

01130
22

**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERIA



**ESTUDIO DE LA ROBUSTEZ NUMERICA EN
ALGORITMOS DE FILTRADO ADAPTABLE PARA
LA IGUALACION DE CANAL EN COMUNICACIONES
DIGITALES**

T E S I S

QUE PARA OBTENER EL TITULO DE :

INGENIERO EN TELECOMUNICACIONES

P R E S E N T A N :

PEDRO GIOVANNI LEON NAJERA

JUAN ALBERTO MONTER SANABRIA

DIRECTOR DE TESIS :
DR. ROGELIO ALCANTARA SILVA

MEXICO, D.F.

JULIO 2003

A

TESIS CON
FALLA DE ORIGEN





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mis padres José Luis y María Elena por el apoyo incondicional que siempre me han prestado.

A mi hermano José Luis cuya amistad, apoyo y experiencia me han conducido por un camino lleno de satisfacciones.

En recuerdo de mi abuela María de la Paz.

A todos mis amigos por su apoyo en momentos difíciles y por compartir los buenos y malos momentos.

Juan Alberto

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo intelectual.

NOMBRE: PEDRO GIOVANNI
LEÓN NAJERA
FECHA: 4 JUL 103
FIRMA: [Firma]

TESIS CON
FALLA DE ORIGEN

B

A mi papá, Mario León

Por su ejemplo de dedicación, respeto, templanza y por su apoyo incondicional en cada uno de mis retos.

A mi mamá, Ma. Antonia Nájera

Por su comprensión, respeto y por todas sus palabras de motivación, así mismo por todo su apoyo sin el cual no hubiera logrado terminar éste ni ninguno de mis proyectos.

A mis hermanos, Bella y Yonathan

Por su comprensión, respeto y compañía.

A mis amigos...

Alejandro, Ivan, Sergio, Jose Luis, Juan, Jose Manuel, Nadir, Alonso, Carina, Eduardo, Carlos, por todos los momentos agradables que hemos pasado.

Pedro Giovanni

C

TESIS CON
FALLA DE ORIGEN

Agradecimientos

Agradecemos a Dios por la vida que nos concede.

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por habernos dado la oportunidad de estudiar en sus aulas.

Agradecemos de la manera más cordial y atenta a todos aquellos profesores que han contribuido de una manera positiva en nuestra formación profesional, especialmente al Dr. Miguel Moctezuma Flores, Dr. Francisco García Ugalde, M.I. Larry H. Escobar Salguero y á la Ing. Gloria Mata Hernández por haber aceptado ser parte de nuestro jurado y por sus valiosos comentarios en la revisión final de este trabajo.

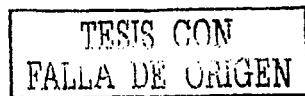
Agradecemos al M.I. Octavio Estrada Castillo por todo su apoyo brindado a lo largo de la carrera.

Finalmente, quisieramos expresar nuestro más sincero agradecimiento al Dr. Rogelio Alcántara Silva, á quien reconocemos su gran apoyo, paciencia y amistad durante nuestro desarrollo como profesionistas y en especial para la realización de este trabajo.

Juan Alberto Monter Sanabria

Pedro Giovanni León Nájera

D



RESUMEN

Una pieza fundamental en todo sistema de comunicaciones es el canal de transmisión. Cuando las señales viajan a través de éste, se distorsionan debido a problemas de ancho de banda limitado y propagación por multitrayectorias en los canales móviles, por esta razón al recibir las señales es importante compensar las distorsiones introducidas por el canal. Los filtros igualadores son una herramienta que nos permiten realizar dicha tarea. Dado que generalmente las características de los canales son variantes con el tiempo, el filtro igualador debe ajustarse a las variaciones del canal para compensar en todo tiempo las distorsiones introducidas por éste, por esta razón hablamos de un igualador adaptable.

El algoritmo de filtrado adaptable tiene la función de actualizar los coeficientes del filtro de acuerdo con un criterio de optimización dado. En particular, en este trabajo estudiamos dos de los criterios de optimización más utilizados, el del error cuadrático promedio (MSE) y el de los mínimos cuadrados (LS), resaltando las características de cada uno de ellos.

Puesto que todo algoritmo deberá implantarse en un dispositivo de procesamiento de señales, donde el número de bits para representar las cantidades es limitado, es importante garantizar que los errores introducidos debido a problemas de representación, no se incrementen a lo largo del proceso de actualización de los parámetros del filtro; puesto que si esto sucede el algoritmo puede diverger como se muestra en este trabajo.

En la primera parte de este documento, presentamos todas las herramientas necesarias para el estudio de los algoritmos de filtrado adaptable y de la problemática de la interferencia entre símbolos, en una segunda parte se estudian los problemas numéricos que presentan los algoritmos, además se muestra un método de análisis de errores por redondeo en algoritmos recursivos. Así mismo, se realiza un estudio de los errores de redondeo en los algoritmos clásicos RLS y FRLS presentando las versiones estabilizadas que garantizan el buen desempeño de los algoritmos en aritmética finita.

Finalmente se presentan un gran número de simulaciones tanto en aritmética entera como flotante, que permiten entender las causas de los problemas numéricos en los

E

TESIS CON
SELLA DE ORIGEN

algoritmos de filtrado adaptable.

Uno de los objetivos primordiales de este trabajo es demostrar que es importante considerar los problemas numéricos que muchas veces hacen diverger los algoritmos y que limitan por tanto la implementación de éstos en aplicaciones reales. Así mismo se desea presentar un material que permita a aquellos interesados en el problema de la igualación de canal introducirse en el área mediante una lectura accesible.

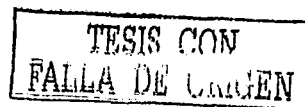
F

Índice general

| | |
|---|-----------|
| Resumen | 4 |
| Indice general | I |
| Indice de figuras | V |
| Indice de cuadros | VIII |
| Introducción | IX |
| 1. LOS CANALES DE COMUNICACIÓN Y LA INTERFERENCIA ENTRE SIMBOLOS (ISI). | 1 |
| 1.1. Introducción | 1 |
| 1.2. Tipos de canales de comunicación | 1 |
| 1.3. Descripción de los canales de propagación | 3 |
| 1.3.1. Cables eléctricos | 3 |
| 1.3.2. Fibra óptica | 4 |
| 1.3.3. Canal satelital | 5 |
| 1.4. Modelado de los canales móviles de comunicación | 6 |
| 1.4.1. El fenómeno de multitrayectorias | 6 |
| 1.4.2. El canal Gaussiano | 8 |
| 1.4.3. El canal Rayleigh con desvanecimientos | 9 |
| 1.4.4. El canal Rician | 10 |
| 1.5. Eliminación de la interferencia entre símbolos (ISI) en las comunicaciones digitales | 11 |
| 1.5.1. Diseño de señales para canales de banda limitada | 12 |
| 1.5.2. Diseño de señales de banda limitada para cero ISI | 16 |
| 1.5.3. La igualación de canal como solución al problema de ISI. | 18 |
| 1.6. Conclusiones | 23 |
| 2. TEORÍA DE FILTROS LINEALES ÓPTIMOS | 24 |
| 2.1. Introducción | 24 |
| 2.2. Modelado de procesos estocásticos | 24 |
| 2.3. Aplicaciones de los filtros digitales. | 28 |

