambas expresiones pueden interpretarse como las ecuaciones que pasan de la solución de un sistema de ecuaciones de orden L al de orden $L+1$ mediante el algoritmo de *Levinson* tal y como se describe en el anexo C. De hecho, la ecuación (B.28) es similar a (C.19), mientras que la ecuación (B.29) lo es a la ecuación (C.36). Por la analogía a dichas ecuaciones, podemos identificar

los coeficientes c_L como $\frac{e^b[n+1]}{\alpha^b[n+1]}$ en el primer caso y como $\frac{e^f[n+1]}{\alpha^f[n+1]}$ en el segundo.

Puesto que partimos de la idea de que estamos generando un método recursivo mediante el cual poder computar el valor de $\mathbf{G}_L^*[n+1]$ a partir de $\mathbf{G}_L^*[n]$ conocido de la iteración actual, podemos entonces computar el valor de $\mathbf{G}_{L+1}^*[n+1]$ haciendo uso de (B.29) y entonces obtener el valor deseado de $\mathbf{G}_L^*[n+1]$ despejando en (B.28).

De esta forma, casi tendríamos completado el algoritmo de actualización de los pesos del filtro de mínimos cuadrados actualizando de una forma recursiva el vector de ganancia de *Kalman* y usando la relación (2.38) para actualizar los pesos. Lo único que nos faltaría es calcular el llamado error a priori $e_{pri}[n+1]$, fácil de calcular mediante la expresión (2.35) si se conocen los valores de la señal deseada y de referencia en el instante actual y el valor de los coeficientes del filtro adaptativo en el instante anterior. También tendríamos que calcular los filtros predictores $\mathbf{b}_L[n+1]$ y $\mathbf{a}_L[n+1]$ que aparecen en las expresiones (B.28) y (B.29) así como los valores de $e^b[n+1]$, $\alpha^b[n+1]$, $e^f[n+1]$ y $\alpha^f[n+1]$. Estos cálculos pueden simplificarse si consideramos que la ecuación (B.28), puede escribirse como:

$$\mathbf{G}_{L+1}^*[n+1] = \begin{pmatrix} \overline{\mathbf{G}}_{L+1}^*[n+1] \\ g_{L+1}^*[n+1] \end{pmatrix} \quad (\text{B.30})$$

donde $\overline{\mathbf{G}}_{L+1}^*[n+1]$ representa un vector de tamaño L con los L primeros valores de $\mathbf{G}_{L+1}^*[n+1]$, y $g_{L+1}^*[n+1]$ es el escalar con el valor restante de dicho vector, que por simple comparación con (B.28) podemos decir que:

$$g_{L+1}^*[n+1] = \frac{e^b[n+1]}{\alpha^b[n+1]}, \quad (\text{B.31})$$

así que dicho cociente puede obtenerse una vez calculado el vector $\mathbf{G}_{L+1}^*[n+1]$ según la ecuación (B.29), por lo que solo faltaría por conocer en dicha ecuación el valor del filtro predictor $\mathbf{b}_L[n+1]$ que según se puede apreciar en la ecuación (B.8) necesita del conocimiento de $\mathbf{G}_L^*[n+1]$, por lo que la recursión no estaría cerrada. Sin embargo, introduciendo la ecuación (B.8) en (B.28), podemos expresar $\overline{\mathbf{G}}_{L+1}^*[n+1]$ como:

$$\overline{\mathbf{G}}_{L+1}^*[n+1] = \mathbf{G}_L^*[n+1] - g_{L+1}^*[n+1] \cdot (\mathbf{b}_L[n] + \mathbf{G}_L^*[n+1]e_{pri}^b[n+1]) \quad (\text{B.32})$$

y despejando el valor de $\mathbf{G}_L^*[n+1]$ obtenemos:

$$\mathbf{G}_L^*[n+1] = \frac{\overline{\mathbf{G}}_{L+1}^*[n+1] + g_{L+1}^*[n+1]\mathbf{b}_L[n]}{1 - g_{L+1}^*[n+1]e_{pri}^b[n+1]}. \quad (\text{B.33})$$

Lo único que faltaría por calcular para cerrar la recursión sería el valor de $\alpha^f[n+1]$ y que aparece en (B.20). Para ello, nos puede resultar útil darnos cuenta que el valor de $\alpha^f[n+1]$ se corresponde con el mínimo valor del error cuadrático total que se puede cometer en la estimación ($E^f[n+1]$). Este error mínimo puede calcularse de forma análoga a como se hizo en (2.29), obteniendo:

$$E^f[n] = \left(\sum_{i=0}^n x^2[i] \right) - \mathbf{a}_L^T[n]\mathbf{r}_L^f[n] \quad (\text{B.34})$$

y por simple comparación de (B.20) con (B.34) podemos decir que $\alpha^f[n]=E^f[n]$. Igualmente, podemos afirmar que $\alpha^b[n]=E^b[n]$, ya que $E^b[n]$ vendría dado por la siguiente expresión:

$$E^b[n] = \left(\sum_{i=0}^n x^2[i-L] \right) - \mathbf{b}_L^T[n]\mathbf{r}_L^b[n]. \quad (\text{B.35})$$

Según se calculó en (2.44) podríamos obtener que:

$$E^f[n+1] = E^f[n] + e_{pri}^f[n+1]e^f[n+1] \quad (\text{B.36})$$

$$E^b[n+1] = E^b[n] + e_{pri}^b[n+1]e^b[n+1] \quad (\text{B.37})$$

con lo que:

$$\alpha^f[n+1] = \alpha^f[n] + e_{pri}^f[n+1]e^f[n+1] \quad (\text{B.38})$$

$$\alpha^b[n+1] = \alpha^b[n] + e_{pri}^b[n+1]e^b[n+1] \quad (\text{B.39})$$

quedando ya cerrada la recursión para el algoritmo recursivo *fast Kalman*, que se describe a continuación:

En el instante n -ésimo disponemos de los valores de $\mathbf{G}_L^*[n]$, $\mathbf{a}_L[n]$, $\alpha^f[n]$, $\mathbf{b}_L[n]$ y $\mathbf{w}_L[n]$ y nos llega la siguiente nueva información: $d[n+1]$ y $x[n+1]$. Entonces:

- Actualizamos el valor del vector de ganancia de *Kalman*

- $e_{pri}^f[n+1] = x[n+1] - \mathbf{x}_L^T[n]\mathbf{a}_L[n]$
- $\mathbf{a}_L[n+1] = \mathbf{a}_L[n] + \mathbf{G}_L^*[n]e_{pri}^f[n+1]$
- $e^f[n+1] = x[n+1] - \mathbf{x}_L^T[n]\mathbf{a}_L[n+1]$
- $\alpha^f[n+1] = \alpha^f[n] + e_{pri}^f[n+1]e^f[n+1]$
- $\mathbf{G}_{L+1}^*[n+1] = \begin{pmatrix} 0 \\ \mathbf{G}_L^*[n] \end{pmatrix} - \frac{e^f[n+1]}{\alpha^f[n+1]} \begin{pmatrix} -1 \\ \mathbf{a}_L[n+1] \end{pmatrix}$
- Reorganizamos el vector de Ganancia de *Kalman*
 - $\mathbf{G}_{L+1}^*[n+1] = \begin{pmatrix} \overline{\mathbf{G}}_{L+1}^*[n+1] \\ g_{L+1}^*[n+1] \end{pmatrix}$
- $e_{pri}^b[n+1] = x[n-L+1] - \mathbf{b}_L^T[n]\mathbf{x}_L[n+1]$
- $\mathbf{G}_L^*[n+1] = \frac{\overline{\mathbf{G}}_{L+1}^*[n+1] + g_{L+1}^*[n+1]\mathbf{b}_L[n]}{1 - g_{L+1}^*[n+1]e_{pri}^b[n+1]}$
- $\mathbf{b}_L[n+1] = \mathbf{b}_L[n] + \mathbf{G}_L^*[n+1]e_{pri}^b[n+1]$
- Actualizamos los coeficientes del filtro FIR de mínimos cuadrados
 - $e_{pri}[n+1] = d[n+1] - \mathbf{x}_L^T[n+1]\mathbf{w}_L[n]$
 - $\mathbf{w}_L[n+1] = \mathbf{w}_L[n] + \mathbf{G}_L^*[n+1]e_{pri}[n+1]$.

El número de multiplicaciones a realizar por iteración para este algoritmo es $10L + 3$.

B.3. Algoritmo *fast Kalman* usando el error a posteriori

Una mejora computacional del algoritmo presentado, es el algoritmo *fast Kalman* usando el error a posteriori, que es el definido en (2.20) para la iteración n -ésima y que puede reescribirse para la iteración $n + 1$ como:

$$e[n+1] = d[n+1] - \mathbf{w}_L^T[n+1]\mathbf{x}_L[n+1]. \quad (\text{B.40})$$

Teniendo en cuenta dicha definición, podríamos obtener una expresión alternativa a (2.38) sin más que sustituir (2.30) y (2.31) en (2.32) con lo cual podríamos escribir que:

$$\mathbf{R}_L[n]\mathbf{w}_{opL}[n+1] - \mathbf{x}_L[n+1]e[n+1] = \mathbf{r}_L[n] \quad (\text{B.41})$$

y definiendo un vector de ganancia de *Kalman* (alternativo al que aparece en (2.37)), $\mathbf{G}_L[n+1]$ de tamaño $L \times 1$ como:

$$\mathbf{R}_L[n]\mathbf{G}_L[n+1] = \mathbf{x}_L[n+1] \quad (\text{B.42})$$

obtenemos, combinando ambas expresiones que:

$$\mathbf{R}_L[n](\mathbf{w}_{opL}[n+1] - \mathbf{G}_L[n+1]e[n+1]) = \mathbf{r}_L[n]. \quad (\text{B.43})$$

Finalmente, comparando (B.43) con (2.27), y despejando $\mathbf{w}_{opL}[n+1]$ se obtiene la siguiente ecuación de actualización recursiva:

$$\mathbf{w}_{opL}[n+1] = \mathbf{w}_{opL}[n] + \mathbf{G}_L[n+1]e[n+1]. \quad (\text{B.44})$$

Se puede demostrar (ver [27]) que la relación entre el vector de ganancia de *Kalman* ($\mathbf{G}_L^*[n]$) y el vector de ganancia de *Kalman* alternativo ($\mathbf{G}_L[n]$) es:

$$\mathbf{G}_L^*[n+1] = \mathbf{G}_L[n+1]/\alpha_L[n+1] \quad (\text{B.45})$$

siendo:

$$\alpha_L[n+1] = 1 + \mathbf{x}_L[n+1]\mathbf{G}_L^T[n+1]. \quad (\text{B.46})$$

Teniendo en cuenta (B.46), y por simple comparación entre (B.44) y (2.38), podemos establecer también las relaciones existentes entre el error a priori $e_{pri}[n+1]$ y el error a posteriori $e[n+1]$:

$$e[n+1] = e_{pri}[n+1]/\alpha_L[n+1]. \quad (\text{B.47})$$

Análogamente:

$$e^f[n+1] = e_{pri}^f[n+1]/\alpha_L[n+1] \quad (\text{B.48})$$

$$e^b[n+1] = e_{pri}^b[n+1]/\alpha_L[n+1]. \quad (\text{B.49})$$

Con las definiciones presentadas se puede realizar un desarrollo parecido al efectuado para el algoritmo *fast kalman*. Así, según la expresión (B.42) podemos decir que:

$$\mathbf{R}_L[n-1]\mathbf{G}_L[n] = \mathbf{x}_L[n] \quad (\text{B.50})$$

y

$$\mathbf{R}_{L+1}[n]\mathbf{G}_{L+1}[n+1] = \mathbf{x}_{L+1}[n+1]. \quad (\text{B.51})$$

Combinando (B.51) y (B.42) junto con (B.23) y (B.25) obtenemos:

$$\mathbf{G}_{L+1}[n+1] = \begin{pmatrix} \mathbf{G}_L[n+1] \\ 0 \end{pmatrix} - \frac{e_{pri}^b[n+1]}{\alpha^b[n]} \begin{pmatrix} \mathbf{b}_L[n] \\ -1 \end{pmatrix}. \quad (\text{B.52})$$

De forma análoga, combinando (B.19), (B.25), (B.50) y (B.51) se obtiene:

$$\mathbf{G}_{L+1}[n+1] = \begin{pmatrix} 0 \\ \mathbf{G}_L[n] \end{pmatrix} - \frac{e_{pri}^f[n+1]}{\alpha^f[n]} \begin{pmatrix} -1 \\ \mathbf{a}_L[n] \end{pmatrix}. \quad (\text{B.53})$$

De nuevo, identificando la ecuación (B.52) con (C.19), y la ecuación (B.53) con (C.36), se pueden identificar los coeficientes c_L como $\frac{e_{pri}^b[n+1]}{\alpha^b[n]}$ en el primer caso y $\frac{e_{pri}^f[n+1]}{\alpha^f[n]}$ en el segundo.

Igual que en el caso del filtrado *fast Kalman*, vamos a tratar de obtener una expresión recursiva para $\mathbf{G}_L[n]$ haciendo uso de $\mathbf{G}_{L+1}[n+1]$. Para ello hemos de tener en cuenta algunas relaciones:

Basándonos en (B.46) definimos $\alpha_{L+1}[n+1]$ como:

$$\alpha_{L+1}[n+1] = 1 + \mathbf{x}_{L+1}[n+1] \mathbf{G}_{L+1}[n+1] \quad (\text{B.54})$$

y usando las ecuaciones (B.25) y (B.53), podemos escribir que:

$$\begin{aligned} \alpha_{L+1}[n+1] = 1 + (x[n+1] \ \mathbf{x}_L^T[n]) & \begin{pmatrix} 0 \\ \mathbf{G}_L[n] \end{pmatrix} \\ & - (x[n+1] \ \mathbf{x}_L^T[n]) \frac{e_{pri}^f[n+1]}{\alpha^f[n]} \begin{pmatrix} -1 \\ \mathbf{a}_L[n] \end{pmatrix} \end{aligned} \quad (\text{B.55})$$

y por último, aplicando las expresiones definidas en (B.46) y (B.9) obtenemos:

$$\alpha_{L+1}[n+1] = \alpha_L[n] + \frac{e_{pri}^f[n+1]}{\alpha^f[n]} e_{pri}^f[n+1]. \quad (\text{B.56})$$

De momento, hemos obtenido una relación recursiva para calcular $\alpha_{L+1}[n+1]$ de $\alpha_L[n]$. Faltaría encontrar la relación entre $\alpha_{L+1}[n+1]$ y $\alpha_L[n+1]$. Pero volviendo a la ecuación (B.54) y mediante las ecuaciones (B.25) y (B.52),

podemos escribir que:

$$\begin{aligned} \alpha_{L+1}[n+1] = 1 + (\mathbf{x}_L^T[n+1] \ x[n-L+1]) \begin{pmatrix} \mathbf{G}_L[n+1] \\ 0 \end{pmatrix} \\ - (\mathbf{x}_L^T[n+1] \ x[n-L+1]) \frac{e_{pri}^b[n+1]}{\alpha^b[n]} \begin{pmatrix} \mathbf{b}_L[n] \\ -1 \end{pmatrix} \end{aligned} \quad (\text{B.57})$$

y teniendo en cuenta las ecuaciones (B.46) y (B.10), encontramos la relación buscada:

$$\alpha_{L+1}[n+1] = \alpha_L[n+1] + \frac{e_{pri}^b[n+1]}{\alpha^b[n]} e_{pri}^b[n+1] \quad (\text{B.58})$$

$$\alpha_L[n+1] = \alpha_{L+1}[n+1] - \frac{e_{pri}^b[n+1]}{\alpha^b[n]} e_{pri}^b[n+1]. \quad (\text{B.59})$$

Faltaría, pues, obtener una relación recursiva para el vector de ganancia de *Kalman* alternativo $\mathbf{G}_L[n]$, pero operando de forma análoga a como se hizo en el algoritmo *fast Kalman*, fraccionando el vector $\mathbf{G}_{L+1}[n+1]$, tendremos:

$$\mathbf{G}_{L+1}[n+1] = \begin{pmatrix} \overline{\mathbf{G}}_{L+1}[n+1] \\ g_{L+1}[n+1] \end{pmatrix} \quad (\text{B.60})$$

donde podemos comprobar que:

$$g_{L+1}[n+1] = \frac{e_{pri}^b[n+1]}{\alpha^b[n]} \quad (\text{B.61})$$

y operando con la expresión (B.52) obtenemos la recursión buscada:

$$G_L[n+1] = \overline{\mathbf{G}}_{L+1}[n+1] - g_{L+1}[n+1] \mathbf{b}_L[n]. \quad (\text{B.62})$$

Con todas estas expresiones, el algoritmo de mínimos cuadrados *fast Kalman* usando el error a posteriori quedaría como sigue:

En el instante n -ésimo disponemos de los valores de $\mathbf{G}_L[n]$, $\alpha^f[n]$, $\alpha^b[n]$, $\alpha_L[n]$, $\mathbf{a}_L[n]$, $\mathbf{b}_L[n]$ y $\mathbf{w}_L[n]$ y nos llega la siguiente nueva información: $d[n+1]$ y $x[n+1]$. Entonces:

- Actualizamos el valor del vector de ganancia de *Kalman* alternativo
 - $e_{pri}^f[n+1] = x[n+1] - \mathbf{x}_L^T[n] \mathbf{a}_L[n]$
 - $e^f[n+1] = e_{pri}^f[n+1] / \alpha_L[n]$
 - $\mathbf{a}_L[n+1] = \mathbf{a}_L[n] + \mathbf{G}_L[n] e^f[n+1]$

- $\alpha^f[n+1] = \alpha^f[n] + e_{pri}^f[n+1]e^f[n+1]$
- $\mathbf{G}_{L+1}[n+1] = \begin{pmatrix} 0 \\ \mathbf{G}_L[n] \end{pmatrix} - \frac{e_{pri}^f[n+1]}{\alpha^f[n+1]} \begin{pmatrix} -1 \\ \mathbf{a}_L[n] \end{pmatrix}$
- Reorganizamos el vector de ganancia de *Kalman* alternativo
 - $\mathbf{G}_{L+1}[n+1] = \begin{pmatrix} \bar{\mathbf{G}}_{L+1}[n+1] \\ g_{L+1}[n+1] \end{pmatrix}$
- $e_{pri}^b[n+1] = g_{L+1}[n+1]\alpha^b[n]$
- $\mathbf{G}_L[n+1] = \bar{\mathbf{G}}_{L+1}[n+1] - g_{L+1}[n+1]\mathbf{b}_L[n]$
- $\alpha_{L+1}[n+1] = \alpha[n]_L - \frac{e_{pri}^f[n+1]}{\alpha^f[n]} e_{pri}^f[n+1]$
- $\alpha_L[n+1] = \alpha_{L+1}[n+1] + g_{L+1}[n+1]e_{pri}^b[n+1]$
- $e^b[n+1] = e_{pri}^b[n+1]/\alpha[n+1]$
- $\alpha^b[n+1] = \alpha^b[n] + e_{pri}^b[n+1]e^b[n+1]$
- $\mathbf{b}[n+1] = \mathbf{b}[n] + \mathbf{G}_L[n+1]e^b[n+1]$
- Actualizamos los coeficientes del filtro FIR de mínimos cuadrados
 - $e_{pri}[n+1] = d[n+1] - \mathbf{x}_L^T[n+1]\mathbf{w}_L[n]$
 - $e[n+1] = e_{pri}[n+1]/\alpha[n+1]$
 - $\mathbf{w}_L[n+1] = \mathbf{w}_L[n] + \mathbf{G}_L[n+1]e[n+1]$.