| | | |
|-----------|--|-----------|
| 2.3.1. | Funciones de los filtros | 29 |
| 2.3.2. | Filtros adaptables | 30 |
| 2.3.3. | La predicción lineal | 31 |
| 2.3.4. | El filtrado | 32 |
| 2.4. | El filtro de Wiener | 33 |
| 2.5. | El filtro de Kalman | 37 |
| 2.6. | Conclusiones | 42 |
| 3. | ALGORITMOS DE FILTRADO ADAPTABLE PARA LA IGUALACIÓN DE CANAL | 43 |
| 3.1. | Introducción | 43 |
| 3.2. | Algoritmos basados en el error cuadrático promedio | 43 |
| 3.2.1. | El método de paso descendente | 43 |
| 3.2.2. | Algoritmo LMS transversal | 48 |
| 3.3. | Algoritmos basados en los mínimos cuadrados | 51 |
| 3.3.1. | El algoritmo RLS transversal | 52 |
| 3.3.2. | El algoritmo FRLS "Fast Kalman" | 55 |
| 3.4. | Evaluación de algoritmos | 61 |
| 3.5. | Conclusiones | 64 |
| 4. | ANÁLISIS DE ROBUSTEZ NUMÉRICA EN ALGORITMOS DE FILTRADO ADAPTABLE | 66 |
| 4.1. | Introducción | 66 |
| 4.2. | Errores Numéricos | 67 |
| 4.2.1. | Tipos de errores | 67 |
| 4.2.2. | Errores por redondeo | 67 |
| 4.2.3. | Errores en operaciones aritméticas | 68 |
| 4.2.4. | Propagación del Error | 69 |
| 4.3. | Análisis general de la propagación de errores en algoritmos recursivos | 70 |
| 4.4. | Análisis de los errores de redondeo en los algoritmos RLS | 72 |
| 4.4.1. | Análisis teórico del error | 73 |
| 4.5. | Generalidades de los algoritmos rápidos | 80 |
| 4.6. | Errores de redondeo en algoritmos rápidos (FRLS) | 82 |
| 4.6.1. | Algoritmos rápidos estabilizados | 83 |
| 4.6.2. | Redundancia y retroalimentación del error | 83 |
| 4.6.3. | Obtención de algoritmos rápidos estabilizados | 86 |
| 4.7. | Conclusiones | 92 |
| 5. | EVALUACIÓN Y COMPARACIÓN DE ALGORITMOS ROBUSTOS DE FILTRADO ADAPTABLE | 94 |
| 5.1. | Algoritmos RLS clásicos | 95 |
| 5.1.1. | Implementación CLS1 | 95 |
| 5.1.2. | Implementación CLS2 | 97 |

| | | |
|-----------|---|------------|
| 5.2. | Algoritmos FRLS con precisión doble | 101 |
| 5.2.1. | Algoritmo Rápido de Kalman | 101 |
| 5.2.2. | Algoritmo Rápido Versión Cioffi | 101 |
| 5.2.3. | Algoritmo Rápido Versión Slock | 103 |
| 5.3. | Algoritmos SFRLS con precisión doble | 104 |
| 5.3.1. | Algoritmo Estabilizado Rápido Versión Cioffi. SFRLSVC. | 105 |
| 5.3.2. | Algoritmo Estabilizado Rápido Versión Slock. SFRLSVS | 106 |
| 5.4. | Algoritmos FRLS con precisión sencilla | 106 |
| 5.4.1. | Algoritmo Rápido de Kalman | 107 |
| 5.4.2. | Algoritmo Rápido Versión Cioffi | 108 |
| 5.4.3. | Algoritmo Rápido Versión Slock | 109 |
| 5.5. | Algoritmos SFRLS con precisión sencilla | 110 |
| 5.5.1. | Algoritmo Estabilizado Rápido Versión Cioffi. SFRLSVC | 110 |
| 5.5.2. | Algoritmo Estabilizado Rápido Versión Slock. SFRLSVS | 111 |
| 5.6. | Algoritmos FRLS en aritmética entera | 112 |
| 5.6.1. | Algoritmo SFRLSVS en aritmética entera | 112 |
| 5.6.2. | Algoritmo Fast Kalman en aritmética entera | 116 |
| 5.7. | Conclusiones | 120 |
| 6. | CONCLUSIONES | 122 |
| A. | LEMA DE INVERSIÓN MATRICIAL | 125 |
| B. | SOLUCIÓN PARA LA PREDICCIÓN FORWARD | 129 |
| B.1. | Minimización del error de predicción basado en el error cuadrático promedio | 129 |
| B.2. | Minimización del error de predicción basado en el criterio de los mínimos cuadrados | 132 |
| C. | ANÁLISIS DE VALORES Y VECTORES CARACTERÍSTICOS. 137 | |
| C.0.1. | Propiedades de los valores característicos y vectores característicos | 138 |
| D. | ALGORITMOS DE FILTRADO FRLS | 146 |
| D.1. | Algoritmo FRLS Versión Cioffi | 146 |
| D.2. | Algoritmo FRLS Versión Slock | 148 |
| E. | ALGUNOS ASPECTOS DE ALGEBRA VECTORIAL | 151 |
| E.1. | Rango de una matriz | 151 |
| E.2. | Normas de matrices y números condición | 152 |
| E.2.1. | Normas de matrices | 152 |
| E.2.2. | Números condición | 154 |
| E.2.3. | Matrices positivas | 156 |
| E.2.4. | Valores singulares de una matriz | 157 |



Apéndices

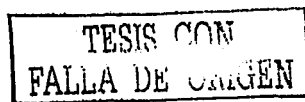
125

Bibliografía

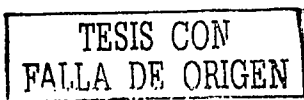
159

Índice de figuras

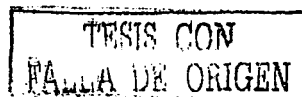
| | |
|--|----|
| 1.1. Clasificación de los canales de comunicación. | 2 |
| 1.2. El fenómeno de multitrayectoria en una zona urbana. | 7 |
| 1.3. Canal gaussiano. | 8 |
| 1.4. FDP Rayleigh | 9 |
| 1.5. FDP Rician normalizada a sus promedios locales. | 10 |
| 1.6. ISI en un sistema de comunicación binario | 12 |
| 1.7. Sistema de comunicación básico | 13 |
| 1.8. Diagrama de ojo | 15 |
| 1.9. Constelación para una transmisión QAM | 15 |
| 1.10. Espectro de $B(f)$ | 17 |
| 1.11. Esquema de un igualador adaptable | 19 |
| 2.1. Modelo de un filtro generador de señales estocásticas. | 25 |
| 2.2. Descripción del filtro generador de señales estocásticas. | 25 |
| 2.3. Estructura generadora de un proceso AR $u(k)$ | 27 |
| 2.4. Estructura generadora de un proceso MA $u(k)$ | 28 |
| 2.5. Sistema para la predicción de señales. | 31 |
| 2.6. Problema fundamental del filtrado. | 33 |
| 2.7. Filtro FIR transversal. | 35 |
| 2.8. Superficie del error | 37 |
| 2.9. Problema fundamental de un filtro Kalman. | 38 |
| 3.1. Filtro transversal adaptable. | 44 |
| 3.2. Banco de correladores para estimar la corrección en el algoritmo de paso descendente. | 47 |
| 3.3. Esquema de igualación del algoritmo LMS. | 49 |
| 3.4. Esquema de igualación de canal. | 61 |
| 3.5. Respuesta es frecuencia del algoritmo LMS. | 62 |
| 3.6. Error en el algoritmo de filtrado LMS. | 63 |
| 3.7. Error en el algoritmo de filtrado RLS. | 63 |
| 3.8. Error en el algoritmo de filtrado rápido RLS. Fast Kalman | 64 |
| 4.1. Errores en los algoritmos | 67 |
| 4.2. Retroalimentación para los algoritmos SFRLS | 84 |