En este caso, el número de multiplicaciones por iteración a realizar se reduce hasta $5L + 8$.

B.4. Algoritmos de ventana deslizante

Al presentar el problema de mínimos cuadrados en el capítulo 2 comentamos que la función de coste incluía toda la historia de señal de error a minimizar y la única hipótesis que se hacía de las señales era la hipótesis de preeventanado (para instantes de tiempo anteriores a cero ($n < 0$), las señales tienen valor cero). Dicha función de coste era

$$\xi[n] = \sum_{i=0}^n e^2[i]. \quad (\text{B.63})$$

Para señales no estacionarias, esta función de coste no era óptima puesto que la historia pasada no nos aporta la misma información que la más reciente sobre las señales a controlar, por lo que nos puede interesar ponderar de alguna forma los valores más recientes de la señal frente a los más antiguos. Una forma de hacer esto es variando la función de coste presentada anteriormente en la ecuación (B.63) introduciendo una atenuación exponencial con el índice temporal (función de coste ya presentada en el algoritmo RLS), que es lo que se conoce como enventanado exponencial

$$\xi[n] = \sum_{i=0}^n \lambda^{n-i} e^2[i]. \quad (\text{B.64})$$

Sin embargo, otra técnica muy usada es lo que se conoce como enventanado deslizante (que también puede ser enventanado exponencial deslizante) y que consiste en considerar para la función de coste solo un conjunto de las muestras más reciente de las señales. Matemáticamente hablando, esto se puede describir mediante la siguiente función de coste:

$$\xi'[n] = \sum_{i=0}^L \lambda^{n-i} e^2[n-i]. \quad (\text{B.65})$$

Si $\lambda=1$ tendríamos el enventanado deslizante y en cualquier otro caso, el enventanado exponencial deslizante. A continuación vamos a estudiar algunos aspectos del algoritmo de mínimos cuadrados con enventanado deslizante puesto que es de gran utilidad para aplicarlos a los algoritmos de proyección afín eficientes (capítulo 3).

B.5. Solución del problema mediante mínimos cuadrados (LS) con enventanado deslizante

El nuevo problema al que nos enfrentamos estaría basado otra vez en el sistema adaptativo presentado en la figura 2.4 (capítulo 2), tratando de encontrar los N coeficientes del filtro adaptativo que minimizan la función de coste reflejada en (B.65). Habitualmente, este problema se conoce como de mínimos cuadrados con enventanado deslizante de orden L , siendo L el tamaño de la ventana.

Si llamamos $y[n]$ a la salida de nuestro filtro $\mathbf{w}'_N(n)$ de N coeficientes, tendremos:

$$y[n] = \sum_{i=0}^{N-1} w'_i[n] x[n-i] = \mathbf{w}'_N{}^T[n] \mathbf{x}_N[n] \quad (\text{B.66})$$

donde

$$\mathbf{w}'_N[n] = (w'_0[n], w'_1[n], \dots, w'_{N-1}[n])^T \quad (\text{B.67})$$

y

$$\mathbf{x}_N[n] = (x[n], x[n-1], \dots, x[n-N+1])^T. \quad (\text{B.68})$$

Pretendemos encontrar los coeficientes del filtro adaptativo que minimizan mediante mínimos cuadrados la suma del error cuadrático en las últimas L iteraciones, o lo que es lo mismo:

$$\xi'[n] = \sum_{i=0}^L e^2[n-i] \quad (\text{B.69})$$

siendo:

$$e[i] = d[i] - \mathbf{w}'_N{}^T[n] \mathbf{x}_N[i]. \quad (\text{B.70})$$

Como se puede apreciar en la ecuación (B.70) vamos a suponer que los coeficientes del filtro permanecen invariantes en cada bloque de L muestras (aunque en la práctica esto no ocurrirá, igual que pasaba para el algoritmo LS, se asume con frecuencia puesto que simplifica mucho las operaciones a realizar y es justificable cuando se da una variación lenta de los coeficientes).

Partiendo de la expresión (B.69), obtenemos:

$$\begin{aligned} \xi'[n] &= \left(\sum_{i=0}^L d^2[n-i] \right) + \mathbf{w}'_N{}^T[n] \left(\sum_{i=0}^L \mathbf{x}_N[n-i] \mathbf{x}_N^T[n-i] \right) \mathbf{w}'_N[n] \\ &\quad - 2\mathbf{w}'_N{}^T[n] \left(\sum_{i=0}^L d[n-i] \mathbf{x}_N[n-i] \right). \end{aligned} \quad (\text{B.71})$$

Para simplificar la notación, llamamos:

$$\begin{aligned} \mathbf{R}_N[n] &= \sum_{i=0}^L \mathbf{x}_N[n-i] \mathbf{x}_N^T[n-i] = \mathbf{A}_N[n] \mathbf{A}_N^T[n] \\ &= \mathbf{R}_N[n-1] + \mathbf{x}_N[n] \mathbf{x}_N^T[n] - \mathbf{x}_N[n-L] \mathbf{x}_N^T[n-L] \end{aligned} \quad (\text{B.72})$$

y

$$\begin{aligned} \mathbf{r}'_N[n] &= \sum_{i=0}^L \mathbf{x}_N[n-i] d[n-i] = \mathbf{A}_N[n] \mathbf{d}_L[n] \\ &= \mathbf{r}'_N[n-1] + \mathbf{x}_N[n] d[n] - \mathbf{x}_N[n-L] d[n-L] \end{aligned} \quad (\text{B.73})$$

donde

$$\mathbf{A}_N[n] = \begin{pmatrix} x[n] & x[n-1] & \dots & x[n-L+1] \\ x[n-1] & x[n-2] & \dots & x[n-L+2] \\ \vdots & & \ddots & \vdots \\ x[n-N+1] & x[n-N] & \dots & x[n-N-L+2] \end{pmatrix} \quad (\text{B.74})$$

y $\mathbf{d}_L[n] = (d[n], d[n-1], \dots, d[n-L+1])^T$ entonces, podemos reescribir la ecuación (B.71) como:

$$\xi'[n] = \left(\sum_{i=0}^L d^2[n-i] \right) + \mathbf{w}'_N{}^T[n] \mathbf{R}_N[n] \mathbf{w}'_N[n] - 2\mathbf{w}'_N{}^T[n] \mathbf{r}'_N[n]. \quad (\text{B.75})$$

Derivando $\xi'[n]$ con respecto $\mathbf{w}'_N[n]$ obtenemos:

$$\frac{\partial(\xi'[n])}{\partial \mathbf{w}'_N[n]} = 2\mathbf{R}_N[n] \mathbf{w}'_N[n] - 2\mathbf{r}'_N[n] \quad (\text{B.76})$$

e igualando a cero, obtenemos que los valores de los coeficientes $\mathbf{w}'_N[n]$ que minimizan la suma parcial de los últimos L elementos del error cuadrático satisface la siguiente ecuación:

$$\mathbf{R}_N[n] \mathbf{w}'_{opN}[n] = \mathbf{r}'_N[n] \quad (\text{B.77})$$

obteniendo para el valor mínimo de dicho error:

$$\begin{aligned} \xi'[n]_{min} &= \left(\sum_{i=0}^N d^2[n-i] \right) + \mathbf{w}'_{opN}{}^T[n] \mathbf{R}_N[n] \mathbf{w}'_{opN}[n] \\ &\quad - 2\mathbf{w}'_{opN}{}^T[n] \mathbf{R}_N[n] \mathbf{w}'_{opN}[n] \\ &= \left(\sum_{i=0}^L d^2[n-i] \right) - \mathbf{w}'_{opN}{}^T[n] \mathbf{R}_N[n] \mathbf{w}'_{opN}[n] \\ &= \left(\sum_{i=0}^L d^2[n-i] \right) - \mathbf{w}'_{opN}{}^T[n] \mathbf{r}'_N[n]. \end{aligned} \quad (\text{B.78})$$

B.6. Mínimos cuadrados recursivos con ventana deslizante

Para el cálculo de los coeficientes óptimos que resuelven el problema que estamos tratando, según la ecuación (B.77) tendremos que calcular una inversa matricial. Esta matriz a invertir es similar a la encontrada en el caso del

problema clásico de mínimos cuadrados. Al igual que entonces, basándonos en el lema de inversión matricial podemos encontrar una relación recursiva para calcular dicha matriz inversa. Antes que nada, hay que darse cuenta de que se cumplen las siguientes relaciones:

$$\begin{aligned}\mathbf{R}_N[n] &= \mathbf{R}_N[n-1] + \mathbf{x}_N[n]\mathbf{x}_N^T[n] - \mathbf{x}_N[n-L]\mathbf{x}_N^T[n-L] \\ &= \mathbf{Q}_N[n] - \mathbf{x}_N[n-L]\mathbf{x}_N^T[n-L]\end{aligned}\quad (\text{B.79})$$

siendo

$$\mathbf{Q}_N[n] = \mathbf{R}_N[n-1] + \mathbf{x}_N[n]\mathbf{x}_N^T[n]. \quad (\text{B.80})$$

Si realizamos las siguientes identificaciones en la ecuación (B.79)

1. $A = \mathbf{R}_N[n]$
2. $B^{-1} = \mathbf{Q}_N[n]$
3. $C = \mathbf{x}_N[n]$
4. $D = -1$

podemos aplicar el lema de inversión matricial (como se define en 2.5.2 para calcular $\mathbf{R}_N^{-1}[n]$, quedando:

$$\begin{aligned}\mathbf{R}_N^{-1}[n] &= \mathbf{Q}_N^{-1}[n] \\ &\quad - \mathbf{Q}_N^{-1}[n]\mathbf{x}_N[n-L](-1 + \mathbf{x}_N^T[n-L]\mathbf{Q}_N^{-1}[n]\mathbf{x}_N[n-L])^{-1}\mathbf{x}_N^T[n-L]\mathbf{Q}_N^{-1}[n]\end{aligned}\quad (\text{B.81})$$

que llamando $\beta[n] = \mathbf{Q}_N^{-1}[n]\mathbf{x}_N[n-L]$, puede reescribirse de forma más sencilla como:

$$\mathbf{R}_N^{-1}[n] = \mathbf{Q}_N^{-1}[n] + \beta[n](1 - \mathbf{x}_N^T[n-L]\beta[n])^{-1}\beta^T[n]. \quad (\text{B.82})$$

Para calcular $\mathbf{Q}_N^{-1}[n]$, volvemos a aplicar el lema de inversión matricial, pero con las siguientes identificaciones en la ecuación (B.80):

1. $A = \mathbf{Q}_N[n]$
2. $B^{-1} = \mathbf{R}_N[n-1]$
3. $C = \mathbf{x}_N[n]$
4. $D = 1$

con lo que tendríamos

$$\mathbf{Q}_N^{-1}[n] = \mathbf{R}_N^{-1}[n-1] - \mathbf{R}_N^{-1}[n-1] \mathbf{x}_N[n] (1 + \mathbf{x}_N^T[n] \mathbf{R}_N^{-1}[n-1] \mathbf{x}_N[n])^{-1} \mathbf{x}_N^T[n] \mathbf{R}_N^{-1}[n] \quad (\text{B.83})$$

que llamando $\alpha[n] = \mathbf{R}_N^{-1}[n-1] \mathbf{x}_N[n]$, puede reescribirse de forma más sencilla como:

$$\mathbf{Q}_N^{-1}[n] = \mathbf{R}_N^{-1}[n-1] - \alpha[n] (1 + \mathbf{x}_N^T[n] \alpha[n])^{-1} \alpha^T[n] \quad (\text{B.84})$$

por lo que un algoritmo recursivo para el cálculo de la inversión matricial (queremos calcular $\mathbf{R}_N^{-1}[n]$ a partir de $\mathbf{R}_N^{-1}[n-1]$), quedaría de la siguiente forma:

1. $\alpha[n] = \mathbf{R}_N^{-1}[n-1] \mathbf{x}_N[n]$
2. $\mathbf{Q}_N^{-1}[n] = \mathbf{R}_N^{-1}[n-1] - \alpha[n] \alpha^T[n] / (1 + \mathbf{x}_N^T[n] \alpha[n])$
3. $\beta[n] = \mathbf{Q}_N^{-1}[n] \mathbf{x}_N[n-L]$,
4. $\mathbf{R}_N^{-1}[n] = \mathbf{Q}_N^{-1}[n] + \beta[n] \beta^T[n] / (1 - \mathbf{x}_N^T[n-L] \beta[n])^{-1}$.

El número de multiplicaciones necesarias para calcular esta inversión matricial es de $2(N^2 + N + 2)$. Una vez obtenida la inversión matricial, la actualización de los coeficientes es sencilla sin más que aplicar la ecuación (B.77).

B.7. Fast RLS con eventanado deslizante

Siguiendo el procedimiento realizado cuando se calculó un método recursivo eficiente para el cálculo de los coeficientes óptimos mediante LS, vamos a intentar conseguir un algoritmo similar para el caso de eventanado deslizante.

Para ello, partimos de las siguientes relaciones:

$$\mathbf{R}_N[n+1] = \mathbf{R}_N[n] + \mathbf{x}_N[n+1] \mathbf{x}_N^T[n+1] - \mathbf{x}_N[n-L+1] \mathbf{x}_N^T[n-L+1] \quad (\text{B.85})$$

y

$$\mathbf{r}'_N[n+1] = \mathbf{r}'_N[n] + \mathbf{x}_N[n+1] d[n+1] - \mathbf{x}_N[n-L+1] d[n-L+1]. \quad (\text{B.86})$$

Introduciendo los valores de $\mathbf{R}_N[n]$ y $\mathbf{r}_N[n]$ que aparecen en las ecuaciones (B.85) y (B.86) respectivamente, en la ecuación (B.77), tendríamos:

$$\begin{aligned} (\mathbf{R}_N[n+1] - \mathbf{x}_N[n+1] \mathbf{x}_N^T[n+1] + \mathbf{x}_N[n-L+1] \mathbf{x}_N^T[n-L+1]) \mathbf{w}'_N[n] = \\ = \mathbf{r}'_N[n+1] - \mathbf{x}_N[n+1] d[n+1] + \mathbf{x}_N[n-L+1] d[n-L+1] \end{aligned} \quad (\text{B.87})$$

$$\begin{aligned} & \mathbf{R}_N[n+1]\mathbf{w}'_N[n] + \mathbf{x}_N[n+1](d[n+1] - \mathbf{x}_N^T[n+1]\mathbf{w}'_N[n]) - \\ & - \mathbf{x}_N[n-L+1]\{d[n-L+1] - \mathbf{x}_N^T[n-L+1]\mathbf{w}'_N[n]\} = \mathbf{r}'_N[n+1]. \end{aligned} \quad (\text{B.88})$$

Usando la definición del error a priori introducida anteriormente en la ecuación (2.35):

$$\mathbf{R}_N[n+1]\mathbf{w}'_N[n] + \mathbf{x}_N[n+1]e_{pri}[n+1] - \mathbf{x}_N[n-L+1]e_{pri}[n-L+1] = \mathbf{r}'_N[n+1]. \quad (\text{B.89})$$

Al igual que hicimos con los algoritmos LS, definimos las constantes de ganancia de *Kalman* como los vectores que cumplen las siguientes relaciones:

$$\mathbf{R}_N[n+1]\mathbf{G}'_N^*[n+1] = \mathbf{x}_N[n+1] \quad (\text{B.90})$$

$$\mathbf{R}_N[n+1]\mathbf{G}'_N^*[n-L+1] = \mathbf{x}_N[n-L+1] \quad (\text{B.91})$$

por lo que (B.89), puede reescribirse como:

$$\begin{aligned} & \mathbf{R}_N[n+1]\mathbf{w}'_N[n] + \mathbf{R}_N[n+1]\mathbf{G}'_N^*[n+1]e_{pri}[n+1] \\ & - \mathbf{R}_N[n+1]\mathbf{G}'_N^*[n-L+1]e_{pri}[n-L+1] = \mathbf{r}'_N[n+1]. \end{aligned} \quad (\text{B.92})$$

Considerando que los pesos óptimos en la iteración $n+1$ vendrían dados por la expresión:

$$\mathbf{R}_N[n+1]\mathbf{w}'_N[n+1] = \mathbf{r}'_N[n+1], \quad (\text{B.93})$$

y por comparación con (B.92), podemos encontrar una ecuación recursiva para la actualización de los pesos:

$$\mathbf{w}'_N[n+1] = \mathbf{w}'_N[n] + \mathbf{G}'_N^*[n+1]e_{pri}[n+1] - \mathbf{G}'_N^*[n-L+1]e_{pri}[n-L+1]. \quad (\text{B.94})$$

También nos resultará útil encontrar una expresión recursiva para calcular el valor del error mínimo que se comete al usar enventanado deslizante. Usando la expresión calculada en (B.78), tenemos que:

$$\begin{aligned} \xi'[n+1]_{min} &= \mathbf{E}'[n+1] = \left(\sum_{i=0}^L d^2[n-i+1] \right) - \mathbf{w}'_N{}^T[n+1]\mathbf{r}'_N[n+1] = \\ &= \left(\sum_{i=0}^L d^2[n-i] \right) + d^2[n+1] - d^2[n-L] - \mathbf{w}'_N{}^T[n]\mathbf{r}'_N[n+1] - \\ & - \mathbf{r}'_N{}^T[n+1]\mathbf{G}'_N^*[n+1]e_{pri}[n+1] + \mathbf{r}'_N{}^T[n+1]\mathbf{G}'_N^*[n-L+1]e_{pri}[n-L+1]. \end{aligned} \quad (\text{B.95})$$

Introduciendo el valor del error a priori y operando de igual forma a como se hizo en el capítulo 2 para el caso de mínimos cuadrados sin inventanar, obtenemos:

$$\mathbf{E}'[n+1] = \mathbf{E}'[n] + e_{pri}[n+1]e[n+1] - e_{pri}[n-L+1]e[n-L+1]. \quad (\text{B.96})$$

De igual forma, se pueden calcular las ecuaciones que proporcionan los filtros predictores de N coeficientes hacia adelante y hacia atrás con inventanado deslizante de orden L , resultando que para los filtros predictores hacia adelante, el error sería el definido en (B.1) y la ecuación que proporciona los coeficientes de los filtros óptimos:

$$\mathbf{R}_N[n-1]\mathbf{a}'_N[n] = \mathbf{r}'_N^f[n] \quad (\text{B.97})$$

siendo $\mathbf{r}'_N^f[n]$

$$\mathbf{r}'_N^f[n] = \sum_{i=0}^L \mathbf{x}_N[n-i-1]x[n-i]. \quad (\text{B.98})$$

El mínimo error cometido vendría dado por la expresión:

$$E'_{min}[n] = \sum_{i=0}^L x^2[n-i] - \mathbf{a}'_N{}^T[n]\mathbf{r}'_N^f[n], \quad (\text{B.99})$$

obteniendo para las expresiones recursivas de este error y de los coeficientes:

$$E'_{min}[n+1] = E'_{min}[n] + e_{pri}^f[n+1]e^f[n+1] - e_{pri}^f[n-L+1]e^f[n-L+1]. \quad (\text{B.100})$$

$$\mathbf{a}'_N[n+1] = \mathbf{a}'_N[n] + \mathbf{G}'_N{}^*[n+1]e_{pri}^f[n+1] - \mathbf{G}'_N{}^*[n-L+1]e_{pri}^f[n-L+1] \quad (\text{B.101})$$

Mientras que para los filtros predictores hacia atrás, el error sería el definido en la ecuación (B.4) y los coeficientes del filtro se obtendrían resolviendo la expresión:

$$\mathbf{R}_N[n]\mathbf{b}'_N[n] = \mathbf{r}'_N^b[n] \quad (\text{B.102})$$

con

$$\mathbf{r}'_N^b[n] = \sum_{i=0}^L \mathbf{x}_N[n-i]x[n-i-L] \quad (\text{B.103})$$

pudiendo calcular recursivamente los coeficientes óptimos, mediante:

$$\mathbf{b}'_N[n+1] = \mathbf{b}'_N[n] + \mathbf{G}'_N{}^*[n+1]e_{pri}^b[n+1] - \mathbf{G}'_N{}^*[n-L+1]e_{pri}^b[n-L+1]. \quad (\text{B.104})$$

Y para el mínimo error de predicción:

$$E_{min}^b[n] = \sum_{i=0}^L x^2[n-i-L] - \mathbf{b}'_N{}^T[n] \mathbf{r}'_N{}^b[n] \quad (\text{B.105})$$

pudiéndose obtener para su expresión recursiva:

$$E_{min}^b[n+1] = E_{min}^f[n] + e_{pri}^b[n+1]e^b[n+1] - e_{pri}^b[n-L+1]e^b[n-L+1]. \quad (\text{B.106})$$

Siguiendo la metodología expuesta cuando se habló de las técnicas eficientes para resolver el problema de mínimos cuadrados (en los puntos anteriores de este anexo), podemos exponer un método eficiente para la inversión matricial que aparece al calcular los coeficientes óptimos.

Así, si llamamos $\mathbf{R}_N[n]$ a la matriz a invertir:

$$\mathbf{R}_N[n] = \mathbf{A}[n]\mathbf{A}[n]^T \quad (\text{B.107})$$

podremos darnos cuenta que se cumplen las siguientes relaciones de forma parecida a lo que sucedía en los algoritmos FTF descritos anteriormente:

$$\mathbf{R}_N[n] = \begin{pmatrix} r'_{oL}[n] & \mathbf{r}'_{N-1}[n]^T \\ \mathbf{r}'_{N-1}[n] & \mathbf{R}_{N-1}[n-1] \end{pmatrix} = \begin{pmatrix} r'_{oL}[n] & \mathbf{r}'_{N-1}[n]^T \\ \mathbf{r}'_{N-1}[n] & \tilde{\mathbf{R}}_N[n] \end{pmatrix} \quad (\text{B.108})$$

$$\mathbf{R}_N[n] = \begin{pmatrix} \mathbf{R}_{N-1}[n] & \mathbf{r}'_{N-1}[n] \\ \mathbf{r}'_{N-1}[n]^T & r'_{o(N-1)}[n] \end{pmatrix}. \quad (\text{B.109})$$

Estas matrices $\mathbf{R}_N[n]$ coinciden con las matrices a invertir en los algoritmos de proyección afín de orden N y L coeficientes en los filtros adaptativos. Así $\mathbf{R}_{N-1}[n]$ sería la matriz a invertir en el algoritmo de proyección afín en caso de usar un orden de proyección de $N-1$, y que coincide con los $(N-1) \times (N-1)$ elementos inferiores-izquierdos de la matriz $\mathbf{R}_N[n+1]$. O sea que $\mathbf{R}_{N-1}[n-1] = \tilde{\mathbf{R}}_N[n]$. Los valores $\mathbf{r}'_{N-1}[n]$ y $\mathbf{r}'_{N-1}[n]$ se calculan de forma similar a los ya definidos para los filtros predictores con enventanado deslizante en las ecuaciones (B.98) y (B.103) respectivamente, siendo:

$$\begin{aligned} \mathbf{r}'_{N-1}[n] &= \sum_{i=0}^L \mathbf{x}_{N-1}[n-i-1]x[n-i] \\ &= \mathbf{r}'_{N-1}[n-1] + \mathbf{x}_{N-1}[n-1]x[n] - \mathbf{x}_{N-1}[n-L-1]x[n-L] \end{aligned} \quad (\text{B.110})$$

$$\begin{aligned}\mathbf{r}'^b_{N-1}[n] &= \sum_{i=0}^L \mathbf{x}_{N-1}[n-i]x[n-i-L] \\ &= \mathbf{r}'^b_{N-1}[n-1] + \mathbf{x}_{N-1}[n]x[n-L] - \mathbf{x}_{N-1}[n-L]x[n-2L]\end{aligned}\quad (\text{B.111})$$

y

$$r'^f_{oL}[n] = \mathbf{x}_L^T[n]\mathbf{x}_L[n] = \sum_{i=0}^L x^2[n-i] = r'^f_{oL}[n-1] + x^2[n] - x^2[n-L] \quad (\text{B.112})$$

$$r'^b_{oL}[n] = \mathbf{x}_L^T[n-L]\mathbf{x}_L[n-L] = \sum_{i=0}^L x^2[n-i-L] = r'^b_{oL}[n-1] + x^2[n-L] - x^2[n-2L]. \quad (\text{B.113})$$

Calculando la inversa de la matriz (B.108) haciendo uso de (B.17) según se describió en las secciones anteriores, tendríamos que:

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} 1/\alpha'^f[n] & -\mathbf{a}'^T_{N-1}[n]/\alpha'^f[n] \\ -\mathbf{a}'_{N-1}[n]/\alpha'^f[n] & \tilde{\mathbf{R}}_N^{-1}[n] + \mathbf{a}'_{N-1}[n]\mathbf{a}'^T_{N-1}[n]/\alpha'^f[n] \end{pmatrix} \quad (\text{B.114})$$

donde:

$$\alpha'^f[n] = r'^f_{oL}[n] - \mathbf{r}'^f_{N-1}[n]^T \mathbf{R}_N^{-1}[n-1] \mathbf{r}'^f_{N-1}[n] = r'^f_{oL}[n] - \mathbf{a}'^T_{N-1}[n] \mathbf{r}'^f_{N-1}[n]. \quad (\text{B.115})$$

La matriz (B.114), puede ponerse de forma más compacta si definimos unos vectores extendidos de los coeficientes adaptativos que incluyen la unidad. Así:

$$e^f[n] = x[n] - \hat{x}[n] = x[n] - \mathbf{a}'^T_{N-1}[n]\mathbf{x}_{N-1}[n-1] = \hat{\mathbf{a}}^T_{N-1}[n]\mathbf{x}_N[n] \quad (\text{B.116})$$

siendo:

$$\mathbf{x}_N[n] = (x[n], x[n-1], \dots, x[n-N+1])^T \quad (\text{B.117})$$

y

$$\hat{\mathbf{a}}^T_{N-1}[n] = (1, -\mathbf{a}'^T_{N-1}[n])^T \quad (\text{B.118})$$

por lo que ahora, podemos poner la ecuación (B.114) como:

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \tilde{\mathbf{R}}_N^{-1}[n] \end{pmatrix} + \hat{\mathbf{a}}^T_{N-1}[n]\hat{\mathbf{a}}^T_{N-1}[n]/\alpha'^f[n] \quad (\text{B.119})$$

siendo $\mathbf{0}_N$ un vector columna de N ceros.

De igual forma, si trabajamos con la definición de $\mathbf{R}_N[n]$ introducida en (B.109), haciendo uso de (B.21) obtenemos que,

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} \mathbf{R}_{N-1}^{-1}[n] + \mathbf{b}'_{N-1}[n]\mathbf{b}'_{N-1}^T[n]/\alpha'^b[n] & -\mathbf{b}'_{N-1}[n]/\alpha'^b[n] \\ -\mathbf{b}'_{N-1}^T[n]/\alpha'^b[n] & 1/\alpha'^b[n] \end{pmatrix} \quad (\text{B.120})$$

donde:

$$\alpha'^b[n] = r'_{oL}[n] - \mathbf{r}'_{N-1}[n]^T \mathbf{R}_{N-1}^{-1}[n] \mathbf{r}'_{N-1}[n] = r'_{oL}[n] - \mathbf{b}'_{N-1}^T[n] \mathbf{r}'_{N-1}[n] \quad (\text{B.121})$$

que puede escribirse de forma más compacta si hacemos uso de los vectores extendidos para los filtros adaptativos predictores hacia atrás, siendo en este caso,

$$e^b[n] = x[n - N] - \hat{x}[n - N] = x[n - N] - \mathbf{b}'_{N-1}^T[n] \mathbf{x}_{N-1}[n] = \hat{\mathbf{b}}'_{N-1}^T[n] \mathbf{x}_N[n] \quad (\text{B.122})$$

donde:

$$\hat{\mathbf{b}}'_{N-1}[n] = (-\mathbf{b}'_{N-1}^T[n], 1)^T \quad (\text{B.123})$$

por lo que ahora, podemos poner la ecuación (B.120) como:

$$\mathbf{R}_N^{-1}[n] = \begin{pmatrix} \mathbf{R}_{N-1}^{-1}[n] & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{pmatrix} + \hat{\mathbf{b}}'_{N-1}[n] \hat{\mathbf{b}}'_{N-1}^T[n] / \alpha'^b[n]. \quad (\text{B.124})$$

Una vez expuestas todas estas expresiones, volvamos sobre la definición de los vectores de *Kalman*. Haciendo uso de las expresiones definidas en (B.90) y (B.91), podemos poner que:

$$\mathbf{G}'_N[n] = \mathbf{R}_N[n]^{-1} \mathbf{x}_N[n] \quad (\text{B.125})$$

$$\mathbf{G}'_N^*[n - L] = \mathbf{R}_N[n]^{-1} \mathbf{x}_N[n - L]. \quad (\text{B.126})$$

Sustituyendo el valor de las matrices inversas calculado en (B.124) tendremos

$$\mathbf{G}'_N^*[n] = \begin{pmatrix} \mathbf{R}_{N-1}^{-1}[n] & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_{N-1}[n] \\ x[n - N] \end{pmatrix} + \hat{\mathbf{b}}'_{N-1}[n] \hat{\mathbf{b}}'_{N-1}^T[n] / \alpha'^b[n] \mathbf{x}_N[n] \quad (\text{B.127})$$

$$\mathbf{G}'_N^*[n] = \begin{pmatrix} \mathbf{G}'_{N-1}^*[n] \\ 0 \end{pmatrix} + \frac{e^b[n]}{\alpha'^b[n]} \widehat{\mathbf{b}}'_{N-1}[n]. \quad (\text{B.128})$$

Análogamente:

$$\mathbf{G}'_N^*[n-L] = \begin{pmatrix} \mathbf{G}'_{N-1}^*[n-L] \\ 0 \end{pmatrix} + \frac{e^b[n-L]}{\alpha'^b[n]} \widehat{\mathbf{b}}'_{N-1}[n]. \quad (\text{B.129})$$

Y usando la expresión de la inversa expresada en la ecuación (B.124), llegaríamos a las siguientes expresiones:

$$\mathbf{G}'_N^*[n] = \begin{pmatrix} 0 \\ \mathbf{G}'_{N-1}^*[n-1] \end{pmatrix} + \frac{e^f[n]}{\alpha'^f[n]} \widehat{\mathbf{a}}'_{N-1}[n]. \quad (\text{B.130})$$

O equivalentemente:

$$\mathbf{G}'_N^*[n-L] = \begin{pmatrix} 0 \\ \mathbf{G}'_{N-1}^*[n-L-1] \end{pmatrix} + \frac{e^f[n-L]}{\alpha'^f[n]} \widehat{\mathbf{a}}'_{N-1}[n]. \quad (\text{B.131})$$

De esta forma, usaríamos las expresiones (B.130) y (B.131) para calcular $\mathbf{G}'_N^*[n]$ y $\mathbf{G}'_N^*[n-L]$ a partir de $\mathbf{G}'_{N-1}^*[n-1]$ y $\mathbf{G}'_{N-1}^*[n-L-1]$, volviendo a las expresiones (B.128) y (B.129) para calcular $\mathbf{G}'_{N-1}^*[n]$ y $\mathbf{G}'_{N-1}^*[n-L]$ a partir de los ya conocidos valores de $\mathbf{G}'_N^*[n]$ y $\mathbf{G}'_N^*[n-L]$.

Además, si dividimos el vector $\mathbf{G}'_N^*[n]$ en dos partes, análogamente a como se realizó en (B.33), tendremos:

$$\mathbf{G}'_N^*[n] = \begin{pmatrix} \overline{\mathbf{G}}'_N^*[n] \\ g'_N^*[n] \end{pmatrix} \quad (\text{B.132})$$

siendo $\overline{\mathbf{G}}'_N^*[n]$ los $N-1$ elementos de $\mathbf{G}'_N^*[n]$, y $g'_N^*[n]$ sería, operando con la ecuación (B.128)

$$g'_N^*[n] = \frac{e^b[n]}{\alpha'^b[n]} \quad (\text{B.133})$$

pudiendo deducir de la ecuación (B.128) la siguiente expresión:

$$\mathbf{G}'_{N-1}^*[n] = \overline{\mathbf{G}}'_N^*[n] + g'_N^*[n] \mathbf{b}'_{N-1}[n]. \quad (\text{B.134})$$

Actuando de forma similar sobre la ecuación (B.129) podemos decir que:

$$\mathbf{G}'_{N-1}[n-L] = \overline{\mathbf{G}}'_N[n-L] + g'_N[n-L]\mathbf{b}'_{N-1}[n] \quad (\text{B.135})$$

siendo, en este caso

$$g'_N[n-L] = \frac{e^b[n-L]}{\alpha'^b[n]}. \quad (\text{B.136})$$

Notar, que de nuevo podemos identificar las ecuaciones (B.128) y (B.129) con (C.19), y las ecuaciones (B.130) y (B.131) con (C.36), obteniendo, en este caso, el valor para los respectivos coeficientes c_L , $\frac{e^b[n]}{\alpha'^b[n]}$, $\frac{e^b[n-L]}{\alpha'^b[n]}$, $\frac{e^f[n]}{\alpha'^f[n]}$, $\frac{e^f[n-L]}{\alpha'^f[n]}$, en cada caso.

Usando las ecuaciones (B.134) y (B.135) en la actualización de los filtros predictores hacia atrás (ecuación (B.104)) obtenemos:

$$\begin{aligned} \mathbf{b}'_{N-1}[n] = & \mathbf{b}'_{N-1}[n-1] + (\overline{\mathbf{G}}'_N[n] + g'_N[n+1]\mathbf{b}'_{N-1}[n])e^b_{pri}[n] \\ & - (\overline{\mathbf{G}}'_N[n-L] + g'_N[n-L]\mathbf{b}'_{N-1}[n])e^b_{pri}[n-L] \end{aligned} \quad (\text{B.137})$$

y despejando $\mathbf{b}'_{N-1}[n]$, tendremos:

$$\begin{aligned} \mathbf{b}'_{N-1}[n](1 - g'_N[n]e^b_{pri}[n] + g'_N[n-L]e^b_{pri}[n-L]) = \\ \mathbf{b}'_{N-1}[n-1] + \overline{\mathbf{G}}'_N[n]e^b_{pri}[n] - \overline{\mathbf{G}}'_N[n-L]e^b_{pri}[n-L] \end{aligned} \quad (\text{B.138})$$

de donde

$$\mathbf{b}'_{N-1}[n] = \frac{\mathbf{b}'_{N-1}[n-1] + \overline{\mathbf{G}}'_N[n]e^b_{pri}[n] - \overline{\mathbf{G}}'_N[n-L]e^b_{pri}[n-L]}{1 - g'_N[n]e^b_{pri}[n] + g'_N[n-L]e^b_{pri}[n-L]} \quad (\text{B.139})$$

que substituyendo el valor de $\mathbf{b}'_{N-1}[n]$ en las ecuaciones (B.134) y (B.135) completamos la recursión, y dejamos preparados los valores de $\mathbf{G}'_{N-1}[n]$ y $\mathbf{G}'_{N-1}[n-L]$ para la siguiente iteración.

Así, un algoritmo adaptativo *fast* RLS con inventanado deslizante sería como se describe a continuación:

En el instante n -ésimo disponemos de los valores de $\mathbf{G}'_{N-1}[n-1]$, $\mathbf{a}_{N-1}[n-1]$, $\alpha'^f[n]$, $\mathbf{b}'_{N-1}[n-1]$ y $\mathbf{w}'_N[n-1]$ y nos llega la siguiente nueva información: $d[n]$ y $x[n]$. Entonces:

- Actualizamos el valor de los vectores de ganancia de *Kalman*
 - $e'_{pri}[n] = x[n] - \mathbf{x}_{N-1}^T[n]\mathbf{a}'_{N-1}[n-1]$
 - $e'_{pri}[n-L] = x[n-L] - \mathbf{x}_{N-1}^T[n-L]\mathbf{a}'_{N-1}[n-1]$
 - $\mathbf{a}'_{N-1}[n] = \mathbf{a}'_{N-1}[n-1] + \mathbf{G}'_{N-1}[n-1]e'_{pri}[n] - \mathbf{G}'_{N-1}[n-L-1]e'_{pri}[n-L]$
 - $e^f[n] = x[n] - \mathbf{x}_{N-1}^T[n]\mathbf{a}'_{N-1}[n]$
 - $e^f[n-L] = x[n-L] - \mathbf{x}_{N-1}^T[n-L]\mathbf{a}'_{N-1}[n]$
 - $\alpha^f[n] = \alpha^f[n-1] + e'_{pri}[n]e^f[n] - e'_{pri}[n-L]e^f[n-L]$
 - $\mathbf{G}'_N[n] = \begin{pmatrix} 0 \\ \mathbf{G}'_{N-1}[n-1] \end{pmatrix} + \frac{e^f[n]}{\alpha^f[n]}\hat{\mathbf{a}}'_{N-1}[n]$
 - $\mathbf{G}'_N[n-L] = \begin{pmatrix} 0 \\ \mathbf{G}'_{N-1}[n-L-1] \end{pmatrix} + \frac{e^f[n-L]}{\alpha^f[n]}\hat{\mathbf{a}}'_{N-1}[n]$
- Reorganizamos los vectores de Ganancia de *Kalman*
 - $\mathbf{G}'_N[n] = \begin{pmatrix} \overline{\mathbf{G}}'_N[n] \\ g'_N[n] \end{pmatrix}$
 - $\mathbf{G}'_N[n-L] = \begin{pmatrix} \overline{\mathbf{G}}'_N[n-L] \\ g'_N[n-L] \end{pmatrix}$
- $e'^b_{pri}[n] = x[n-N] - \mathbf{b}'_{N-1}[n-1]\mathbf{x}_{N-1}[n]$
- $e'^b_{pri}[n-L] = x[n-N-L] - \mathbf{b}'_{N-1}[n-1]\mathbf{x}_{N-1}[n-L]$
- $\mathbf{b}'_{N-1}[n] = \frac{\mathbf{b}'_{N-1}[n-1] + \overline{\mathbf{G}}'_N[n]e'^b_{pri}[n] - \overline{\mathbf{G}}'_N[n-L]e'^b_{pri}[n-L]}{1 - g'_N[n]e'^b_{pri}[n] + g'_N[n-L]e'^b_{pri}[n-L]}$
- $\mathbf{G}'_{N-1}[n] = \overline{\mathbf{G}}'_N[n] + g'_N[n]\mathbf{b}'_{N-1}[n]$
- $\mathbf{G}'_{N-1}[n-L] = \overline{\mathbf{G}}'_N[n-L] + g'_N[n-L]\mathbf{b}'_{N-1}[n]$
- Actualizamos los coeficientes del filtro FIR de mínimos cuadrados
 - $e'_{pri}[n] = d[n] - \mathbf{x}_N^T[n]\mathbf{w}'_N[n-1]$
 - $e'_{pri}[n-L] = d[n-L] - \mathbf{x}_N^T[n-L]\mathbf{w}'_N[n-1]$
 - $\mathbf{w}'_N[n] = \mathbf{w}'_N[n-1] + \mathbf{G}'_N[n]e_{pri}[n] - \mathbf{G}'_N[n-L]e'_{pri}[n-L]$.