| | | |
|-------|---|-----|
| 5.1. | Implementación CLS1. (a) Comportamiento del error de filtrado. $\lambda = 1, 0.96, 0.93$. (b) Comportamiento de la norma de la matriz de covarianza $R_p^{-1}(k)$ para $\lambda = 1$ | 95 |
| 5.2. | Comportamiento de la norma de la matriz de covarianza $R_p^{-1}(k)$ para la implementación CLS1. (a) $\lambda = 0.96$. (b) $\lambda = 0.93$ | 96 |
| 5.3. | Implementación CLS1. (a) Respuesta en frecuencia. (b) Convolución $h_c * h_f$ para $\lambda = 1$ | 97 |
| 5.4. | Implementación CLS1. convolución $h_c * h_f$ (a) $\lambda = 0.93$. (b) $\lambda = 0.96$ | 98 |
| 5.5. | Implementación CLS2. (a) Comportamiento del error de filtrado. (b) Comportamiento de la norma de matriz de covarianza $R_p^{-1}(k)$ para $\lambda = 1$ | 99 |
| 5.6. | Comportamiento de la norma de la matriz de covarianza $R_p^{-1}(k)$ para la implementación CLS2. (a) $\lambda = 0.96$. (b) $\lambda = 0.93$ | 99 |
| 5.7. | Implementación CLS2. (a) Respuesta en frecuencia. (b) Convolución $h_c * h_f$ para $\lambda = 1$ | 100 |
| 5.8. | Implementación CLS2. convolución $h_c * h_f$ (a) $\lambda = 0.96$. (b) $\lambda = 0.93$ | 100 |
| 5.9. | Algoritmo Rápido Fast Kalman en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 102 |
| 5.10. | Algoritmo Rápido versión Cioffi en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 103 |
| 5.11. | Algoritmo Rápido versión Slock en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 104 |
| 5.12. | Algoritmo Rápido Estabilizado versión Cioffi (algoritmo de Botto.) en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 105 |
| 5.13. | Algoritmo Rápido Estabilizado versión Slock en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 106 |
| 5.14. | Algoritmo Rápido Fast Kalman en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 107 |
| 5.15. | Algoritmo Rápido versión Cioffi en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 108 |
| 5.16. | Algoritmo Rápido versión Slock en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 109 |
| 5.17. | Algoritmo Rápido Estabilizado versión Cioffi (algoritmo de Botto.) en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 110 |



| | |
|--|-----|
| 5.18. Algoritmo Rápido Estabilizado versión Slock en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ | 111 |
| 5.19. Respuesta en frecuencia de los algoritmos rápidos estabilizados. Cioffi-Botto y Slock, en precisión sencilla. (a) $\lambda = 0.96$. (b) $\lambda = 0.93$ | 112 |
| 5.20. Comportamiento del error de filtrado del algoritmo SFRLSVS en aritmética entera. (a) $\lambda = 0.98$. (b) $\lambda = 0.95$ | 113 |
| 5.21. Comportamiento de la variable de verosimilitud en el algoritmo SFRLSVS en aritmética entera. (a) 32 bits. (b) 16 bits. | 114 |
| 5.22. Comportamiento de los errores de redondeo en el error de filtrado $e(k)$. (a) aritmética de 16 bits. (b) aritmética de 32 bits. | 115 |
| 5.23. Comportamiento de los errores de redondeo en la ganancia dual de Kalman $C_p(k)$. (a) aritmética de 16 bits. (b) aritmética de 32 bits. | 115 |
| 5.24. Respuesta en frecuencia de la implementación SFRLSVS en aritmética entera (a) $\lambda = 0.98$. (b) $\lambda = 0.95$ | 116 |
| 5.25. Comportamiento del error de filtrado en el algoritmo "Fast Kalman". (a) aritmética de 16 bits ($\lambda = 0.98$). (b) aritmética de 32 bits | 117 |
| 5.26. Comportamiento de los errores de redondeo en el error de filtrado $e(k)$ algoritmo Fast Kalman. (a) aritmética de 16 bits ($\lambda = 0.98$). (b) aritmética de 32 bits. | 118 |
| 5.27. Respuesta en frecuencia de la implementación "Fast Kalman" en aritmética entera (a) $\lambda = 0.98$ (b) $\lambda = 1$ | 119 |



Índice de cuadros

| | |
|---|-----|
| 3.1. Algoritmo de Filtrado LMS | 51 |
| 3.2. Algoritmo de filtrado RLS | 56 |
| 3.3. Algoritmo FRLS "Fast Kalman". | 60 |
| 4.1. Algoritmos RLS | 73 |
| 4.2. Variables sensibles a errores de redondeo y utilizadas para la estabi- lización de los algoritmos FRLS. | 85 |
| 4.3. Algoritmo SFRLS Versión Cioffi | 87 |
| 4.4. Continuación algoritmo SFRLS Versión Cioffi | 88 |
| 4.5. Algoritmo SFRLS Versión Slock. | 90 |
| 4.6. Continuación algoritmo SFRLS Versión Slock | 91 |
| B.1. Algoritmo de Levinson-Durbin | 132 |
| B.2. Algoritmo predictor RLS | 136 |
| D.1. Algoritmo FRLS Versión Cioffi | 147 |
| D.2. Algoritmo FRLS Versión Slock | 149 |
| D.3. Continuación algoritmo FRLS Versión Slock | 150 |

INTRODUCCIÓN

En todo sistema de comunicaciones, el ancho de banda de transmisión es limitado, por esta razón las señales recibidas difieren de las señales transmitidas, en particular: un pulso digital requiere de un ancho de banda infinito para ser transmitido sin ser distorsionado. Dado que el ancho de banda es limitado este pulso se expande en el tiempo, lo que ocasiona que se interfiera con otros símbolos enviados en subsecuentes instantes de tiempo que a su vez también son distorsionados.

El problema entonces consiste en recuperar la señal transmitida sin distorsión y para tal efecto se utilizan igualadores de canal con una respuesta en frecuencia que complemente la respuesta en frecuencia del canal, es decir: que amplifique frecuencias donde el canal atenúa la señal con el objetivo de simular un ancho de banda suficiente para evitar la interferencia entre símbolos. El canal de transmisión en general no mantiene una respuesta en frecuencia fija durante toda la transmisión puesto que este se ve influenciado por efectos de clima y movimiento del receptor principalmente (aunque no son las únicas razones), por consecuencia el igualador de canal debe ajustarse para seguir compensando la respuesta del canal de transmisión y por tanto se le conoce como igualador o filtro adaptable. La rapidez y exactitud con que los parámetros del filtro adaptable son calculados dependen tanto del criterio de optimización que se utilice (máxima verosimilitud, mínimos cuadrados, error cuadrático promedio, etc) como de la estructura del filtro.

En esta tesis se estudian básicamente dos de los criterios más utilizados (mínimos cuadrados y error cuadrático promedio) para la solución de este problema, presentándose las ventajas de cada uno de ellos. En particular, se estudian los filtros de respuesta impulsional finita (FIR) con estructura transversal. El hecho de utilizar los filtros FIR es debido a que estos son estables por definición y de esta manera solo nos enfocamos en estudiar la estabilidad del algoritmo, dado que éstos presentan estructuras recursivas y pueden presentar problemas de inestabilidad.

Como se verá a lo largo del trabajo, los algoritmos basados en el criterio de los mínimos cuadrados presentan muy buenas propiedades en cuanto a versatilidad con el tipo de señales que pueden trabajar y a la velocidad de convergencia. Sin embargo, presentan ciertos problemas numéricos cuando se trabaja con factores de olvido ($\lambda < 1$), sobre todo cuando se implementan en aritmética finita, 16, 32, 64 bits.

Tales problemas numéricos pueden hacerlos diverger. Dada las buenas propiedades de los algoritmos basados en los mínimos cuadrados, a lo largo de este trabajo se hace un estudio más detallado de éstos.

Puesto que al implementar un dispositivo adaptable dentro de un sistema de comunicaciones, el algoritmo adaptable deberá correr dentro de un dispositivo de procesamiento digital en tiempo real con una longitud de palabra de precisión finita, es de fundamental importancia analizar la robustez numérica de tales algoritmos. En particular, por robustez numérica nos referimos al estudio de los problemas de divergencia ocasionados por los errores numéricos que aparecen en las variables utilizadas en los algoritmos que se presentan debido a errores de truncamiento y redondeo dada la cantidad limitada de bits para representar tales variables.