En este caso se precisarían $18N - 5$ multiplicaciones para el cálculo de la inversa matricial y la actualización de los coeficientes.

Apéndice C

Recursión de *Levinson*

C.1. Introducción

El algoritmo de *Levinson* [99], fue propuesto para reducir el coste computacional en la resolución de un sistema de ecuaciones $\mathbf{Ax} = \mathbf{y}$, donde la matriz principal \mathbf{A} de dimensiones $L \times L$ es del tipo *Toeplitz* simétrica. Aprovechando las simetrías de esta matriz, se puede calcular el vector \mathbf{x} con un coste computacional aproximado de $2L^2$ productos. El algoritmo comienza con una solución trivial para el sistema escalar de orden 1×1 y a partir de dicha solución, operando iterativamente, se incrementa el orden de las ecuaciones hasta alcanzar el sistema completo $L \times L$. Es decir, que conociendo la solución del sistema para un orden determinado $N - 1$ podemos calcular la solución de orden N . Esto nos resulta especialmente interesante porque en el anexo B, aparecen diferentes ecuaciones donde se relaciona la solución de un sistema de matriz principal *Toeplitz* de orden $N + 1$ con la de orden N . Puesto que dicho proceder se asemeja a lo realizado en un paso del algoritmo de *Levinson*, a continuación se describirá brevemente dicho algoritmo para así poder encontrar las analogías entre las expresiones que se reflejan en el anexo B y el proceder del algoritmo de *Levinson*.

C.2. Recursión de *Levinson*

Consideremos un sistema genérico $\mathbf{R}_{L+1}[n]\mathbf{g}_{L+1}[n] = \mathbf{x}_{L+1}[n]$, del tipo que aparece en la ecuación (B.16), donde $\mathbf{R}_{L+1}[n]$ es una matriz de tamaño $(L + 1) \times (L + 1)$, y tanto $\mathbf{g}_{L+1}[n]$ como $\mathbf{x}_{L+1}[n]$ son vectores de tamaño

$(L + 1) \times 1$.

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_L \end{pmatrix} = \begin{pmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L] \end{pmatrix} \quad (\text{C.1})$$

El algoritmo de *Levinson* nos proporciona la solución del sistema, $\mathbf{g}_{L+1}[n]$, operando del siguiente modo:

Partimos del sistema de orden 1

$$R[0]g_0^0 = x[n], \quad (\text{C.2})$$

con lo cual se obtiene para g_0^0

$$g_0^0 = \frac{x[n]}{R[0]}. \quad (\text{C.3})$$

Además, definimos un coeficiente auxiliar que cumple:

$$R[0]a_0^0 = \alpha_1. \quad (\text{C.4})$$

Eligiendo el valor unidad para a_0^0 tenemos que $\alpha_1 = R[0]$. Esta constante nos ayudará en la siguiente iteración. Una vez inicializado el proceso, el algoritmo de *Levinson* nos permitiría conocer la solución de orden $L + 1$ a partir de la de orden L . La forma de proceder para incrementar el orden del sistema en una iteración genérica sería:

Partimos de \mathbf{g}^{L-1} , \mathbf{a}^{L-1} y α_L conocidos de la iteración anterior, que son los vectores y el escalar que resuelven los siguientes sistemas:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L-1] \\ R[1] & R[0] & \cdots & R[L-2] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} g_0^{L-1} \\ g_1^{L-1} \\ \vdots \\ g_{L-1}^{L-1} \end{pmatrix} = \begin{pmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \end{pmatrix} \quad (\text{C.5})$$

y

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L-1] \\ R[1] & R[0] & \cdots & R[L-2] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} a_0^{L-1} \\ a_1^{L-1} \\ \vdots \\ a_{L-1}^{L-1} \end{pmatrix} = \begin{pmatrix} \alpha_L \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (\text{C.6})$$

El siguiente paso es ampliar el sistema (C.5) aumentando el orden de la matriz:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} g_0^{L-1} \\ g_1^{L-1} \\ \vdots \\ g_{L-1}^{L-1} \\ 0 \end{pmatrix} = \begin{pmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \\ \gamma_L \end{pmatrix} \quad (\text{C.7})$$

γ_L se puede calcular directamente a partir de:

$$\gamma_L = \sum_{i=0}^{L-1} R[L-i]g_i^{L-1}. \quad (\text{C.8})$$

A continuación, ampliaremos el sistema (C.6).

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} a_0^{L-1} \\ a_1^{L-1} \\ \vdots \\ a_{L-1}^{L-1} \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha_L \\ 0 \\ \vdots \\ 0 \\ \beta_L \end{pmatrix}, \quad (\text{C.9})$$

pudiéndose deducir β_L del mismo:

$$\beta_L = \sum_{i=0}^{L-1} R[L-i]a_i^{L-1}. \quad (\text{C.10})$$

Puesto que la matriz principal es *Toeplitz*, podemos invertir el orden de las incógnitas y los coeficientes del sistema en (C.9) sin cambiar la matriz principal, cumpliéndose que:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} 0 \\ a_{L-1}^{L-1} \\ \vdots \\ a_1^{L-1} \\ a_0^{L-1} \end{pmatrix} = \begin{pmatrix} \beta_L \\ 0 \\ \vdots \\ 0 \\ \alpha_L \end{pmatrix}. \quad (\text{C.11})$$

Si restamos a la ecuación (C.9) la ecuación (C.11) multiplicada por una constante k_L :

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \left[\begin{pmatrix} a_0^{L-1} \\ a_1^{L-1} \\ \vdots \\ a_{L-1}^{L-1} \\ 0 \end{pmatrix} - k_L \begin{pmatrix} 0 \\ a_{L-1}^{L-1} \\ \vdots \\ a_1^{L-1} \\ a_0^{L-1} \end{pmatrix} \right] \\ = \begin{pmatrix} \alpha_L \\ 0 \\ \vdots \\ 0 \\ \beta_L \end{pmatrix} - k_L \begin{pmatrix} \beta_L \\ 0 \\ \vdots \\ 0 \\ \alpha_L \end{pmatrix}. \quad (\text{C.12})$$

Eligiendo k_L de forma que el término independiente del sistema sólo tenga un elemento distinto de cero, esto es $k_L = \frac{\beta_L}{\alpha_L}$, obtenemos un sistema equivalente a (C.6) pero de un orden superior, donde tendríamos el vector de coeficientes auxiliares \mathbf{a}^L y la constante α_{L+1} , para la siguiente iteración.

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} a_0^L \\ a_1^L \\ \vdots \\ a_{L-1}^L \\ a_L^L \end{pmatrix} = \begin{pmatrix} \alpha_{L+1} \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad (\text{C.13})$$

con $\alpha_{L+1} = \frac{\alpha_L^2 - \beta_L^2}{\alpha_L}$, y

$$a_0^L = a_0^{L-1} = 1$$

$$a_1^L = a_1^{L-1} - k_L a_{L-1}^{L-1}$$

$$\vdots$$

$$a_i^L = a_i^{L-1} - k_L a_{L-i}^{L-1}$$

$$\vdots$$

$$a_L^L = -k_L a_0^{L-1}.$$

La ecuación (C.13) también permite invertir los coeficientes y las incógnitas gracias a la estructura *Toeplitz* de la matriz principal, obteniendo:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} a_L^L \\ a_{L-1}^L \\ \vdots \\ a_1^L \\ a_0^L \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \alpha_{L+1} \end{pmatrix}, \quad (\text{C.14})$$

así, restando de la ecuación (C.7), la ecuación (C.14) multiplicada por una constante c_L , quedaría,

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \left[\begin{pmatrix} g_0^{L-1} \\ g_1^{L-1} \\ \vdots \\ g_{L-1}^{L-1} \\ 0 \end{pmatrix} - c_L \begin{pmatrix} a_L^L \\ a_{L-1}^L \\ \vdots \\ a_1^L \\ a_0^L \end{pmatrix} \right] \\ = \begin{pmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \\ \gamma_L \end{pmatrix} - c_L \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \alpha_{L+1} \end{pmatrix}. \quad (\text{C.15})$$

Escogiendo c_L de forma que:

$$\gamma_L - c_L \alpha_{L+1} = x[n-L], \quad (\text{C.16})$$

es decir

$$c_L = \frac{\gamma_L - x[n-L]}{\alpha_{L+1}}, \quad (\text{C.17})$$

se obtiene

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} g_0^L \\ g_1^L \\ \vdots \\ g_{L-1}^L \\ g_L^L \end{pmatrix} = \begin{pmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \\ x[n-L] \end{pmatrix}, \quad (\text{C.18})$$

donde

$$g_0^L = g_0^{L-1} - c_L a_L^L \\ g_1^L = g_1^{L-1} - c_L a_{L-1}^L$$

$$\begin{aligned}
& \vdots \\
g_i^L &= g_i^{L-1} - c_L a_{L-i}^L \\
& \vdots \\
g_L^L &= -c_L a_0^L = -c_L
\end{aligned}$$

que expresado de forma matricial quedaría:

$$\mathbf{g}_L = \begin{pmatrix} \mathbf{g}_L^{L-1} \\ 0 \end{pmatrix} - c_L \begin{pmatrix} -k_L \\ a_{L-1}^{L-1} - k_L a_1^{L-1} \\ a_{L-2}^{L-1} - k_L a_2^{L-1} \\ \vdots \\ 1 \end{pmatrix}. \quad (\text{C.19})$$

Esta ecuación es similar a las del tipo (B.28) del anexo B, donde se obtienen los vectores de ganancia de *Kalman* de orden $L + 1$ a partir de los de orden L en el mismo instante de tiempo.

Es de notar, que cuando el sistema de ecuaciones a resolver mediante el algoritmo de *Levinson*, representa las ecuaciones normales de un sistema predictor, los coeficientes c_L que aparecen en dicho algoritmo se denominan coeficientes de reflexión o PARCOR (PARTIAL CORrelation) [100], ya que la solución óptima para estos coeficientes en este caso, puede interpretarse como una correlación cruzada normalizada entre la señal del error de predicción de orden L hacia adelante $e_L^f[n]$ y la del error de predicción de orden L hacia atrás $e_L^b[n - 1]$ ($c_L = \frac{E\{e_L^f[n]e_L^b[n - 1]\}}{E\{e_L^f[n]^2\}}$).

En el anexo B también aparece una recursión parecida pero donde se obtiene el vector de ganancia de *Kalman* de orden $L + 1$ en el instante $n + 1$ a partir del vector de ganancia de *Kalman* de orden L en el instante n -ésimo. En el siguiente apartado se presenta cómo modificar el proceso descrito para poder obtener esta recursión.

C.3. Recursión de *Levinson* modificada

Supongamos que en un instante determinado conocemos los coeficientes del sistema $\mathbf{R}_L[n]\mathbf{g}_L[n] = \mathbf{x}_L[n]$, y a partir de éstos queremos obtener los coeficientes que satisfacen el sistema un orden superior $\mathbf{R}_{L+1}[n + 1]\mathbf{g}_{L+1}[n + 1] = \mathbf{x}_{L+1}[n + 1]$. Este es el problema que se plantea en las ecuaciones del tipo (B.29)

en el anexo B. Obsérvese que la recursión propuesta no es exactamente la que resuelve el algoritmo de *Levinson*, pero bajo determinadas circunstancias podemos obtener una recursión similar modificando ligeramente el procedimiento estándar seguido en el algoritmo de *Levinson*. Así, cuando la matriz $\mathbf{R}_L[n]$ representan los valores de la autocorrelación de una señal estacionaria, como en el caso de la ecuación (B.29), resulta que dicha matriz es del tipo *Toeplitz* simétrica, y además, $\mathbf{R}_{L+1}[n+1] = \mathbf{R}_{L+1}[n]$. Bajo estas circunstancias, se puede desarrollar una recursión del *Levinson* modificada tal y como se describe a continuación:

Se parte del sistema trivial de orden 1

$$R[0]g_0^0[n] = x[n], \quad (\text{C.20})$$

obteniendo igual que en el caso estándar,

$$g_0^0[n] = \frac{x[n]}{R[0]}. \quad (\text{C.21})$$

Volvemos a definir, de nuevo, un coeficiente auxiliar que cumple:

$$R[0]a_0^0 = \alpha_1. \quad (\text{C.22})$$

que considerando $a_0^0 = 1$ nos proporciona el primer valor para el coeficiente auxiliar, $\alpha_1 = R[0]$. A partir de aquí, volvemos a intentar alcanzar una expresión recursiva que nos permitiría conocer la solución del sistema de orden $L + 1$ en el instante $n + 1$ a partir de la solución del sistema de orden L en el instante n . Para ello, partimos de los valores $\mathbf{g}^{L-1}[n]$, \mathbf{a}^{L-1} y α_L conocidos de la iteración anterior, y que aparecen en los siguientes sistemas:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L-1] \\ R[1] & R[0] & \cdots & R[L-2] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} g_0^{L-1}[n] \\ g_1^{L-1}[n] \\ \vdots \\ g_{L-1}^{L-1}[n] \end{pmatrix} = \begin{pmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \end{pmatrix} \quad (\text{C.23})$$

y

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L-1] \\ R[1] & R[0] & \cdots & R[L-2] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} a_0^{L-1} \\ a_1^{L-1} \\ \vdots \\ a_{L-1}^{L-1} \end{pmatrix} = \begin{pmatrix} \alpha_L \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (\text{C.24})$$

El siguiente paso es ampliar el sistema (C.23) aumentando el orden de la matriz, de forma que los valores existentes encajen en el nuevo sistema buscado $\mathbf{R}_{L+1}[n+1]\mathbf{g}_{L+1}[n+1] = \mathbf{x}_{L+1}[n+1]$. Así:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} 0 \\ g_0^{L-1}[n] \\ g_1^{L-1}[n] \\ \vdots \\ g_{L-1}^{L-1}[n] \end{pmatrix} = \begin{pmatrix} \gamma_L \\ x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \end{pmatrix} \quad (\text{C.25})$$

siendo ahora γ_L ,

$$\gamma_L = \sum_{i=0}^{L-1} R[i+1]g_i^{L-1}[n]. \quad (\text{C.26})$$

El sistema (C.24), se amplía de la misma forma que en el caso estándar.

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} a_0^{L-1} \\ a_1^{L-1} \\ \vdots \\ a_{L-1}^{L-1} \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha_L \\ 0 \\ \vdots \\ 0 \\ \beta_L \end{pmatrix}, \quad (\text{C.27})$$

Obteniendo el mismo valor para β_L :

$$\beta_L = \sum_{i=0}^{L-1} R[L-i]a_i^{L-1}. \quad (\text{C.28})$$

De nuevo, aprovechando la estructura *Toeplitz* de la matriz principal en la ecuación (C.27), invertimos el orden de las incógnitas y los coeficientes del sistema, obteniendo:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} 0 \\ a_{L-1}^{L-1} \\ \vdots \\ a_1^{L-1} \\ a_0^{L-1} \end{pmatrix} = \begin{pmatrix} \beta_L \\ 0 \\ \vdots \\ 0 \\ \alpha_L \end{pmatrix}, \quad (\text{C.29})$$

que operando igual que en el caso estándar (restando de la ecuación (C.27) la ecuación (C.29) multiplicada por una constante k_L), nos permite escribir:

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \left[\begin{pmatrix} a_0^{L-1} \\ a_1^{L-1} \\ \vdots \\ a_{L-1}^{L-1} \\ 0 \end{pmatrix} - k_L \begin{pmatrix} 0 \\ a_{L-1}^{L-1} \\ \vdots \\ a_1^{L-1} \\ a_0^{L-1} \end{pmatrix} \right] \\ = \begin{pmatrix} \alpha_L \\ 0 \\ \vdots \\ 0 \\ \beta_L \end{pmatrix} - k_L \begin{pmatrix} \beta_L \\ 0 \\ \vdots \\ 0 \\ \alpha_L \end{pmatrix}. \quad (\text{C.30})$$

Fijamos k_L para que en el término independiente del sistema sólo tengamos un elemento distinto de cero, resultando que $k_L = \frac{\beta_L}{\alpha_L}$. Así obtenemos un sistema equivalente a (C.24) pero de un orden superior, con los nuevos coeficientes auxiliares \mathbf{a}^L y el valor de la constante α_{L+1} , para la siguiente iteración.

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} a_0^L \\ a_1^L \\ \vdots \\ a_{L-1}^L \\ a_L^L \end{pmatrix} = \begin{pmatrix} \alpha_{L+1} \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad (\text{C.31})$$

con $\alpha_{L+1} = \frac{\alpha_L^2 - \beta_L^2}{\alpha_L}$, y

$$\begin{aligned} a_0^L &= a_0^{L-1} = 1 \\ a_1^L &= a_1^{L-1} - k_L a_{L-1}^{L-1} \\ &\vdots \\ a_i^L &= a_i^{L-1} - k_L a_{L-i}^{L-1} \\ &\vdots \\ a_L^L &= -k_L a_0^{L-1}. \end{aligned}$$

Estas constantes coincidirían con las calculadas en el caso estándar.