La igualación de canal no es un problema nuevo ni exclusivo de las comunicaciones digitales. Desde el inicio de las transmisiones a distancia utilizando pares de cobre, se vio la necesidad de utilizar técnicas que permitieran compensar las distorsiones de los canales de propagación. Sin embargo, el problema es de actualidad dada la necesidad de mejorar e implementar nuevos servicios de comunicaciones, i.e. telefonía celular, redes de datos y servicios en tiempo real. La capacidad de realizar estas tareas depende de que tanto podamos aprovechar el canal de comunicaciones y por tanto de nuestra capacidad de compensar las deficiencias del canal mediante la utilización de igualadores adaptables.

Por otra parte, el empleo de igualadores adaptables es solo una parte del sistema de comunicaciones que esta compuesto por codificadores fuente, que nos permiten enviar la máxima cantidad de información en el ancho de banda disponible, codificadores de canal, que permiten la mayor inmunidad posible a ruido e interferencia y moduladores que adecúan el tipo de señal al canal de transmisión. Por esta razón, el problema de la igualación de canal va de la mano con estos otros problemas mencionados y en conjunto se busca hacer más eficiente la comunicación.

En un mundo donde las comunicaciones digitales forman parte de la sociedad es importante conocer los problemas que se presentan y saber como solucionarlos. La comprensión del problema nos permitirá el diseño de dispositivos y evitar ser tan solo consumidores de productos diseñados en otros países.

Finalmente, este trabajo tiene como objetivo fundamental, comprender el problema y estudiar los algoritmos de filtrado adaptable que lo solucionen. Así mismo, como hemos mencionado, la idea es que estos algoritmos sean implementados dentro de un dispositivo de procesamiento digital en tiempo real donde la representación numérica y operaciones aritméticas se realizan generalmente mediante una precisión finita, lo que ocasiona que los resultados obtenidos difieran de los teóricos y muchas veces esto ocasione divergencia del algoritmo. Por esta razón, de manera simultanea



a la búsqueda de algoritmos de filtrado adaptable que solucionen el problema de la igualación de canal, se realizará un análisis numérico de los errores ocasionados debido a errores numéricos dentro de los procesadores y se presentarán los algoritmos estables bajo aritmética finita que aseguren un buen funcionamiento dentro del dispositivo de procesamiento, e.g. DSP.

En este trabajo, además de plantear el problema de la igualación de canal como solución a la ISI. Se hace un estudio de la propagación de los errores en algoritmos recursivos del tipo LS basándonos en algunos documentos publicados. Así mismo, se realza la importancia de los errores numéricos en los algoritmos mediante la simulación de éstos bajo aritmética de precisión limitada.

ORGANIZACIÓN DE LA TESIS

En el **capítulo 1** se plantea el problema de la interferencia entre símbolos (ISI) en las comunicaciones digitales, ocasionado por el ancho de banda finito de los canales de comunicación fijos y por la propagación multitrayectoria en los canales inalámbricos. Por tanto, se resalta la necesidad de emplear técnicas que permitan eliminar la ISI y mejorar el desempeño de los sistemas de comunicación.

Se presenta un estudio de las propiedades de los canales de propagación más usados como lo es el par de cobre, el cable coaxial, la fibra óptica y el canal inalámbrico resaltando el modelado del canal inalámbrico.

En la última parte de este capítulo se presenta la igualación de canal como solución al problema de ISI y la necesidad de utilizar igualadores adaptables dadas las características de variancia con el tiempo del canal de propagación.

En el **capítulo 2** se presentan las herramientas básicas para el estudio de los algoritmos adaptables. Presentamos el modelado de procesos estocásticos mediante modelos ARMA, AR y MA dado que una gran cantidad de señales se pueden modelar de esta manera. Así mismo se presentan las diferentes tareas que pueden realizar los filtros digitales como lo son: la **predicción**, el **suavizado** y el **filtrado**.

Con el propósito de comprender a detalle el problema del filtrado se presenta la solución de los filtros de Wiener y Kalman lo que nos facilitará la comprensión del problema del filtrado adaptable.

En el **capítulo 3** entramos al estudio de los algoritmos de filtrado adaptable basados en dos de los criterios de optimización más utilizados, el del error cuadrático promedio (MSE) y el de los mínimos cuadrados (LS) resaltando las propiedades de cada uno de ellos.

Así mismo se obtienen los algoritmos LMS, RLS convencional y el "Fast Kalman" presentándose en la última parte de este capítulo las simulaciones correspondientes a estos algoritmos bajo precisión infinita de cálculo.

En el **capítulo 4** nos enfocamos al estudio de la robustez numérica de los algoritmos basados en los mínimos cuadrados. Se resalta la importancia del estudio del comportamiento numérico de los algoritmos dado que muchas veces pueden diverger precisamente debido a problemas numéricos.

En una primera parte analizamos la robustez numérica de dos implementaciones del algoritmo RLS clásico obtenido en el capítulo 3, llegando a la conclusión que el comportamiento numérico de un mismo algoritmo depende de la manera en que este se implemente.

Así mismo resaltamos los problemas numéricos de los algoritmos rápidos y la necesidad de trabajar con algoritmos estabilizados para aprovechar las buenas propiedades de los algoritmos FRLS. Dada la necesidad de trabajar con algoritmos estabilizados, presentamos una estrategia de estabilización propuesta por [15].

En el **capítulo 5** se presentan las simulaciones correspondientes a los algoritmos presentados en el capítulo 4 tanto en precisión infinita de cálculo como en aritmética entera resaltando el aumento de inestabilidad en los algoritmos debido a errores numéricos cuando se limita la precisión de cálculo.

En un último capítulo se presentan las **conclusiones** y para completar el trabajo se presentan algunos **apéndices** que nos ayudan a comprender mejor las ideas presentadas a lo largo de la tesis.

Capítulo 1

LOS CANALES DE COMUNICACIÓN Y LA INTERFERENCIA ENTRE SIMBOLOS (ISI).

1.1. Introducción

El canal de propagación es una parte fundamental del sistema de comunicación y determina en buena medida su eficiencia. Por esta razón, en este capítulo nos enfocaremos a la descripción de las características de los canales de propagación más significativos y presentaremos el modelado de los canales inalámbricos.

El estudio de los diferentes tipos de canales nos permitirá comprender los principales problemas que se presentan y que afectan el buen desempeño de los sistemas de comunicaciones. En particular, uno de dichos problemas es el de la interferencia entre símbolos (ISI) causado por el ancho de banda limitado de los canales y la propagación multitrayectoria en los canales inalámbricos.

En la sección 1.5 se estudia el problema de la interferencia entre símbolos y se presentan algunas alternativas para disminuirla. Así mismo, en la sección 1.5.3 se presenta la igualación de canal como solución al problema de la interferencia entre símbolos presentándose diferentes esquemas de igualación.

1.2. Tipos de canales de comunicación

Podemos ver un canal de comunicación como el enlace entre dos puntos a lo largo de una trayectoria de comunicación. Cuando definimos un canal específico debemos

PAGINACIÓN DISCONTINUA

indicar bajo que condiciones el canal presenta la propiedad de linealidad y/o reciprocidad.

Frecuentemente un canal se comporta de forma lineal solo sobre ciertas regiones de voltaje de entrada, temperatura, voltaje de alimentación, etc. La linealidad del canal es de gran importancia cuando se emplean esquemas de modulación sensitivos en amplitud como la modulación en cuadratura de amplitud *QAM*.

Un canal se denomina recíproco si su comportamiento no cambia conforme cambia la dirección en la que fluye la información.

Los canales utilizados en los sistemas de comunicación se muestran en la fig. 1.1. [29].