Ahora, restando de la ecuación (C.25), la ecuación (C.31) multiplicada por una constante c_L , quedaría,

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} 0 \\ g_0^{L-1}[n] \\ g_1^{L-1}[n] \\ \vdots \\ g_{L-1}^{L-1}[n] \end{pmatrix} - c_L \begin{pmatrix} a_0^L \\ a_1^L \\ \vdots \\ a_{L-1}^L \\ a_L^L \end{pmatrix} \\ = \begin{pmatrix} \gamma_L \\ x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \end{pmatrix} - c_L \begin{pmatrix} \alpha_{L+1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (\text{C.32})$$

Escogiendo c_L de forma que:

$$\gamma_L - c_L \alpha_{L+1} = x[n+1], \quad (\text{C.33})$$

es decir

$$c_L = \frac{\gamma_L - x[n+1]}{\alpha_{L+1}}, \quad (\text{C.34})$$

se obtiene

$$\begin{pmatrix} R[0] & R[1] & \cdots & R[L] \\ R[1] & R[0] & \cdots & R[L-1] \\ \vdots & & \ddots & \vdots \\ R[L-1] & R[L-2] & \cdots & R[1] \\ R[L] & R[L-1] & \cdots & R[0] \end{pmatrix} \begin{pmatrix} g_0^L[n+1] \\ g_1^L[n+1] \\ \vdots \\ g_{L-1}^L[n+1] \\ g_L^L[n+1] \end{pmatrix} = \begin{pmatrix} x[n+1] \\ x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \end{pmatrix}, \quad (\text{C.35})$$

donde

$$\begin{aligned} g_0^L[n+1] &= -c_L a_0^L = -c_L \\ g_1^L[n+1] &= g_0^{L-1}[n] - c_L a_1^L \\ &\vdots \\ g_i^L[n+1] &= g_{i-1}^{L-1}[n] - c_L a_i^L \\ &\vdots \\ g_L^L[n] &= g_{L-1}^{L-1}[n] - c_L a_L^L \end{aligned}$$

que expresado de forma matricial quedaría:

$$\mathbf{g}_L[n+1] = \begin{pmatrix} 0 \\ \mathbf{g}_{L-1}[n] \end{pmatrix} - c_L \begin{pmatrix} 1 \\ a_1^{L-1} - k_L a_{L-1}^{L-1} \\ a_2^{L-1} - k_L a_{L-2}^{L-1} \\ \vdots \\ -k_L \end{pmatrix}, \quad (\text{C.36})$$

con lo que se obtiene una ecuación similar a las del tipo de la ecuación (B.29) en el anexo B.

Apéndice D

Simulador software de sistemas acústicos

D.1. Introducción

El simulador acústico es una aplicación desarrollada para *matlab* (versiones 5.x) que facilita todo tipo de simulaciones multicanal que involucren generación y monitorización de audio, automatizando el proceso acústico natural que sufre el sonido (propagación, reflexión,...) a partir de los modelos descritos en las respuestas impulsionales entre diferentes puntos del espacio. Dicho simulador está concebido como un código abierto, donde el programador puede incluir el procesamiento de datos oportuno para lograr los objetivos propuestos, olvidándose de programar todo lo referente a la parte acústica, por lo que resulta adecuado para simular el comportamiento de algoritmos de procesamiento multicanal de audio destinados a ejecutarse en procesadores digitales de señal (DSP) o cualquier tipo de computadores dotados de tarjetas de adquisición y generación de datos, sin más que definir el código del algoritmo siguiendo unas sencillas reglas de interacción con el resto del programa que a continuación se detallarán. Además, el simulador implementa algunas otras facilidades como el almacenamiento de la señales que intervienen en la simulación, capacidad para reproducir el sonido existente en los micrófonos del sistema simulado, análisis de las señales y de los coeficientes de los filtros adaptativos (si los hubiera), o herramientas para la definición, modificación o estudio de los sistemas acústicos. Dicho simulador ha sido desarrollado para estudiar en un sistema real los algoritmos adaptativos propuestos en esta tesis doctoral sin necesidad de programarlos en una plataforma DSP ni tener un sistema acústico in situ. Además, el simulador se presenta como una herramienta imprescindible que facilita el tránsito entre la concepción matemática

inicial del cualquier algoritmo y su programación final en cualquier plataforma digital. Aunque las ventajas y capacidades del simulador son muchas, a continuación se describen sus características principales.

D.2. Breve descripción de la barra de menús

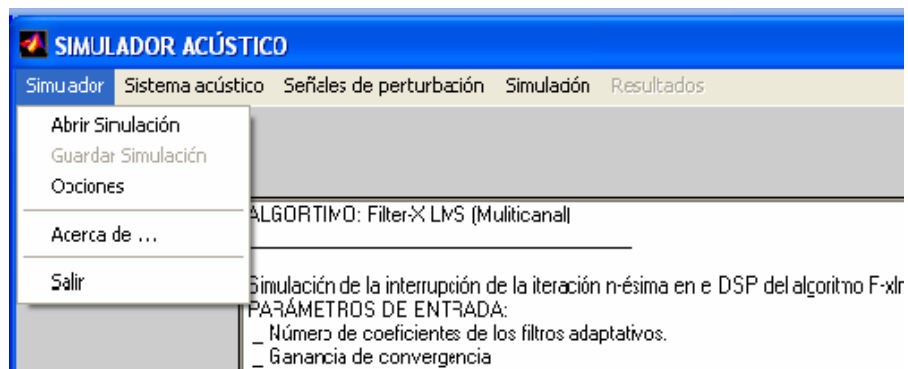


Figura D.1: Menú principal y submenú de la opción *Simulador*.

La barra del menú principal se compone de las siguientes opciones con sus correspondientes submenús:

1. Simulador:

- Abrir simulación: sirve para abrir archivos donde se hayan guardado simulaciones realizadas previamente con el simulador. Los archivos de simulación tienen extensión *sim*.
- Guardar simulación: inicialmente desactivada, se activa tras la realización de una simulación e invoca el cuadro de *guardar como...* para almacenar los resultados y el entorno de la simulación realizada. Estos resultados pueden recuperarse mediante la opción descrita anteriormente.
- Opciones: abre un cuadro de diálogo (figura D.2) donde se pueden modificar diversas opciones relativas al funcionamiento del programa. Permite fijar:
 - La frecuencia del sistema, siendo esta la frecuencia a la que se reproducen las señales cuando se desean escuchar o a la que se graban cuando se decide almacenar en un archivo **.wav*

OPCIONES GENERALES

OPCIONES DE REPRODUCCIÓN Y GRABACIÓN:

Frecuencia de muestreo del sistema (Hz): 8192

OPCIONES PARA LA DEFINICIÓN DE LAS PERTURBACIONES:

Canal de los archivos wav estéreo a usar: 1

Factor de amplificación: 2

Factor de atenuación: 2

Valor de la componente de continua a sumar: 1

Valor de la componente de continua a decrementar: 1

OPCIONES DE VISUALIZACIÓN:

Número de puntos de los espectros frecuenciales: 250

Número de bloques para la visualización de la evolución espectral: 20

Solape entre bloques (% de muestras) para la visualización de la evolución espectral: 50

Factor de olvido para el cálculo de la potencia instantánea: 0.999

OPCIONES DE ALMACENAMIENTO:

Guardar coeficientes de los filtros adaptativos en las iteraciones múltiples de: 50

Cancelar Aceptar

Figura D.2: Cuadro para modificar las opciones del simulador.

- El canal de los archivos *wav* estéreo a usar. Lo que se fija en este caso es si a la hora de definir una señal mono para ser usada en el simulador que se encuentra en un archivo estéreo *wav*, ésta se corresponde con el canal 1 (izquierdo) o el canal 2 (derecho) de dicho archivo.
- El factor de amplificación, que es el factor por el que se amplificaran las señales de perturbación en el proceso de definición de las mismas, si se decide amplificar.
- El factor de atenuación, que es análogo al caso anterior.
- Valor de la componente continua a sumar, donde se fija la constante a añadir a la señal de perturbación en el proceso de definición de las mismas si se decide añadir componente de continua.
- Valor de la componente continua a decrementar, que es análogo al caso anterior pero para decrementar el valor de continua.
- Número de puntos de los espectros frecuenciales, para fijar el número de puntos con que se calculan las FFT cuando se decide hacer uso de alguna opción de visualización en el dominio de

la frecuencia.

- Número de bloques para la visualización de la evolución espectral, donde se fijan los bloques temporales usados en los cálculos de los espectros tiempo-frecuencia (espectrograma y evolución espectral).
 - Solape entre bloques, donde se fija el tanto por ciento de muestras que se solapan en dos de los bloques temporales contiguos definidos en el punto anterior.
 - Factor de olvido para el cálculo de la potencia de las señales, donde se fija el ancho de la ventana exponencial que se usa para la estimación de dichas potencias que posteriormente se usarán en los cálculos de la atenuación (cuanto más cerca de la unidad, más peso tiene la historia pasada en el cálculo, y cuanto más próximo esté este valor a cero, más depende la potencia instantánea de la señal).
 - Por, último, también se puede fijar, cada cuantas iteraciones almacenamos los coeficientes de los filtros adaptativos, si los hubiera en el algoritmo a simular.
- Acerca de... : Abre un cuadro de diálogo con información acerca del programa.
 - Salir: Cierra el programa.

2. Sistema acústico

- Cargar sistema acústico: sirve para abrir un archivo que define un sistema acústico. Estos archivos, tienen la extensión *aco*, y almacenan la información necesaria para la introducción del sistema acústico en las simulaciones. Los sistemas acústicos quedan definidos por el número de señales de referencias (fuentes primarias), número de fuentes secundarias y número de sensores de error, junto con las respuestas impulsionales de los caminos acústicos y la estimación de estos más las respuestas de los caminos inversos si fuera necesario. Es decir, que el sistema acústico define un recinto cerrado y la posición donde se encuentran las fuentes primarias de sonido, los sensores de error y las fuentes secundarias. El simulador permitirá habilitar o deshabilitar los elementos que se consideren oportunos para cada simulación.
- Definir sistema acústico: aunque los sistemas acústicos pueden definirse a partir de respuestas impulsionales reales medidas con herramientas desarrolladas para ello, el simulador también ofrece la posibilidad de definir o manipular sistemas acústicos al gusto del pro-

gramador. Para ello, esta opción del menú se desglosa en dos submenús:

- **Modificar actual:** inicialmente desactivado, se activa cuando se abre un sistema acústico existente o cuando se define uno nuevo. Sirve para manipular el sistema acústico (modificar los coeficientes de la respuestas impulsionales, añadir ruido, ...). Abre el cuadro de dialogo definido en la figura D.4, que se describe más adelante.
- **Definir nuevo:** con esta opción se puede definir un sistema acústico nuevo desde cero. Para ello, primeramente se abre el cuadro de diálogo representado en la figura D.3, donde se fijan el número de fuentes primarias, fuentes secundarias y sensores de error, así como el número de coeficientes para los caminos primarios, caminos secundarios y caminos secundarios estimados. Una vez definidos los parámetros anteriores del sistema

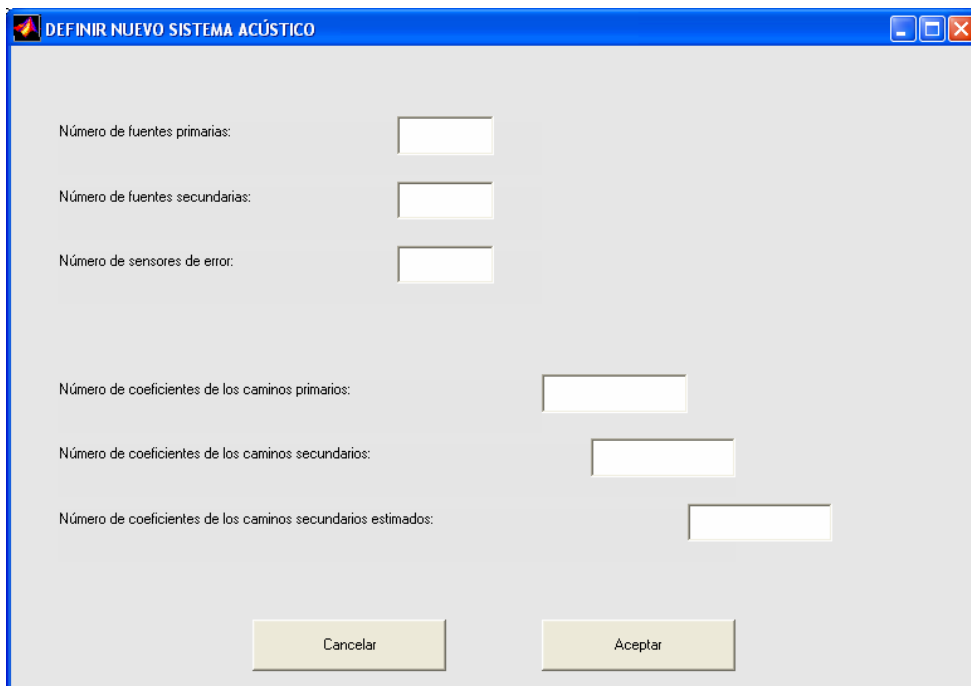


Figura D.3: Cuadro inicial para la definición de un sistema acústico.

acústico, pasaríamos a describir el resto del sistema, es decir, los coeficientes de las respuestas impulsionales del sistema acústico. Para ello se abre el cuadro representado en la figura D.4,

que es el mismo que se usa para modificar un sistema acústico existente.

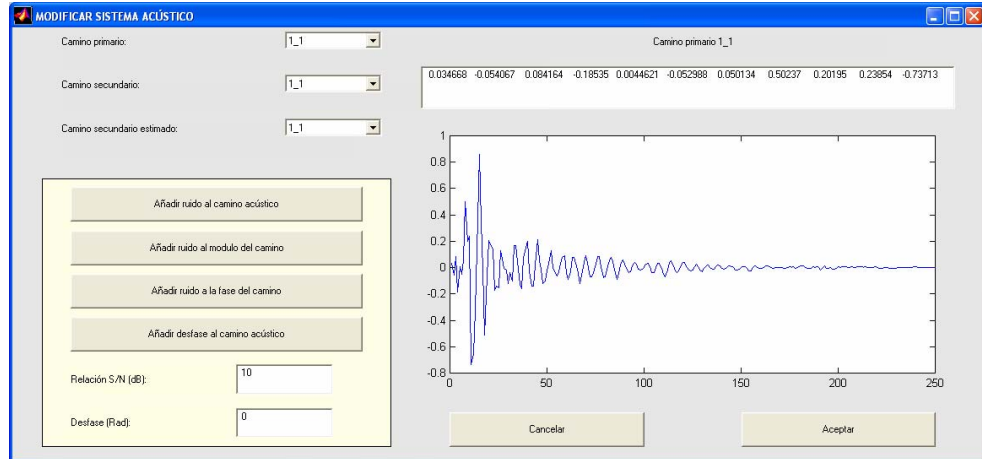


Figura D.4: Cuadro para definir o modificar un sistema acústico.

Según se puede apreciar en la figura D.4, en la parte superior izquierda de la ventana elegimos el camino acústico que queremos definir o modificar, y en la parte derecha aparece tanto numéricamente (parte superior) como gráficamente (parte central), la respuesta del camino acústico. Situándonos sobre la parte numérica, se pueden modificar los coeficientes, quedando reflejado en la representación gráfica. En la parte inferior izquierda, podemos modificar el camino acústico seleccionado, añadiendo ruido a la respuesta impulsional, a su módulo o fase, o añadir un desfase, modificando los campos correspondientes a la relación señal a ruido o desfase, dentro de la misma ventana.

- Estudiar sistema acústico: se habilita cuando existe un sistema acústico definido. Abre la ventana D.5 y sirve para visualizar y comparar las respuestas impulsionales de los caminos acústicos tanto en el dominio del tiempo como en la frecuencia. También podemos calcular las respuestas de los caminos inversos.
 - Grabar sistema acústico: Al igual que la opción anterior, se habilita al definir o cargar un sistema acústico. Abre el cuadro de diálogo de *Guardar como...* para almacenar el sistema acústico en uso. Los sistemas almacenados, pueden ser abiertos en cualquier momento con la opción de abrir sistema acústico.
3. Señales de perturbación: inicialmente está deshabilitada y se habilita al

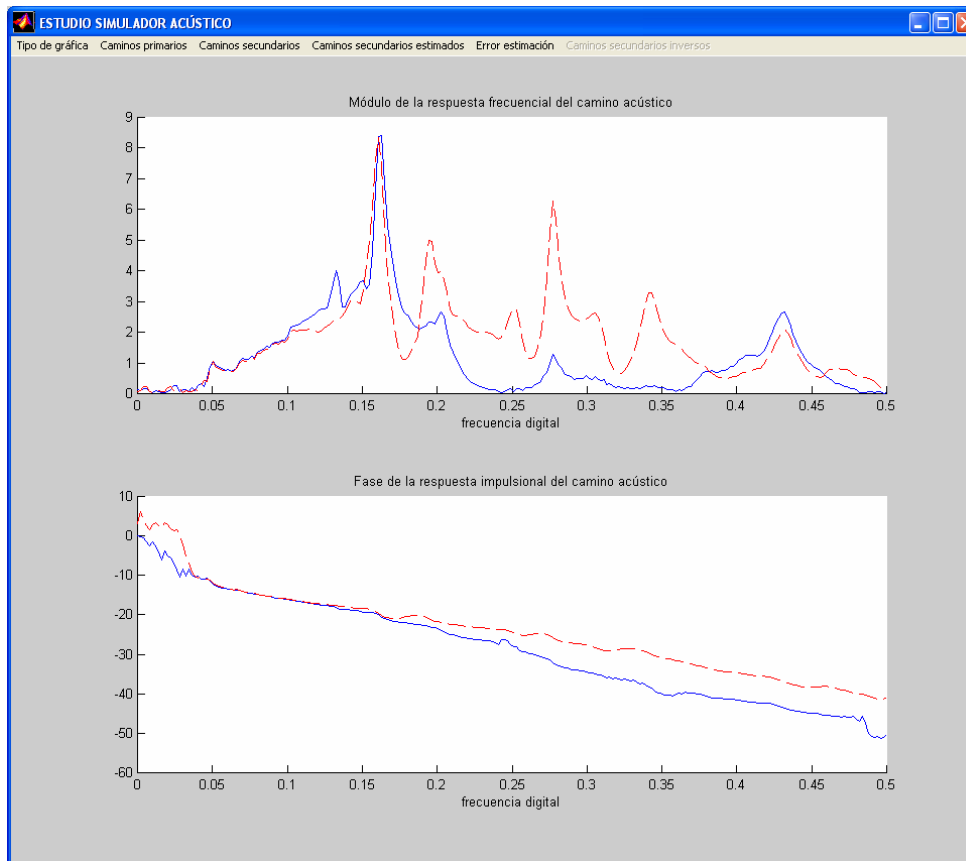


Figura D.5: Cuadro para estudiar el sistema acústico.

cargar un sistema acústico, definirlo o abrir una simulación existente. Es un menú que se confecciona dinámicamente y se habilitan en él tantos elementos de selección como posibles señales perturbadoras (número de fuentes primarias) acepte el sistema acústico existente. Los elementos de este menú se nombrarán $Fuente_i$, siendo i el número de orden de la fuente perturbadora. Cada fuente perturbadora que intervenga en las simulaciones debe tener asociada una señal de perturbación que se definirá pulsando sobre el elemento de este menú que hace referencia a dicha fuente. De esta forma se abrirá la ventana reflejada en la figura D.6. En dicho cuadro podemos visualizar la señal de perturbación tanto en el dominio del tiempo como en el de la frecuencia. También, en la parte superior izquierda tenemos varios botones para elegir el origen de una señal de perturbación ya definida. Así, podemos elegir una señal procedente de un archivo *wav*, o creada con el simulador (con exten-

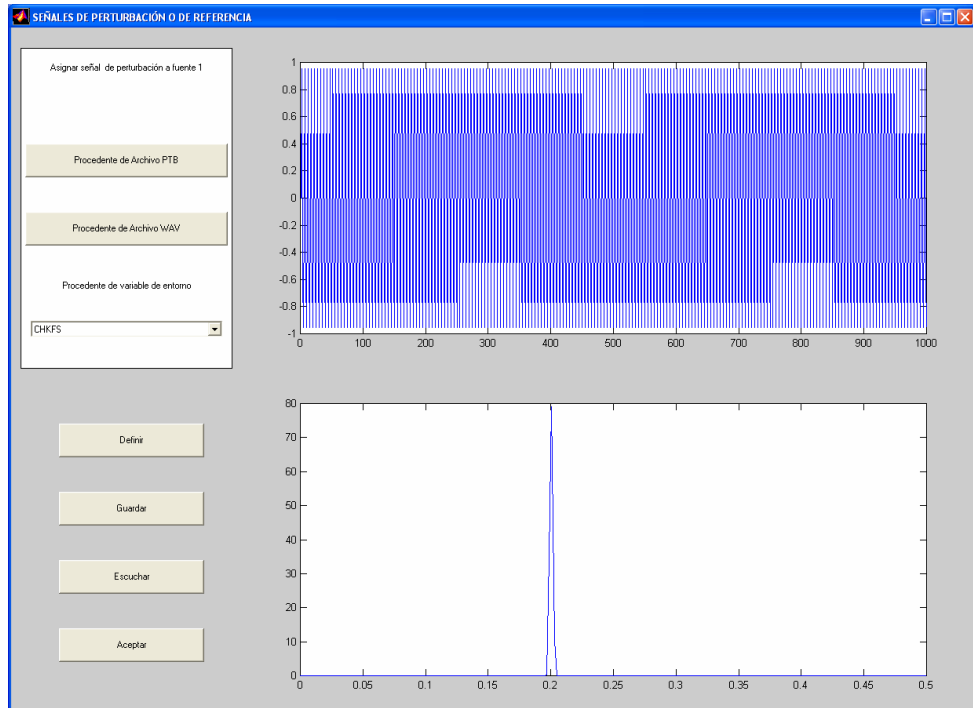


Figura D.6: Cuadro para definir las señales de perturbación.

ción *ptb*), o definida por alguna de las variables existentes en el área de trabajo. Si pulsamos sobre los botones *Procedente de archivo PTB* o *Procedente de archivo WAV*, se abrirá el cuadro de diálogo de *abrir como...* para que podamos elegir el archivo fuente de la señal de perturbación. En caso de que la perturbación proceda de un archivo *wav* estéreo, se asignará el canal definido en las opciones generales del programa dentro del menú *Simulador*. Para asignar una variable del entorno a la perturbación solo hay que seleccionar la variable deseada dentro del menú desplegable existente para dicho uso, donde se encuentran todas las variables unidimensionales existentes en el área de trabajo.

Aparte de escoger perturbaciones existentes en variables o archivos, también se pueden definir éstas usando la herramienta diseñada en el simulador y que se abre pulsando el botón *Definir*. Con esta herramienta podemos crear señales de perturbación de armónicos, señales aleatorias de cualquier tipo de distribución estadística, o señales de tipo chirp. Al margen de esto, también podemos manipular señales de perturbación ya definidas, amplificándolas o atenuándolas, quitándoles la componente

continúa o modificándoles la misma.

Por último, indicar que una vez asignada una señal de perturbación, ésta puede escucharse (a la frecuencia de reproducción del sistema definida en las opciones del simulador dentro del menú *Simulador*) o guardarse en un archivo de perturbación que posteriormente podrá ser usado en cualquier momento mediante la opción de asignar perturbación procedente de archivo *PTB*.

4. Simulación: en este menú se recogen todas las opciones relativas a las simulaciones. En él encontramos los siguientes submenús:
 - Elegir algoritmo de simulación: despliega un submenú donde poder elegir el algoritmo con el que se desea ejecutar la simulación. Una vez seleccionado aparece en la pantalla principal una breve descripción del mismo y el resto de opciones de simulación se ajustan en función de dicho algoritmo. La figura D.7 visualiza un ejemplo de esto.

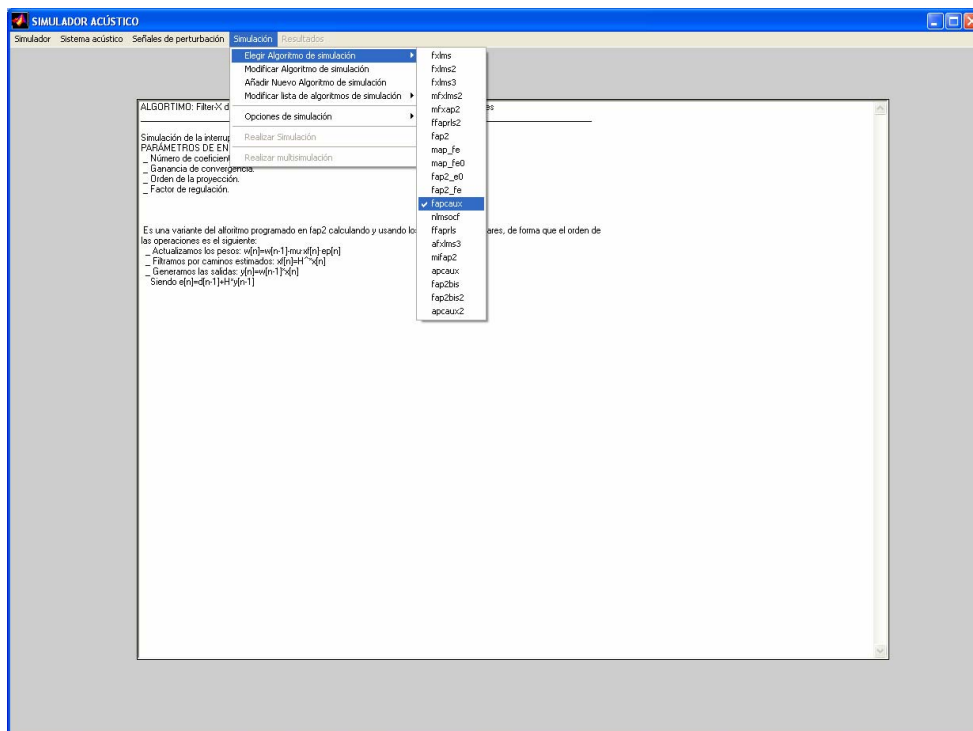


Figura D.7: Menú desplegable para la elección del algoritmo a simular.

- Modificar algoritmo de simulación: sirve para editar el código del algoritmo seleccionado por si se desea realizar alguna modificación del mismo. Abre el editor de textos de *matlab* con el código del algoritmo a modificar.
- Añadir nuevo algoritmo de simulación: opción análoga a la anterior pero tras abrir un cuadro de diálogo para introducir un identificativo del nuevo algoritmo, activa el editor de textos de *matlab* con una plantilla para la introducción del código del nuevo algoritmo.
- Modificar lista de algoritmos: opción que se subdivide en dos (añadir algoritmos a la lista y eliminar algoritmos de la lista) y que sirve justo para esto, ya que a veces no conviene tener todos los algoritmos programados en el menú desplegable (para mejorar la manejabilidad de la lista) y es preferible quitar algunos. Cuando sea necesario podremos volver a añadirlos ya que los algoritmos que no aparecen en la lista y están programados, se almacenan en otra lista a la que se accede desde la opción *Añadir algoritmos a la lista*.
- Opciones de simulación: en este elemento del menú definimos las opciones particulares bajo las cuales se realizará la simulación. Se subdivide en dos apartados:
 - Opciones de simulación generales, donde fijamos el número de muestras a computar, y seleccionamos de las fuentes primarias, fuentes secundarias y sensores de error, cuales están activos para la simulación a ejecutar.
 - Opciones de simulación del algoritmo particular, donde se fijan los parámetros de inicialización del algoritmo elegido para la simulación. Cada algoritmo tiene los suyos propios por lo que el cuadro de diálogo al que se accede desde esta opción es propio de cada algoritmo.
- Realizar simulación: una vez se han definido unas opciones de simulación válidas (dentro de las posibles según el sistema acústico), las perturbaciones y el algoritmo de simulación elegidos, se habilita esta opción que inicialmente estaba desactivada y pulsando sobre ella podremos realizar la simulación pertinente. Tras seleccionar dicha opción se abre un cuadro de diálogo (figura D.8) donde se nos presenta un resumen de todas las opciones de simulación elegidas así como una estimación del tiempo de simulación y el número de flops por iteración que *matlab* empleará.

En dicho cuadro tenemos la opción de aceptar (realizamos la simulación), cancelar (no realizamos la simulación) o continuar, que en caso de que se acabe de realizar una simulación con las mismas opciones de simulación se habilita y nos permite repetir la simulación

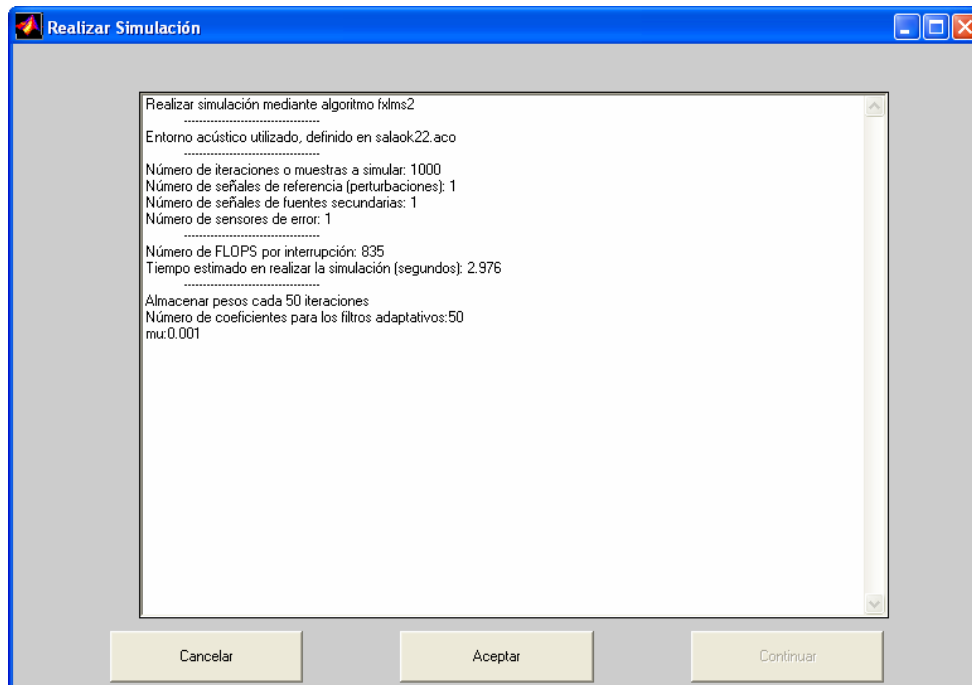


Figura D.8: Cuadro de diálogo para realizar la simulación.

sin inicializar variables como los *buffer* de datos o coeficientes de los filtros adaptativos.

Si aceptamos o continuamos una simulación, esta comenzará a realizarse mostrándose dinámicamente el progreso de la misma mediante una barra de estado. Una vez finalizada la simulación, se nos mostrarán los resultados en cuanto a atenuación obtenidos en cada uno de los sensores de error y un cuadro similar al de la figura D.8 aparecerá pero con los tiempo de simulación y número de flops por iteración calculados a posteriori (no estimados).

5. Resultados: una vez realizadas las simulaciones, este menú se activa y nos permite explorar los resultados obtenidos. Inicialmente se subdivide en dos elementos diferenciados para poder estudiar los resultados tanto de las señales recogidas en los sensores de error como de los coeficientes de los filtros adaptativos (si los algoritmos los usaran). Cada una de estas opciones abre un cuadro de diálogo para interactuar con todas las opciones de presentación de resultados:
 - Señales en sensores de error: dicha opción abre el cuadro de diálogo

de la figura D.9 donde al margen de las opciones de visualización, también podemos escuchar las señales, guardarlas en archivos de audio (*wav*) o almacenarlas en formato de *matlab*. En la parte iz-

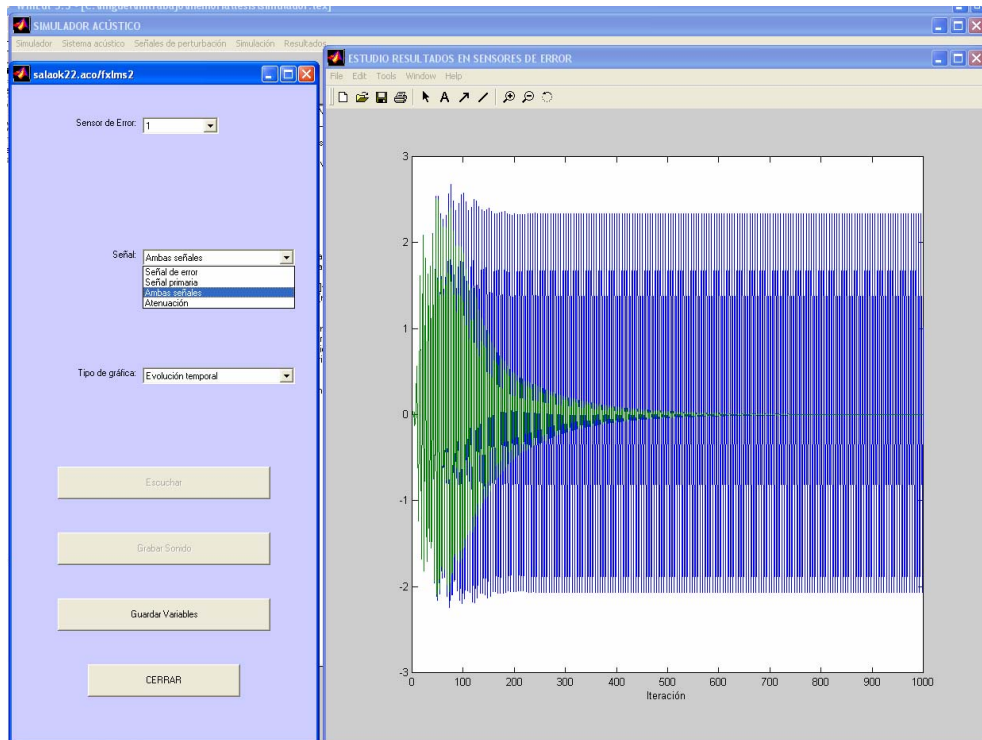


Figura D.9: Cuadro de diálogo para estudiar los resultados en los sensores de error.

quierda de la figura D.9 nos encontramos con la ventana de control que nos permite ejecutar la acción deseada sobre los resultados, mientras que en la parte derecha se encuentra la ventana de visualización donde se van representando gráficamente aquello que se selecciona en la ventana de control. En dicha ventana tenemos la opción de seleccionar el sensor de error sobre el que queremos realizar las oportunas operaciones de visualización de resultados. Se puede visualizar las señales de perturbación (como si no actuara el algoritmo), la señal de error o señal residual (cuando sí que actúa el algoritmo y que es la suma acústica de las señales generadas que llegan al sensor y la de perturbación), ambas señales a la vez, y la gráfica de atenuación (que es la representación de la evolución temporal de la potencia instantánea de la señal residual respecto de

la señal de perturbación). Todo esto en cuanto a la evolución temporal, pero existen otro tipo de gráficas que también pueden ser visualizadas como el espectrograma (tanto de la señal de referencia como de la señal residual, en el sensor de error que se encuentre seleccionado), o la evolución espectral (donde se podrá apreciar como evolucionan los espectros de la señal de referencia frente a la señal residual con el tiempo). Para el cálculo tanto del espectrograma como de la evolución espectral se usan las opciones y parámetros especificados en el submenú de opciones dentro del menú simulador. Al margen de la opciones de visualización, también podemos realizar otras operaciones sobre las señales en los sensores de error: escucharlas, grabarlas como archivo de audio (*wav*) o grabarlas como variables de *matlab*. Para cada una de estas operaciones existen los correspondientes botones en la parte inferior izquierda de la ventana de control.

- Coeficientes de los filtros adaptativos: esta opción sirve para estudiar los coeficientes de los filtros adaptativos cuando estos intervienen en los algoritmos. En las opciones del simulador podemos decidir cada cuantas iteraciones queremos almacenar dichos coeficiente para su posterior estudio. Al seleccionar esta opción en la pantalla aparecen las ventanas que se reflejan en la figura D.10. Igual que anteriormente, en la parte izquierda nos encontramos con la ventana de control donde determinamos qué representación gráfica acerca de los coeficientes queremos visualizar, y en la derecha se encuentra la ventana de visualización donde se dibujan dichos resultados. Tendremos opción de estudiar tantos filtros adaptativos como señales de perturbación para cada fuente secundaria existan en la simulación y podremos realizar dichos estudios tanto para los filtros adaptativos en bloque como para los coeficientes (independientemente) de cada filtro. Decidimos sobre qué filtro adaptativo queremos realizar el estudio seleccionando la señal de perturbación y la fuente secundaria oportuna en la ventana de control, así como la iteración sobre la cual queremos visualizar el valor de los coeficientes. También podemos realizar una animación con la evolución completa de los coeficientes del filtro adaptativo pulsando el botón *Evolución completa*.

D.3. Definición de los algoritmos

Otra cosa a tener en cuenta al margen del funcionamiento básico de los menús del simulador, es cómo definir los algoritmos para su correcto funcio-

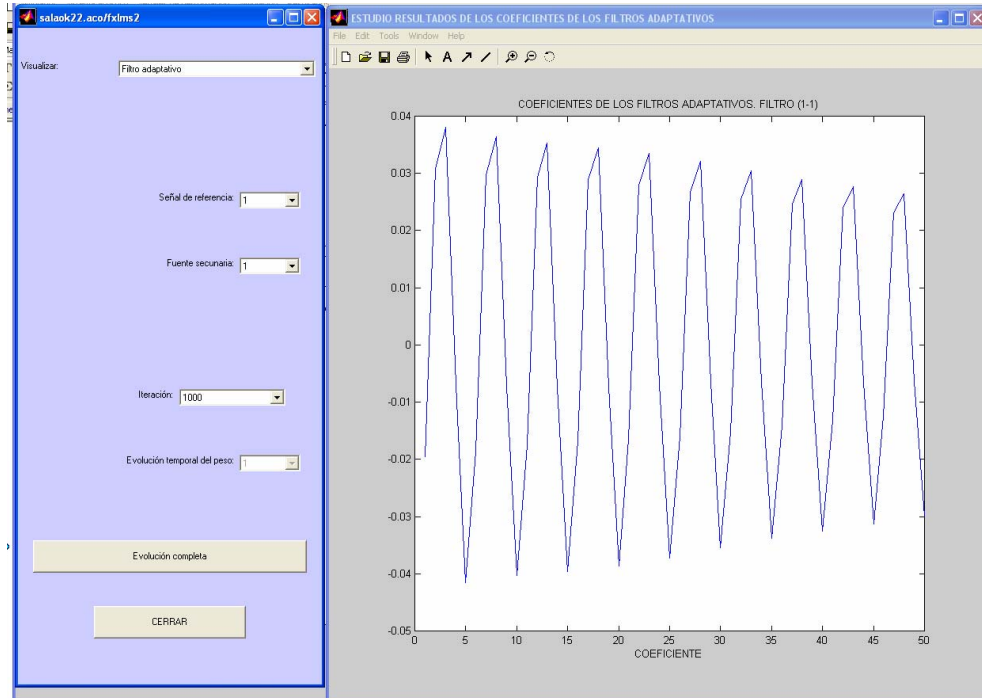


Figura D.10: Cuadro de diálogo para estudiar los resultados en los filtros adaptativos.

namiento dentro del simulador.