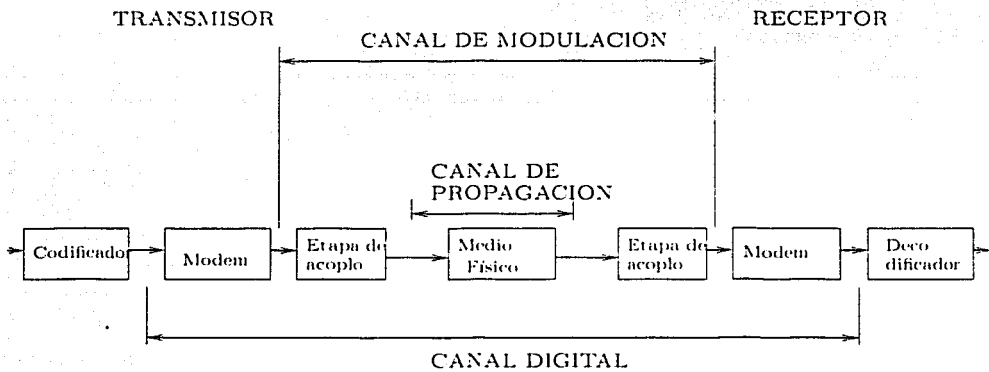


Figura 1.1: Clasificación de los canales de comunicación.

El canal de propagación es el que transporta las señales desde el equipo de acoplamiento en el transmisor hasta el equipo de acoplamiento en el receptor. Se asume que el canal de propagación es lineal y recíproco, sin embargo puede presentar la característica de variación con el tiempo.

Para el caso de un canal inalámbrico, el canal de propagación es recíproco si las antenas empleadas presentan reciprocidad. Las antenas presentan los mismos patrones de transmisión y recepción en el espacio libre si estas son bilaterales, lineales y pasivas. [19].

El canal de modulación se extiende desde la salida del modulador hasta la entrada del demodulador. Este canal es de gran interés para diseñadores de esquemas de

modulación y sistemas de codificación en Trellis, [29]. Si asumimos el canal de radio lineal, la linealidad del canal de modulación se determina por las características de transmisión de los equipos receptores y transmisores de acoplamiento. Los sistemas de modulación que emplean modulación multinivel de amplitud, requieren que el canal de modulación sea aproximadamente lineal.

El canal digital consiste de todos los componentes del sistema. (incluyendo el canal de propagación) enlazando la secuencia digital no modulada en el transmisor con la secuencia regenerada en el receptor. El canal digital es de importancia para ingenieros en codificación fuente y codificación de canal.

El hecho de definir estos tipos de canales tiene como objetivo mostrar como cada parte de un sistema de comunicación puede ser visto como un canal, esto es, para un ingeniero que se dedica a trabajar en codificación fuente, la parte de modulación, amplificación y el propio canal físico son para él, el canal de transmisión. [14].

1.3. Descripción de los canales de propagación

Existen diferentes medios o canales de propagación a través de los cuales se transmite la información en forma de señales. El tipo de señales a utilizar depende del tipo de canal de propagación: así, mientras en un par de cobre hacemos uso de señales eléctricas, en una fibra óptica utilizamos señales ópticas y en un canal inalámbrico usamos señales electromagnéticas.

Cada uno de los diferentes canales presentan características que los hacen adecuados para aplicaciones específicas, por esta razón es importante conocer los diferentes medios de transmisión.

1.3.1. Cables eléctricos

Para frecuencias inferiores a 300 MHz, se pueden utilizar cables eléctricos para transportar información en forma de señales eléctricas.

El cable telefónico está constituido por varios pares de hilos, cada uno de los cuales posee dos conductores retorcidos de cobre ó aluminio recubiertos de polietileno. En el canal telefónico la atenuación de la señal eléctrica varía en función de la raíz cuadrada de la frecuencia, esto significa que la atenuación se multiplica por dos si la frecuencia se multiplica por cuatro. En el caso de un par telefónico esta atenuación es del orden de 3dB después de un Km para una señal de 100KHz, y es la principal limitación para ancho de banda. Además de este inconveniente, el cable telefónico

también presenta una notable sensibilidad a los cambios de temperatura así como la presencia de diafonía¹. [31].

Un segundo tipo de cable es el cable coaxial. Está constituido por dos conductores concéntricos que se encuentran separados por un espacio lleno de un aislante eléctrico (que puede ser aire). El conductor interno es un hilo metálico y el conductor externo es un cilindro metálico. El cilindro externo limita la diafonía y las perturbaciones electromagnéticas externas.

Este cable presenta una atenuación más baja que el par de audio, y es del orden de 1.3dB para un Km de cable a 1MHz. Los cables coaxiales se utilizan rara vez a más de 100MHz. [31]. Dado que la atenuación varía en función de la raíz cuadrada de la frecuencia, podemos decir, por tanto que, tanto para el cable telefónico como para el cable coaxial, una distancia larga implica una baja frecuencia, o que una alta frecuencia corresponde a una corta distancia. Esto es un gran inconveniente cuando se intenta obtener un sistema de telecomunicaciones de alta eficiencia.

1.3.2. Fibra óptica

Es una guía de onda dieléctrica que transporta señales de luz de un lugar a otro tal como un par de cobre o un cable coaxial transporta señales eléctricas. Consiste de un núcleo central con índice de refracción n_1 dentro del cual es confinada la propagación del campo electromagnético. Este núcleo es rodeado por un recubrimiento de índice de refracción n_2 con $n_2 < n_1$, y finalmente el recubrimiento es cubierto por un delgado plástico.

Las fibras ópticas tienen características únicas que las hacen atractivas como medios de transmisión. En particular ofrecen las siguientes ventajas.[14]:

- Enorme potencial de ancho de banda.²
- Pocas pérdidas $0.2dB/Km$.
- Inmunidad a la interferencia electromagnética.
- Tamaño y peso pequeños.

¹Fenómeno de interferencia entre pares telefónicos adyacentes.

²Esto permite usar portadoras ópticas con frecuencias de alrededor de 200THz, con tan alta frecuencia portadora y un ancho de banda más ó menos igual al 10% de la frecuencia portadora, tenemos como ancho de banda teórico alrededor de 2×10^{13} Hz.

- Resistencia y flexibilidad.
- Bajo costo.

1.3.3. Canal satelital

Un canal satelital adiciona otra invaluable dimensión a las redes públicas de telecomunicaciones ya que provee una cobertura de gran área incluso intercontinental. Además, el acceso a áreas remotas cubiertas por cables convencionales o fibras ópticas, es una característica distintiva de los satélites.

En casi todos los sistemas de comunicación satelitales, los satélites son colocados en órbitas geoestacionarias. Visto desde la tierra, un satélite en órbita geoestacionaria parece estar estacionario en el cielo. Consecuentemente una estación terrena no tiene que rastrear el satélite, por el contrario, solamente tiene que apuntar su antena en una dirección fija hacia el satélite. Los sistemas de comunicación en órbitas geoestacionarias ofrecen las siguientes ventajas.[14]:

- Cobertura global.
- Enlaces de transmisión confiables.
- Amplios anchos de banda.

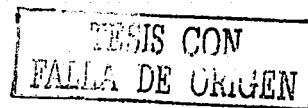
En términos de servicios, los satélites pueden proveer enlaces punto a punto fijos, extendiéndose a lo largo de grandes distancias y dentro de áreas remotas, comunicación para plataformas móviles como buques o naves espaciales, así mismo la capacidad de radiodifusión.

La banda de frecuencias más popular para comunicaciones satelitales es 6 GHz para la subida y 4 GHz para el enlace de bajada. El uso de esta banda de frecuencias ofrece las siguientes ventajas:

- Equipo de microondas relativamente barato.
- Baja atenuación por lluvia.¹
- Ruido celeste² insignificante.

¹La atenuación por lluvia es la principal causa atmosférica de pérdida de la señal.