Un algoritmo nuevo debe definirse a partir de la opción *Añadir nuevo algoritmo de simulación* dentro del menú *Simulación*, tras lo cual después de haber definido una cadena de texto identificativa del nuevo algoritmo, se abrirán en el editor de *matlab* tres archivos (plantillas) para introducir el código de forma adecuada. Tras guardar estos tres archivos, el nuevo algoritmo quedará insertado en la lista de algoritmos (*elegir algoritmo de simulación* dentro del menú *Simulación*).

Cada uno de estos archivos se llamará del mismo modo que el identificador del algoritmo y se guardarán en directorios diferentes. La información que debe incluir cada archivo y su funcionalidad se describe a continuación:

- Archivo almacenado en el subdirectorio *Algoritmos* dentro del directorio *Simulador*: es donde propiamente se describe el código del algoritmo. Este archivo es el que interactúa con el simulador en cada iteración (se ejecuta en cada iteración). Para una correcta interacción es necesario tener en cuenta las siguientes variables globales que maneja el simulador:

- *e*: variable donde se almacenan los valores adquiridos por los sensores de error.
- *in*: variable donde se almacenan los valores adquiridos de las señales de referencia.
- *out*: variable donde hay que escribir los valores de las señales a generar.
- *H*: matriz que contiene las respuestas impulsionales de los caminos secundarios estimados. Cada camino secundario está en una columna diferente y el orden es el siguiente:

$$[H_{1-1}, H_{1-2}, \dots, H_{1-NMIC}, H_{2-1}, H_{2-2}, \dots, H_{2-NMIC}, H_{NSEC-1}, H_{NSEC-2}, \dots, H_{NSEC-NMIC}]$$
- *NPRI*: número de fuentes primarias o señales de referencia del sistema.
- *NSEC*: número de fuentes secundarias.
- *NMIC*: número de sensores de error.

Si además en el algoritmo se manejan filtros adaptativos cuyos coeficientes se desean analizar posteriormente en el simulador, entonces se deberían almacenar en una variable llamada *coefA* que tendría tantas filas como coeficientes y tantas columnas como filtros adaptativos, almacenándolos del siguiente modo:

$$[FA_{1-1}, FA_{1-2}, \dots, FA_{1-NSEC}, FA_{2-1}, FA_{2-2}, \dots, FA_{2-NSEC}, FA_{NPRI-1}, FA_{NPRI-2}, \dots, FA_{NPRI-NSEC}]$$

Por tanto el algoritmo debería leer de las variables *in* y *e*, realizar las operaciones oportunas con ellas y escribir en la variable *out*.

- Archivo almacenado en el subdirectorio *Algoritmos\ini* dentro del directorio *Simulador*: este archivo sólo se ejecutará en la primera iteración y sirve para definir e inicializar las variables que intervienen en el algoritmo. Si vamos a usar filtros adaptativos cuyos coeficientes puedan ser analizados posteriormente deberemos definir obligatoriamente dos variables:
 - *numerocoeff*: variable que almacena el número de coeficientes de los filtros adaptativos y que bien puede definirse en este archivo o en el archivo donde se declaran los parámetros de entrada del algoritmo (definido a continuación).
 - *coefA*: matriz donde se almacenarán los valores de los coeficientes de los filtros adaptativos.

- Archivo almacenado en el subdirectorio *Algoritmos\setup* dentro del directorio *Simulador*: en este archivo se deberán definir los parámetros de entrada que intervienen en el algoritmo, así como el número de los mismos, y una cadena descriptiva del algoritmo, que aparecerá cada vez que dicho algoritmo sea seleccionado para realizar simulaciones. Esta cadena de texto se almacenará en una variable llamada *CADENAAYUDA*. El número de parámetros de entrada se deberá escribir en la variable *NUMPARENTRAG* y los parámetros han de definirse en las variables dispuestas en el archivo plantilla correspondiente (tres por parámetro: uno para describirlo, otro para definir el nombre de la variable que se usará en el código y otro con su valor por defecto). Este archivo se ejecuta al seleccionar '*Opciones de simulación del algoritmo...*' dentro del menú '*Simulación*'. Al seleccionar dicha opción se abrirá una ventana como la de la figura D.11 donde se podrán variar los parámetros de entrada del algoritmo en cuestión.

Parámetro	Valor
Número de coeficientes para los filtros adaptativos	50
mu	0.001
Orden de la proyección Alin	2
Factor de regulación	0.1

Figura D.11: Cuadro para la introducción de los parámetros del algoritmo a simular.

Apéndice E

Publicaciones relacionadas con la tesis

Revistas Internacionales:

- A. Gonzalez, M. Ferrer, M. de Diego, G. Piñero, J.J. López, “Practical implementation of multichannel adaptive filters based on FTF and FAP algorithms for active noise control,” *International Journal of Adaptive Control and Signal Processing*, vol 19, No. 2, pp 89-105, 2005.
- M. Ferrer, A. Gonzalez, M. de Diego, G. Piñero, “Fast efficient coefficient update for filtered-x affine projection algorithms,” *Journal of Multimedia (JMM)*, vol 2, No. 2, pp 83-89, Abril 2007.
- M. Ferrer, A. Gonzalez, M. de Diego, G. Piñero, “Fast affine projection algorithms for filtered-x multichannel active noise control,” *IEEE Transactions on Audio, Speech and Language Processing*, vol 16, No 8, pp 1396-1408, Noviembre 2008.

Congresos Internacionales:

- M. Ferrer, A. Gonzalez, M. de Diego, G. Piñero, “Steady-state performance of multichannel affine projection algorithms for active noise control,” *16th European Signal Processing Conference*, Laussane (Suiza), Agosto 2008.
- M. Ferrer, A. Gonzalez, M. de Diego, G. Piñero, “Mean square analysis of a fast filtered-x affine projection algorithm,” *IEEE International*

Conference on Acoustics, Speech, and Signal Processing (ICASSP2008), Las Vegas, Nevada, (USA), Abril 2008.

- A. González, M. Ferrer, M. de Diego, G. Piñero, “Fast filtered-x affine projection algorithm with efficient computation of coefficient update,” *2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2006)*, Toulouse (France), Mayo 2006.
- A. Gonzalez, M. de Diego, M. Ferrer, G. Piñero, “Multichannel adaptive affine projection algorithms for local sound control,” *Second International Symposium on Control, Communications and Signals Processing (ISCCSP2006)*, Marrakech (Morocco), Marzo 2006.
- A. Gonzalez, M. Ferrer, M. de Diego, G. Piñero, “Fast filtered-x affine projection algorithm for active noise control,” *IEEE Workshop on Application of Signal Processing to Audio and Acoustics WASPAA 05*, New Paltz, New York (EEUU), Octubre 2005.
- M. Ferrer, A. Gonzalez, M. de Diego, P. Faus, “Multichannel affine projection algorithm for active noise control,” *Twelfth International Congress on Sound and Vibration*, Lisboa (Portugal), Julio 2005.
- A. Gonzalez, M. de Diego, M. Ferrer, G. Piñero, “Efficiente error vector calculation in affine projection algorithms for active noise control,” *118 Audio Engineering Society Convention*, Barcelona (España), Mayo 2005.
- A. Gonzalez, M. Ferrer, M. de Diego, L. Fuster, “Efficient implementation of matrix recursions in the multichannel affine projection algorithm for multichannel sound,” *MMSP 2004 IEEE International Workshop on Multimedia Signal Processing*, Siena (Italia), Sept 2004.
- M. Ferrer, M. de Diego, A. Gonzalez, G. Piñero, “Modified affine projection algorithm for active noise control,” *Eleventh International Congress on Sound and Vibration*, Sant Petersburg (Russia), Julio 2004.
- M. Ferrer, M. de Diego, A. Gonzalez, G. Piñero, “Efficient implementation of the affine projection algorithms for Active Noise Control application,” *12th European Signal Processing Conference*, Viena (Austria), Septiembre 2004.
- M. de Diego, A. Gonzalez, M. Ferrer, G. Piñero, “An adaptive algorithms comparison for real multichannel active noise control,” *12th European Signal Processing Conference*, Viena (Austria), Septiembre 2004.

- M. de Diego, A. Gonzalez, M. Ferrer, G. Piñero, “Computationally Efficient version of the affine projection algorithm for multichannel active noise control,” *Third IEEE Sensor Array and Multichannel Signal Processing Workshop*, Barcelona (España), Julio 2004.
- M. Ferrer, A. Gonzalez, P. Zuccarello, A. Camacho “On the practical implementation of multichannel adaptive filters based on LMS, RLS, FTF and FAP algorithms for active control,” *Tenth International Congress on Sound and Vibration* Stockholm (Suecia), Julio 2003.

Congresos Nacionales:

- A. Gonzalez, M. Ferrer, P. Zuccarello, M. de Diego, “Implementación de algoritmos de proyección afín multicanal para control activo de ruido,” *XVIII Simposium Nacional URSI 2003*, La Coruña, Septiembre 2003

Índice de Figuras

2.1. Actuación de los métodos pasivos para el control de ruido (a) principio de reflexión, (b) principio de dispersión.	7
2.2. Principio en el que se basa el control activo de ruido.	7
2.3. Componentes de un sistema de control activo local de ruido monocanal <i>feedforward</i>	11
2.4. Sistema adaptativo típico.	16
2.5. Estructura de filtro transversal (FIR).	17
2.6. Superficie de error.	22
2.7. Sistema de control activo de ruido.	39
2.8. Diagrama de bloques del sistema de control activo de ruido. . .	40
2.9. Post-ecualización por el camino inverso.	43
2.10. Pre-ecualización por el camino inverso.	44
2.11. Solución empleada para contrarrestar el efecto del camino secundario haciendo uso de la estructura de filtrado-x convencional.	45
2.12. Aproximación intercambiando sistemas lineales invariantes. . .	46
2.13. Estructura de filtrado-x convencional para control activo de ruido.	47
2.14. Esquema de la estructura de filtrado-x modificada para aplicaciones de CAR.	48
2.15. Esquema de la estructura adjunta para aplicaciones de CAR. . .	50
2.16. Esquema de la estructura de filtrado-x con ecualización para CAR.	52
4.1. Prototipo experimental para un sistema CAR multicanal	88
4.2. Curvas de convergencia para el algoritmo de proyección afín de orden $N = 5$, con ruido aleatorio. Sensor de error 1 (a) y 2 (b).	90

4.3. Curvas de convergencia para el algoritmo de proyección afín de orden $N = 10$, con ruido aleatorio. Sensor de error 1 (a) y 2 (b).	91
5.1. Curvas del MSE empleando el modelo (5.47) (a) para distintos órdenes de proyección (línea continua) comparada con los valores experimentales (línea discontinua), (b) para diferentes órdenes de proyección comparadas entre sí.	103
5.2. Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.	107
5.3. Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.	107
5.4. Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.	108
5.5. Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.	108
5.6. Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.	112
5.7. Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.	112
5.8. Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.	113
5.9. Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.	113
5.10. Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.	117
5.11. Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.	117
5.12. Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.	118
5.13. Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.	118
5.14. Comparación entre el MSE para $N = 2$ obtenido mediante diferentes modelos y simulación.	121
5.15. Comparación entre el MSE para $N = 4$ obtenido mediante diferentes modelos y simulación.	121
5.16. Comparación entre el MSE para $N = 6$ obtenido mediante diferentes modelos y simulación.	122

5.17. Comparación entre el MSE para $N = 8$ obtenido mediante diferentes modelos y simulación.	122
5.18. Curvas del MSE para diferentes órdenes de proyección, según el modelo (5.73)).	124
5.19. Curvas del MSE para diferentes algoritmos considerando modelos con la aproximación $\mu \approx 0$	125
5.20. Curvas del MSE para diferentes algoritmos considerando $N = 2$	126
5.21. Curvas del MSE para diferentes algoritmos considerando $N = 8$	126
5.22. Comparación entre el MSE multicanal para $N = 2$ obtenido mediante modelo y simulación.	140
5.23. Comparación entre el MSE multicanal para $N = 3$ obtenido mediante modelo y simulación.	140
5.24. Comparación entre el MSE multicanal para $N = 4$ obtenido mediante modelo y simulación.	141
5.25. Comparación entre el MSE multicanal para $N = 5$ obtenido mediante modelo y simulación.	141
5.26. Comparación entre el MSE multicanal con estructura de filtrado- x para $N = 2$ obtenido mediante modelo y simulación.	146
5.27. Comparación entre el MSE multicanal con estructura de filtrado- x para $N = 3$ obtenido mediante modelo y simulación.	146
5.28. Comparación entre el MSE multicanal con estructura de filtrado- x para $N = 4$ obtenido mediante modelo y simulación.	147
5.29. Comparación entre el MSE multicanal con estructura de filtrado- x para $N = 5$ obtenido mediante modelo y simulación.	147
6.1. Niveles de atenuación en sensor 1 (a), y sensor 2 (b), cuando comparamos los diferentes algoritmos bajo el mismo régimen de funcionamiento empleando ruido blanco como señal a cancelar.	168
6.2. Niveles de atenuación en sensor 1 (a), y sensor 2 (b), cuando comparamos los diferentes algoritmos bajo el mismo régimen de funcionamiento empleando ruido blanco como señal a cancelar.	169
6.3. Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia empleando ruido blanco como señal a cancelar.	170
6.4. Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia, y ruido coloreado como señal a cancelar.	173

6.5. Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia, y un tono ($f = 0,1$) como señal de ruido.	174
6.6. Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia, y señal periódica como señal a cancelar.	175
6.7. Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia y caminos secundarios con retardo adicional de 50 muestras, empleando ruido blanco como señal a cancelar.	177
6.8. Atenuación en sensor 1 (a), y en sensor 2 (b) con los algoritmos funcionando a su máxima velocidad de convergencia y caminos secundarios con retardo adicional de 150 muestras, empleando ruido blanco como señal a cancelar.	178
7.1. Esquema del sistema de CAR experimental.	180
7.2. Imagen de la sala con el maniquí y el sistema 1:2:2 de CAR y detalle de la cabeza del maniquí y uno de los sensores de monitorización.	182
7.3. Esquema del controlador del sistema experimental de CAR.	183
7.4. Características del camino acústico c11 entre el sensor ES1 y el actuador SS1 (a) Respuesta al impulso, (b) Respuesta en frecuencia.	186
7.5. Características del camino acústico c12 entre el sensor ES2 y el actuador SS1 (a) Respuesta al impulso, (b) Respuesta en frecuencia.	187
7.6. Características del camino acústico c21 entre el sensor ES1 y el actuador SS2 (a) Respuesta al impulso, (b) Respuesta en frecuencia.	188
7.7. Características del camino acústico c22 entre el sensor ES2 y el actuador SS2 (a) Respuesta al impulso, (b) Respuesta en frecuencia.	189
7.8. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	191

7.9. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	192
7.10. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	193
7.11. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares $N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	194
7.12. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	195
7.13. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	196
7.14. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	197
7.15. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares $N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	198
7.16. Señal captada en el sensor de error ES1 con el algoritmo FXLMS normalizado. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia. . .	199
7.17. Señal captada en el sensor de error ES2 con el algoritmo FXLMS normalizado. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia. . .	200
7.18. Espectros comparativos de las señales en los sensores de error usando ruido blanco (a) ES1 usando coeficientes auxiliares en los algoritmos de proyección afín, (b) ES2 sin usar coeficientes auxiliares en los algoritmos de proyección afín.	201

7.19. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	203
7.20. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	204
7.21. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	205
7.22. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	206
7.23. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	207
7.24. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	208
7.25. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	209
7.26. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	210
7.27. Señal captada en el sensor de error ES1 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	211
7.28. Señal captada en el sensor de error ES2 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	212

7.29. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	214
7.30. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 3$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	215
7.31. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	216
7.32. Señal captada en el sensor de error ES2 con el algoritmo de proyección sñín (usando coeficientes auxiliares ($N = 3$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	217
7.33. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	218
7.34. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín ($N = 2$). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	219
7.35. Señal captada en el sensor de error ES1 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	220
7.36. Señal captada en el sensor de error ES2 con el algoritmo de proyección afín (usando coeficientes auxiliares ($N = 2$)). En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	221
7.37. Señal captada en el sensor de error ES1 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	222
7.38. Señal captada en el sensor de error ES2 con el algoritmo FXLMS. En verde con control, y en azul sin control. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	223

7.39. Transitorio del nivel de presión sonora (en dB_{SPL}) aplicando los diferentes algoritmos para una señal de ruido periódico. . . .	224
7.40. Señal captada en el oído izquierdo (MS_I) del maniquí usando el algoritmo de proyección afín ($N = 2$) cuando la señal a cancelar es un tono variante con la frecuencia. (a) Evolución en el tiempo. (b) Espectrograma.	226
7.41. Señal captada en el oído derecho (MS_D) del maniquí usando el algoritmo de proyección afín ($N = 2$) cuando la señal a cancelar es un tono variante con la frecuencia. (a) Evolución en el tiempo. (b) Espectrograma.	227
7.42. Señal captada en el oído izquierdo maniquí usando el algoritmo de proyección afín ($N = 2$) cuando la señal a cancelar es un tono al que se le añade ruido incorrelado. (a) Evolución en el tiempo. (b) Densidad espectral de potencia.	228
A.1. Curvas de la evolución del transitorio para (a) $\mu = 0.07$ y diferentes órdenes de proyección, (b) $N = 6$ y diferentes valores de μ . En trazo continuo la curva obtenida mediante simulaciones y en trazo discontinuo la obtenida mediante el modelo matemático.	243
D.1. Menú principal y submenú de la opción <i>Simulador</i>	288
D.2. Cuadro para modificar las opciones del simulador.	289
D.3. Cuadro inicial para la definición de un sistema acústico.	291
D.4. Cuadro para definir o modificar un sistema acústico.	292
D.5. Cuadro para estudiar el sistema acústico.	293
D.6. Cuadro para definir las señales de perturbación.	294
D.7. Menú desplegable para la elección del algoritmo a simular. . . .	295
D.8. Cuadro de diálogo para realizar la simulación.	297
D.9. Cuadro de diálogo para estudiar los resultados en los sensores de error.	298
D.10. Cuadro de diálogo para estudiar los resultados en los filtros adaptativos.	300
D.11. Cuadro para la introducción de los parámetros del algoritmo a simular.	302

Índice de Tablas

4.1. Número de multiplicaciones por iteración para diferentes algoritmos de proyección afín con estructura modificada que emplean estrategias eficientes. Ejemplo para $L = 50$, $M = 250$, $N = 5$, $I = 1$, $J = 2$ y $K = 2$	84
4.2. Número de multiplicaciones por iteración para diferentes algoritmos de proyección afín con estructura de filtrado-x convencional que emplean estrategias eficientes. Ejemplo para $L = 50$, $M = 250$, $N = 5$, $I = 1$, $J = 2$ y $K = 2$	87
7.1. Primeras frecuencias de resonancia	184
7.2. Número de ciclos por iteración y frecuencia máxima posible de la señal a cancelar para los algoritmos de proyección afín en un sistema multicanal de control activo de ruido 1:2:2 usando diferentes órdenes de proyección.	230