²Este ruido, debido a emisiones de ruido aleatorio de fuentes galácticas, solares o terrestres alcanza su menor nivel entre 1 y 10 GHz.



1.4. Modelado de los canales móviles de comunicación

Los sistemas inalámbricos presentan características muy importantes que permiten dar servicios a grandes distancias sin embargo, el canal de propagación es el causante de muchos de los problemas y limitaciones que afectan los sistemas inalámbricos. Por ejemplo, la propagación por multitrayectorias, la cual es una de las características principales de los canales móviles de radio, ocasionada por la difracción y dispersión debido a las características del terreno y edificios.[23].

La propagación por multitrayectorias ocasiona distorsión en los sistemas analógicos y afecta severamente el desempeño de los sistemas digitales reduciendo las relaciones señal a ruido y señal a interferencia.[23].

Una comprensión del fenómeno y en consecuencia el modelado matemático del canal es muy importante ya que facilita una predicción más precisa del rendimiento del sistema y provee el mecanismo para probar y evaluar métodos de eliminación de los efectos negativos introducidos por el canal.

Como hemos mencionado, el canal de propagación influye en el rendimiento del sistema de radio comunicaciones, las señales llegan al receptor mediante un mecanismo de dispersión y la existencia de multitrayectorias con diferentes retardos, atenuaciones y fases. Esto da lugar a un canal complejo variante con el tiempo [23]. Un estudio detallado de los canales inalámbricos se puede consultar en [3].

1.4.1. El fenómeno de multitrayectorias

Los métodos para obtener predicciones de la intensidad de la señal en áreas urbanas ó dentro de edificios, son de gran importancia dado que la mayoría de los sistemas de comunicación operan en ciudades ó zonas muy pobladas. El mayor problema en las áreas de edificios es ocasionado debido a que la antena móvil se encuentra por debajo de los edificios que la rodean, lo que ocasiona que no exista línea de vista con el transmisor. De esta forma, la propagación se da a través de dispersión y difracción de la señal en los edificios. [23].

En la práctica, la energía llega al receptor por medio de diferentes caminos simultáneamente, presentándose así una situación "multitrayectoria". Las señales recibidas se combinan vectorialmente en la antena receptora dando lugar a una señal resultante, la cual puede ser grande o pequeña dependiendo de la distribución de fases entre las componentes recibidas. Es evidente que un receptor en alguna posición dada puede recibir una señal resultante con una potencia muy diferente a la recibida a tan solo unos metros de ésta última, esto es debido a que la relación de fases entre las difer-

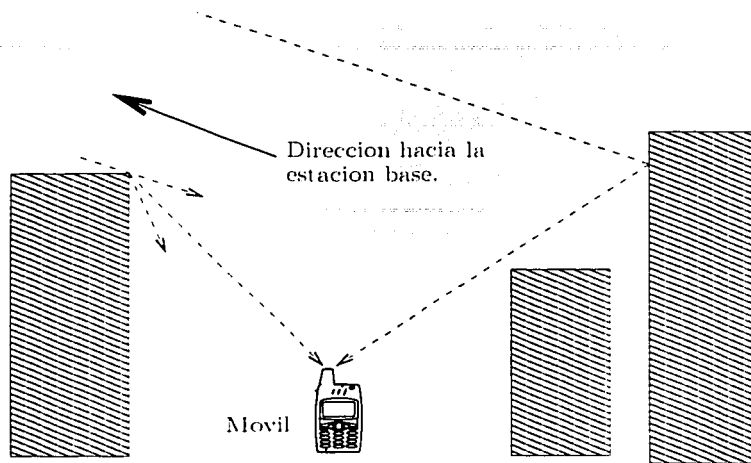


Figura 1.2: El fenómeno de multitrayectoria en una zona urbana.

entes componentes de la señal resultante (multitrayectorias) cambia de una posición a la otra. Por tanto, variaciones substanciales ocurren en la amplitud de la señal a lo largo del tiempo y del espacio. Estas fluctuaciones de la señal se conocen como desvanecimientos, si tales fluctuaciones ocurren en un corto tiempo el fenómeno se denomina *desvanecimiento rápido* para distinguirlo de aquel que presenta variaciones más lentas en el valor promedio de la señal a largo del tiempo, conocido como *desvanecimiento lento*. Este último es ocasionado por el movimiento sobre distancias suficientemente largas para producir variaciones brutales en la trayectoria global entre el transmisor y el receptor, generalmente estas variaciones son causadas por el movimiento del receptor dentro de la sombra de montes ó edificios conociéndosele también como *sombra*. [23].

El desvanecimiento es básicamente un fenómeno espacial, pero las variaciones espaciales se experimentan como variaciones temporales por un receptor moviéndose a lo largo de un campo de multitrayectorias.

En general, la propagación multitrayectoria, además de ocasionar desvanecimientos también ocasiona un ensanchamiento de los pulsos debido a que las diferentes componentes llegan al receptor con retardos, si estos retardos son mayores comparados con la velocidad de transmisión se produce interferencia entre símbolos y por tanto distorsión de los pulsos.[29].

TESIS CON
FALLA DE ORIGEN

1.4.2. El canal Gaussiano

El canal más simple es el canal gaussiano, a menudo llamado canal con ruido aditivo blanco gaussiano de sus siglas en inglés AWGN. Este canal se representa como un canal ideal en donde no existe multitrayectorias y se le agrega una fuente de ruido del tipo AWGN en el receptor tal como se muestra en la figura 1.3. Se asume que el ruido tiene una densidad espectral de potencia constante sobre el ancho de banda del canal y una función de densidad de probabilidad (*FDP*) gaussiana. Este tipo de canal prodría ser considerado como no realizable en comunicaciones inalámbricas, sin embargo no es así. Cuando se tienen microcélulas es posible tener una línea de vista, esencialmente sin multitrayectorias, dando un canal gaussiano. Incluso, cuando existe desvanecimiento por multitrayectorias, pero el móvil esta estacionario y no existen otros objetos moviéndose, como vehículos en su vecindad, el canal móvil puede pensarse como un canal gaussiano, representando los efectos del desvanecimiento por una trayectoria de perdidas local. [29].

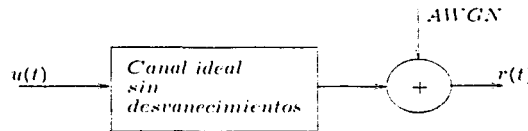


Figura 1.3: Canal gaussiano.

El canal gaussiano es importante ya que provee una idea de cual sería el mejor rendimiento del sistema sin la presencia de multitrayectorias, así, para un esquema de modulación dado, podríamos calcular la tasa de bits en error (*TBE*) en presencia de un canal gaussiano. Cuando los desvanecimientos por multitrayectoria ocurren, la *TBE* se incrementará para una relación señal a ruido dada. Utilizando técnicas para disminuir el desvanecimiento por multitrayectorias, tales como diversidad, igualación de canal, codificación de canal, alternación de datos, podemos observar como la *TBE* se aproxima a la que presentaría un canal gaussiano [29].

De esta forma podríamos evaluar que tan eficientes son nuestras técnicas de igualación, codificación, diversificación, etc: comparando el rendimiento de nuestro sistema con el que presentaría un sistema con un canal gaussiano para una relación señal a ruido dada.

TESIS CON
FALLA DE ORIGEN

Podemos notar que el canal gaussiano puede ser representado como una respuesta al impulso que tiene una función delta de peso constante, un canal ideal, al cual se le agrega una fuente de AWGN.

1.4.4. El canal Rician

En canales móviles con microcélulas una trayectoria dominante, la cual puede ser una trayectoria de línea de vista, frecuentemente aparece en el receptor, en adición con muchas otras trayectorias dispersas. Esta trayectoria dominante puede decrementar significativamente, la profundidad del desvanecimiento. En este caso la FDP de la envolvente recibida se denomina Rician. Para el modelado de este tipo de canal es conveniente introducir un parámetro K , conocido como parámetro Rician, con el propósito de medir la relación de potencia entre la señal dominante y el resto de las señales recibidas.[29].

$$K = \frac{\text{Potencia en la trayectoria dominante}}{\text{Potencia en las trayectorias dispersas}}$$

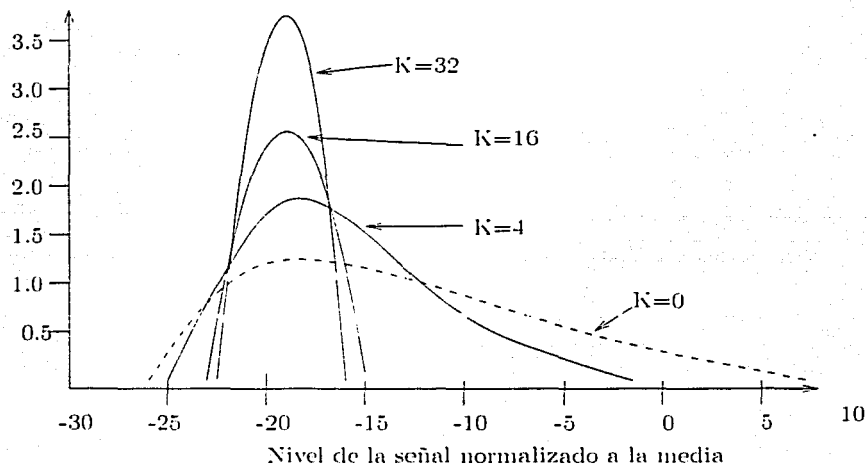


Figura 1.5: FDP Rician normalizada a sus promedios locales.

Cuando K tiende a cero el canal es del tipo Rayleigh, mientras que si K tiende infinito el canal es gaussiano. De esta manera, podemos considerar al canal Rician como un canal más general. Los desvanecimientos tienen una alta probabilidad de

ser más profundos cuando $K = 0$ (desvanecimiento Rayleigh) a ser muy poco profundos cuando $K = 32$ (aproximadamente gaussiano). Cuando la señal recibida está en un desvanecimiento profundo por debajo del nivel promedio del ruido del canal, un error ocurre. Sin embargo, el mismo promedio de ruido no causará error cuando K sea más grande. [29].

Las FDP's Rician para diferentes K normalizados con respecto a sus valores promedio locales se muestran en la fig. 1.5. Es evidente que la FDP Rayleigh tiene la mayor probabilidad de presentar un profundo desvanecimiento por debajo de la media, mientras que la FDP gaussiana tiene la menor.

1.5. Eliminación de la interferencia entre símbolos (ISI) en las comunicaciones digitales

En las comunicaciones digitales, el ancho de banda es limitado. Dado que para transmitir pulsos cuadrados se requiere de un ancho de banda infinito, es necesario filtrarlos, si estos pulsos no se filtran adecuadamente cuando pasan a través de un sistema de comunicaciones se dispersan en el tiempo y el pulso correspondiente a cada símbolo se alarga hasta llegar a interferir con la ranura de tiempo adyacente ocasionando lo que conocemos como interferencia entre símbolos (ISI)[5].

En canales de radio, la interferencia entre símbolos es debida a la propagación por multitrayectorias, la cual puede ser vista como una transmisión a través de un grupo de canales con diferentes retardos y amplitudes. Los igualadores adaptables son capaces de corregir la ISI debido a multitrayectorias de la misma forma que corrigen la ISI por distorsión lineal en canales telefónicos. [24].

En las líneas telefónicas, la dispersión temporal resulta cuando la respuesta en frecuencia del canal se desvía de la ideal que es amplitud constante y fase lineal (retardo constante). La función del equalizador es entonces compensar éstas características no ideales mediante el filtrado. [25].

En la figura 1.6 se muestra el problema de la interferencia entre símbolos para un sistema de comunicación binario, podemos apreciar que cada pulso transmitido es distorsionado por el canal y sufre una dispersión temporal lo que ocasiona que los pulsos se "empalmen". Al hacer la superposición de los pulsos en el receptor podemos detectar niveles incorrectos de voltaje además de limitar la capacidad de sincronización del sistema.

La ISI aparece en todos los sistemas de modulación de pulso incluyendo FSK, PSK y QAM. Sus efectos pueden ser fácilmente descritos si tomamos en cuenta un sistema

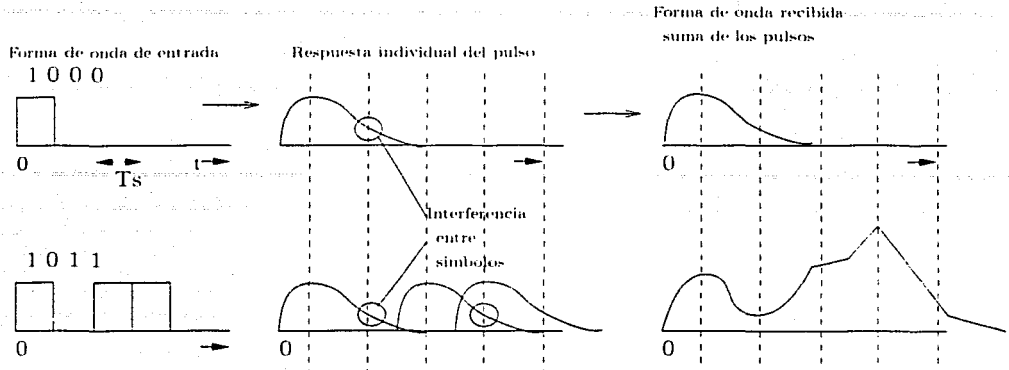


Figura 1.6: ISI en un sistema de comunicación binario

de modulación banda base en amplitud (PAM). [25].

El problema consiste en encontrar el ancho de banda necesario para que no exista interferencia entre símbolos. Por supuesto que si limitamos el ancho de banda para la transmisión de los pulsos, la forma de estos no seguirá siendo cuadrada. En la sección 1.5.2 presentamos el diseño de señales de banda limitada para canales de banda limitada como solución al problema de la ISI.

1.5.1. Diseño de señales para canales de banda limitada

En el esquema 1.7 se muestra un sistema de comunicaciones, donde $v(t)$ es una señal paso bajas con modulación digital que se puede representar como:

$$v(t) = \sum_{n=0}^{\infty} I(n)g(t - nT) \quad (1.1)$$

donde:

$I(n)$, representa la información discreta de la secuencia de símbolos.

$g(t)$, es un pulso de banda limitada ($G(f) = 0: \forall \|f\| > W$).

$v(t)$ se transmite por un canal que tiene una respuesta en frecuencia $C(f) = 0: \forall \|f\| \geq W$.

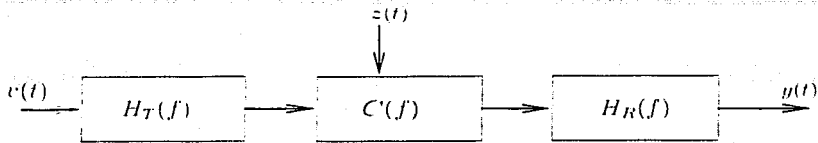


Figura 1.7: Sistema de comunicación básico

Entonces la señal recibida puede ser representada como:

$$r(t) = \sum_{n=0}^{\infty} I(n)h(t - nT) + z(t) \quad (1.2)$$

donde:

$$h(t) = \int_{-\infty}^{\infty} g(\tau)c(t - \tau)d\tau$$

y $z(t)$ representa ruido aditivo blanco gaussiano (AWGN).

La señal recibida se pasa por un filtro óptimo acoplado al pulso recibido con respuesta en frecuencia $H^*(f)$. [24].