Bibliografía

- [1] S. Kuo and D. Morgan, *Active Noise Control Systems*. John Wiley-Sons, 1996. [2.1](#)
- [2] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice-Hall, Ed., Upper Saddle River, NJ, 2002. [2.1](#), [2.2.3](#), [2.3](#), [2.5.2](#), [2.5.2](#), [3.1](#), [3.2](#), [3.2](#), [3.3](#)
- [3] M. Recuero, *Acondicionamiento Acústico*. Cengage Learning Editores, 2001. [2.2.2](#)
- [4] B. Widrow, J. Glover, J. McCool, J. Kaunitz, C. Williams, R. Hearn, J. R. Zeidler, E. Dong, and R. Goodlin, “Adaptive noise cancelling: principles and applications,” *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Diciembre 1975. [2.2.2](#)
- [5] P. Lueg, “Process of silencing sound oscillations,” *U.S. patent*, no. 2,043,416,1936, 1936. [2.2.3](#)
- [6] D. Guicking, “On the invention of active noise control,” *Journal of the Acoustical Society of America*, vol. 87, pp. 2251–2254, 1990. [2.2.3](#)
- [7] H. Olson and E. May, “Electronic sound absorber,” *Journal of the Acoustical Society of America*, vol. 25, pp. 1130–1136, 1953. [2.2.3](#)
- [8] W. Conover, “Fighting noise with noise,” *Noise Control*, vol. 2, pp. 78–82, 1956. [2.2.3](#)
- [9] K. Kido, “Reduction of noise by use of additional sound sources,” *Proceedings of Inter-Noise*, vol. 75, pp. 667–650, 1975. [2.2.3](#)
- [10] G. Chaplin and R. Smith, “The sound of silence,” *Engineering*, vol. 218, pp. 672–673, 1978. [2.2.3](#)
- [11] S. Elliott, C. Boucher, and P. A. Nelson, “The behaviour of a multiple channel active control system,” *IEEE Transactions on Signal Processing*, vol. 40, no. 5, pp. 1041–1052, 1992. [2.2.3](#)

- [12] M. de Diego, A. Gonzalez, and C. Garcia., “On the performance of a local active noise control system,” *ICASSP99, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 885–888, Marzo 1999. [2.2.3](#)
- [13] S. Kuo and Y. Yang, “Broadband adaptive noise equalizer,” *IEEE Signal Processing Letters*, vol. 3, no. 8, pp. 234–235, Agosto 1996. [2.2.3](#)
- [14] M. de Diego, A. Gonzalez, C. Garcia, and M. Ferrer, “Some practical insights in multichannel active noise control equalization,” *ICASSP 2000, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 837–840, 2000. [2.2.3](#)
- [15] S. Elliott, P. Joseph, A. Bullmore, and P. A. Nelson, “Active cancellation at a point in a pure tone diffuse sound field,” *Journal of Sound and Vibration*, vol. 120, pp. 183–189, 1988. [2.2.4](#)
- [16] H. Kallman, “Transversal filters,” *Proceedings IRE*, vol. 28, pp. 302–310, 1940. [2.3](#)
- [17] F. Itakura and S. Saito, “On the optimum quantization of feature parameters in the parcor speech synthesizer,” *IEEE Conference Speech Communication Processing*, pp. 434–437, 1972. [2.3](#)
- [18] H. Kung and C. Leiserson, “Systolic arrays (for VLSI),” *Sparse Matrix Proceeding*, pp. 256–282, 1978. [2.3](#)
- [19] B. Widrow and S. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, N.J: Prentice-Hall, 1985. [2.3](#), [2.4.2](#), [2.4.3](#), [2.4.3](#), [2.7](#), [3.1](#), [4.2](#)
- [20] S. Alexander and A. Ghirnikar, “A method for recursive least-squares filtering based upon an inverse QR decomposition,” *IEEE Transactions on Signal Processing*, vol. 35, pp. 1250–1258, 1993. [2.3](#)
- [21] J. Cioffi and T. Kailath, “Fast, recursive-least-squares transversal filters for adaptive filtering,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 4, pp. 304–337, Abril 1984. [2.3](#)
- [22] T. Kailath, “An innovation approach to least-squares estimation, part I: linear filtering in additive white noise,” *IEEE Transactions on Automatical Control*, vol. 13, pp. 646–655, 1968. [2.3](#)
- [23] N. Gopalan and A. Spanias, “Fast adaptive algorithms using eigenspace projections,” *Signal Processing*, no. 83, pp. 1923–1935, 2003. [2.4.2](#)

- [24] R. H. Kwong, and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1633–1642, Julio 1992. [2.4.3](#), [8.2](#)
- [25] K. Mayyas and T. Aboulnasr, "Leaky LMS algorithm: MSE analysis for gaussian data," *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 927–934, Abril 1997. [2.4.3](#)
- [26] A. Feuer, "Performance analysis of the block least mean square algorithm," *IEEE Transactions on Circuits and System*, vol. 32, no. 9, pp. 960–963, Agosto 1985. [2.4.3](#)
- [27] G. Carayannis, D. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, pp. 1394–1402, 1983. [2.5.1](#), [B.1](#), [B.3](#)
- [28] R. Plackett, "Some theorems in least squares," *Biometrika*, vol. 37, p. 149, 1950. [2.5.2](#)
- [29] R. Hastings-James and M. Sage, "Recursive generalized-least-squares procedure for on line identification of process parameters," *Proceedings IEEE (London)*, vol. 116, pp. 2057–2062, 1969. [2.5.2](#)
- [30] N. Hubing and S. Alexander, "Statistical analysis of incialization methods for RLS adaptive filters," *IEEE Transactions on Signal Processing*, vol. 39, no. 8, Agosto 1991. [2.5.2](#)
- [31] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions ASME Journal Basic Engineering*, vol. 83, pp. 95–107, 1960. [2.6](#)
- [32] R. Kalman and R. Bucy, "New result in linear filtering and prediction theory," *Transactions ASME Journal Basic Engineering*, vol. 83, pp. 95–107, 1960. [2.6](#)
- [33] A. Sayed and T. Kailath, "A state space approach to adaptive RLS filtering," *IEEE Processing Magazine*, vol. 11, no. 3, pp. 18–60, 1994. [2.6](#)
- [34] R. Fraanje, A. Sayed, M. Verhaegen, and N. Doelman, "A fast array Kalman filter solution to active noise control," *International Journal of Adaptive Control and Signal Processing*, vol. 19, pp. 125–152, 2005. [2.6](#)
- [35] B. Widrow, D. Shur, and S. Shaffer, "On adaptive inverse control," *Proceedings of the 15th ASILOMAR Conference of Circuits, Systems and Computers*, pp. 185–195, 1981. [2.7.2](#), [2.7.2](#)

- [36] J. Burguess, "Active adaptive sound control in a duct: a computer simulation," *Journal of the Acoustical Society of America*, vol. 70, pp. 715–726, Septiembre 1981. [2.7.2](#)
- [37] D. Morgan and C. Sanford, "A control theory approach to the stability and transient analysis of the filtered-x LMS adaptive notch filter," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2341–2346, Septiembre 1992. [2.7.2](#)
- [38] E. Bjarnason, "Analysis of the filtered-x LMS algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 6, pp. 504–514, 1995. [2.7.2](#)
- [39] S. J. Elliott and P. A. Nelson, "Multiple-point equalization in a room using adaptive digital filters," *Journal of the Audio Engineering Society*, vol. 37, no. 11, pp. 899–907, 1989.
- [40] G. Long, F. Ling, and J. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. 37, no. 9, pp. 1397–1405, 1989.
- [41] G. Long, "Corrections to the LMS algorithm with delayed coefficient adaptation," *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. 40, no. 1, pp. 230–232, 1992. [2.7.2](#)
- [42] M. Rupp and A. Sayed, "Robust FXLMS algorithms with improved convergence performance," *IEEE Transaction on Speech and Audio Processing*, vol. 6, no. 1, pp. 78–85, 1998. [2.7.2](#)
- [43] E. Bjarnason, "Active noise cancellation using a modified form of the filtered-x LMS algorithm," *Proceedings of the 6th European Signal Processing Conference*, vol. 2, pp. 1053–1056, 1992. [2.7.3](#), [4.1](#)
- [44] S. Douglas, "Fast exact filtered-x LMS and LMS algorithms for multichannel active noise control," *ICASSP97, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 399–402, 1997. [2.7.3](#)
- [45] S. Douglas, "An efficient implementation of the modified filtered-x LMS algorithm," *IEEE Signal Processing Letters*, vol. 4, pp. 286–288, Octubre 1997. [2.7.3](#)
- [46] S. Douglas, "Fast implementation of the filtered-x LMS and LMS algorithms for multichannel active noise control," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 4, pp. 454–464, Julio 1999. [2.7.3](#)

- [47] E. A. Wan, “Adjoint-LMS: an efficient alternative to the filtered-x LMS and multiple error LMS algorithms,” *ICASSP96, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1361–1368, Mayo 1996, Atlanta, GA. [2.7.4](#)
- [48] S. M. Kuo and J. Tsai, “Residual noise shaping technique for active noise control systems,” *Journal of the Acoustical Society of America*, vol. 95, pp. 1665–1668, Marzo 1994. [2.7.4](#)
- [49] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Proceedings of the Electronics and Communications in Japan*, vol. J67-A, pp. 126–132, Febrero 1984. [3.1](#), [4.1](#)
- [50] S. Douglas, “A family of normalized LMS algorithms,” *IEEE Signal Processing Letters*, vol. 1, no. 3, pp. 49–51, Marzo 1994. [3.1](#)
- [51] J. Nagumo and A. Noda, “A learning method for system identification,” *IEEE Transactions on Automatic Control*, vol. 12, no. 3, pp. 282–287, Junio 1967. [3.1](#)
- [52] Y. Choi and H. Shin, “Robust regularization for normalized LMS algorithms,” *IEEE Transactions on Circuits and Systems*, vol. 53, no. 8, pp. 627–631, 2006. [3.3](#)
- [53] A. Carini and G. Sicuranza, “Optimal regularization parameter of the multichannel filtered-x affine projection algorithm,” *IEEE Transactions on Signal Processing*, vol. 55, no. 2, 2007. [3.3](#)
- [54] K. Chen, J. Lu, and B. Xu, “A method to adjust regularization parameter of fast affine projection algorithm,” *The 8th International Conference on Signal Processing*, vol. 1, pp. 16–20, 2006. [3.3](#)
- [55] Y. Choi, H. Shin, and W. Song, “Adaptive regularization matrix for affine projection algorithm,” *IEEE Transactions on Circuits and Systems*, vol. 54, no. 12, pp. 1087–1091, Diciembre 2007. [3.3](#)
- [56] G. Rombouts and M. Moonen, “A sparse block exact affine projection algorithm,” *IEEE Transaction on Speech and Audio Processing*, vol. 10, no. 2, pp. 100–108, 2002. [3.3](#)
- [57] S. G. Sankaran and A. L. Beex, “Normalized LMS algorithm with orthogonal correction factors,” *Proceeding 31 Asilomar Conference*, pp. 1670–1673, Noviembre 1997. [3.3](#)

- [58] S. G. Kratzer and D. R. Morgan, “The partial rank algorithm for adaptive beamforming,” *Proceeding SPIE*, vol. 564, pp. 9–14, 1985. [3.3](#)
- [59] S. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*, 2nd ed. Kubler Academic Publishers, 2001. [3.3](#), [3.5](#), [3.5.2](#), [4.1](#)
- [60] S. Gay, “Fast projection algorithms with application to voice excited echo cancellers,” Ph.D. dissertation, Rutgers University, New Jersey 1997. [3.4](#), [3.5](#), [4.1](#)
- [61] M. Ferrer, M. de Diego, A. Gonzalez, and G. Piñero, “Efficient implementation of the affine projection algorithms for active noise control application,” *Proceeding of the 12th European Signal Processing Conference*, Septiembre 2004. [3.4.1](#), [4.2.1](#)
- [62] S. Gay and S. Tavathia, “The fast affine projection algorithm,” *ICASSP95, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 3023–3026, Mayo 1995, Detroit, MI. [3.5](#), [8.1](#)
- [63] S. Douglas, “The fast affine projection algorithm for active noise control,” *Proceedings of the 29th Asilomar Conference on Signals, Systems and Components*, vol. 2, pp. 1245–1249, Octubre 1995, Pacific Grove, CA. [3.5](#), [4.1](#), [4.2](#), [8.1](#)
- [64] M. Bouchard, “Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 1, pp. 54–60, Enero 2003. [3.5](#), [3.5.1](#), [3.5.3](#), [4.1](#), [4.2](#)
- [65] S. Douglas, “Efficient approximate implementations of the fast affine projection algorithm using orthogonal transforms,” *ICASSP96, Proceedings International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1656–1659, Mayo 1996, Atlanta, USA. [3.5](#)
- [66] M. Bouchard and F. Albu, “The multichannel gauss-seidel fast affine projection algorithm for active noise control,” *Proceedings of the 7th International Symposium on Signal Processing and its Applications (ISSPA)*, pp. 1–4, Julio 2003, Paris, Francia. [3.5.1](#)
- [67] A. Gonzalez, M. Ferrer, M. de Diego, and G. Piñero, “Fast filtered-x affine projection algorithm with efficient computation of coefficient update,” *ICASSP06, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 285–288, 2006, Toulouse, Francia. [3.5.2](#)

- [68] P. A. Nelson and S. J. Elliott, *Active Control of Sound*. Academic Press, 1992. [4.1](#)
- [69] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*. Wiley-IEEE, 2004. [4.1](#)
- [70] E. Bjarnason, “Analysis of the filtered-x LMS algorithm,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 6, pp. 504–514, 1985. [4.1](#)
- [71] S. Elliott, *Signal Processing for Active Control*. Academic Press, 2001. [4.1](#)
- [72] F. Albu and M. Bouchard, “A low cost and fast convergence gauss-seidel pseudo fast affine projection algorithm for multichannel active noise control,” *ICASSP04, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 121–124, 2004, Montreal, Canada. [4.1](#)
- [73] Y. Zakharov and F. Albu, “Coordinate descent iterations in fast affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 12, no. 5, pp. 353–356, 2005.
- [74] F. Albu, M. Bouchard, and Y. Zakharov, “Pseudo-affine projection algorithms for multichannel active noise control,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1044–1052, 2007.
- [75] M. Bouchard and H. Ding, “An exact relaxed fast affine projection algorithm for multichannel active noise control,” *Proceeding of the 15th International Conference on Digital Signal Processing (DSP)*, pp. 47–50, 2007, Cardiff, Gales. [4.1](#)
- [76] A. Gonzalez, M. Ferrer, M. de Diego, and G. Piñero, “Fast filtered-x affine projection algorithm for active noise control,” *Proceedings of IEEE Workshop on Application of Signal Processing to Audio and Acoustics WASPAA 05*, pp. 162–165, 2005, New York, USA. [4.1](#)
- [77] A. Gonzalez, M. de Diego, M. Ferrer, and G. Piñero, “Efficient error vector calculation in affine projection algorithms for active noise control,” *Proceedings of the 118th AES Conference*, Mayo 2005, Barcelona, España. [4.1](#), [4.3](#)
- [78] M. Bouchard and S. Quednau, “Multichannel recursive-least-square algorithms and fast-transversal-filter algorithms for active noise control

- and sound reproduction systems,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 606–618, 2000. [4.2](#)
- [79] H. Shin and A. Sayed, “Mean-square performance of a family of affine projection algorithms,” *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 90–102, Enero 2004. [5.1](#), [5.2](#), [5.2](#), [5.2](#), [8.2](#)
- [80] T. Y. Al-Naffouri and A. H. Sayed, “Transient analysis of data-normalized adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 639–650, Marzo 2003. [5.2](#), [A.2](#)
- [81] A. H. Sayed, *Fundamentals of Adaptive Filtering*, 1st ed. John Wiley-sons, 2003. [5.2](#)
- [82] C. W. Lee, H. Cho, S. J. Ban, and S. W. Kim, “A derivation of excess mean square error for affine projection algorithms using the condition number,” *International Conference on Control, Automation and System 2007*, Octubre 2007, Seoul, Korea. [5.2](#)
- [83] A. Carini and G. Sicuranza, “Transient and steady-state analysis of filtered-x affine projection algorithm,” *IEEE Transactions on Signal Processing*, vol. 54, no. 2, 2006. [5.5](#), [8.2](#)
- [84] A. Gonzalez, “Cancelación activa de ruido mediante técnicas de procesamiento digital adaptativa,” Tesis doctoral, Universidad Politécnica de Valencia, 1997. [5.5](#)
- [85] “TMS320C6713 DSK Technical Reference.” [7.2](#)
- [86] “TMS320C6713 Floating Point Digital Signal Processor.” [7.2](#)
- [87] “ADK-6713 Hardware Reference Manual.” [7.2](#)
- [88] T. Instrument, “Code Composer Studio. Getting Started Guide,” Noviembre 2001. [7.4](#)
- [89] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, “Fast projection algorithm and its step size control,” *ICASSP95, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 945–948, Mayo 1995, Detroit, MI. [8.1](#)
- [90] S. J. Elliott, I. M. Stothers, and P. A. Nelson, “A multiple error LMS algorithm and its application to the active control of sound and vibration,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 35, no. 10, pp. 1423–1434, Octubre 1987. [8.1](#), [8.2](#)

- [91] H. Hamada, "Signal processing for active noise control: adaptive signal processing," *Proceeding International Symposium on Active Control of Sound and Vibration*, vol. 1, no. 33-44, Tokio 1991. 8.2
- [92] A. Gonzalez, "Stochastic gradient algorithms in active control," *Proceedings of the European Signal Processing Conference*, vol. IV, no. 2333-2336, Septiembre 1998, Isla de Roda, Grecia. 8.2
- [93] H. Shin and A. Sayed, "Variable step-size NLMS and affine projection algorithms," *IEEE Transactions Signal Processing Letters*, vol. 11, no. 2, pp. 132–135, Febrero 2004. 8.2
- [94] T. Dai, B. Shahrrava, and X. Chen, "A variable step-size affine projection algorithm with a weighted projection matrix," *Canadian Conference on Electrical and Computer Engineering*, pp. 320–323, 2005. 8.2
- [95] K. Mayyas, "An affine projection algorithm with an adaptive step-size equation," *2007 IEEE International Symposium on Signal Processing and Information Technology*, pp. 554–557, Diciembre 2007. 8.2
- [96] J. Lee, Y. Park, and D. H. Youn, "Robust pseudo affine projection algorithm with variable step-size," *Electronics-Letters*, vol. 44, no. 3, pp. 250–251, Enero 2008. 8.2
- [97] N. J. Bershad, J. C. Bermudez, and J. Tournernet, "An affine combination of two LMS adaptive filters - transient mean square analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1853–1864, Mayo 2008. 8.2
- [98] L. Azpicueta-Ruiz, A. Figueiras-Vidal, and J. Arenas-Garcia, "A normalized adaptation scheme for the convex combination of two adaptive filters," *ICASSP08, Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 3301–3304, 2008, Las Vegas, USA. 8.2
- [99] N. Levinson, "The wiener RMS error criterion in filter design and prediction," *Journal Mathematical Physical*, vol. 25, pp. 261–278, 1947. C.1
- [100] P. M. Clarkson, *Optimal and Adaptive Signal Processing*. CRC Press, 1993. C.2