La salida de este filtro la representamos como:

$$y(t) = \sum_{n=0}^{\infty} I(n)x(t - nT) + \nu(t)$$

donde:

$x(t)$. representa la respuesta del filtro al pulso $h(t)$.

$\nu(t)$. representa la respuesta del filtro al ruido $z(t)$.

Si ahora se muestrea a $t = kT + \tau_0$

$$y(kT + \tau_0) = y(k) = \sum_{n=0}^{\infty} I(n)x(kT - nT + \tau_0) + \nu(kT + \tau_0).$$

o bien:

$$y(k) = \sum_{n=0}^{\infty} I(n)x(k-n) + \nu(k); \quad \forall k = 0, 1, \dots \quad (1.3)$$

Donde τ_0 es el retardo que introduce el canal.

La ecuación 1.3 puede escribirse de la forma:

$$y(k) = x(0)(I(k) + \frac{1}{x(0)} \sum_{n=0, n \neq k}^{\infty} I(n)x(k-n)) + \nu(k); \quad \forall k = 0, 1, \dots$$

Considerando $x(0)$ como un factor de escala, e igualandolo a 1 por conveniencia, obtenemos:

$$y(k) = I(k) + \sum_{n=0, n \neq k}^{\infty} I(n)x(k-n) + \nu(k); \quad \forall k = 0, 1, \dots \quad (1.4)$$

donde:

$I(k)$, es la información deseada del símbolo al instante k de muestreo.

$\sum_{n=0, n \neq k}^{\infty} I(n)x(k-n)$, representa la interferencia entre símbolos.

$\nu(k)$, es el AWGN al instante k de muestreo.

Mediante un patrón de ojo se pueden analizar los efectos de la ISI. Básicamente la ISI aumenta la probabilidad de errores debido al AWGN y distorsiona la posición de cruce por cero ocasionando que el sistema sea más sensible a errores de sincronización.

Para el caso de modulación en fase o QAM podemos apreciar los efectos de la interferencia entre símbolos en las siguientes constelaciones:

Del lado izquierdo de la figura 1.9 se muestra una constelación de la señal transmitida con modulación 8PSK. Del lado derecho se muestra la señal recibida y los efectos de la ISI y el ruido. Como podemos observar, si la ISI es muy severa o trabajamos con más símbolos (e.g. 16PSK), la probabilidad de que se detecte un símbolo erróneo es mayor, por esta razón es importante el estudio de técnicas que permitan eliminar la ISI.

Ahora que hemos comprendido los efectos de la ISI consideraremos el diseño de señales bajo la condición de cero ISI en los instantes de muestreo.

TESIS CON
FALLA DE ORIGEN

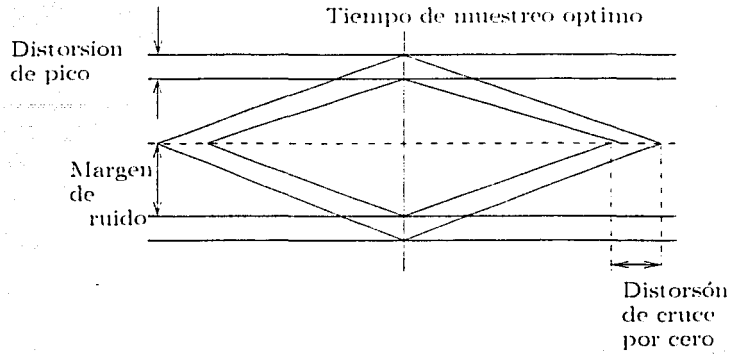


Figura 1.8: Diagrama de ojo

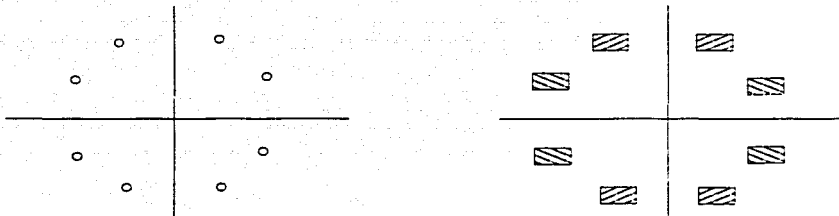


Figura 1.9: Constelación para una transmisión QAM

1.5.2. Diseño de señales de banda limitada para cero ISI

Si el canal de banda limitada tiene una respuesta ideal:

$$C(f) = 1: \quad \forall \|f\| \leq W$$

Entonces $x(t)$ tiene una característica espectral:

$$X(f) = \|G(f)\|^2$$

Donde:

$$x(t) = \int_{-W}^W X(f) e^{j2\pi ft} df \quad (1.5)$$

Nos interesan las propiedades espectrales del pulso $x(t)$ y por tanto del pulso $g(t)$ que nos permita cero ISI.

De la ecuación 1.4 la condición para cero ISI es:

$$x(t = kT) = x(k - n) = \begin{cases} 1: & k - n = 0 \\ 0: & k - n \neq 0 \end{cases} \quad (1.6)$$

La condición necesaria y suficiente sobre $X(f)$ para que $x(t)$ cumpla con lo anterior se conoce como el criterio de la forma del pulso de Nyquist o "condición de Nyquist para cero ISI" [24]

Se puede demostrar que la condición necesaria y suficiente para que $x(t)$ satisfaga la ecuación 1.6 es que su transformada de Fourier satisfaga, [24]

$$\sum_{m=-\infty}^{\infty} X(f + \frac{m}{T}) = T \quad (1.7)$$

Si consideramos W como el ancho de banda del canal $C(f) = 0 \quad \forall \|f\| > W$, en consecuencia $X(f) = 0: \quad \forall \|f\| > W$.

Podemos apreciar tres casos:

1.- Si $T < \frac{1}{2W}$, entonces:

$$B(f) = \sum_{m=-\infty}^{\infty} X(f + \frac{m}{T})$$

Consiste de réplicas de $X(f)$ que no se empalman y que se encuentran separadas $\frac{1}{T}$, por lo tanto no existe forma de eliminar la ISI.

2.- Si $T = \frac{1}{2W}$ (Tasa de Nyquist), entonces las replicas de $X(f)$ están continuas y por lo tanto existe una opción para que $B(f) = T$. Esto es si:

$$X(f) = \begin{cases} T: & \|f\| < W \\ 0: & \text{otro caso} \end{cases}$$

De acuerdo a lo anterior podemos observar que el valor mínimo de T para que sea posible una transmisión con cero ISI es $T = \frac{1}{2W}$ y para tal valor $x(t)$ debe de ser una función seno cardinal. En este caso $x(t)$ es no causal y por tanto no realizable.

3.- Si $T > \frac{1}{2W}$, entonces:

$B(f)$ consiste de replicas empalmadas de $X(f)$ separadas $1/T$. La figura 1.10 muestra el espectro de $B(f)$ en donde las réplica de $X(f)$ se empalman. Existen por lo tanto numerosas opciones para lograr $B(f) = T$.

Para este caso el espectro del coseno elevado tiene propiedades espectrales deseables y es usado ampliamente, [24].

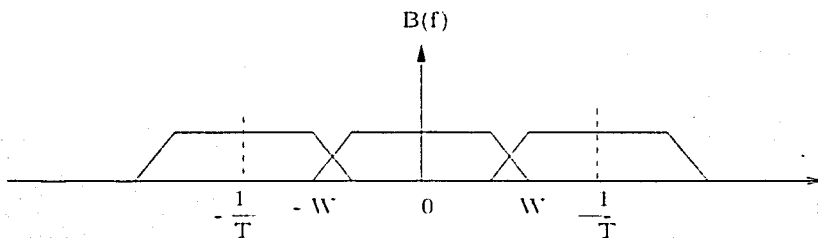


Figura 1.10: Espectro de $B(f)$

Otra opción para la eliminación de la ISI es el empleo de señales de banda limitada con ISI controlada, esto es: de antemano se diseña el pulso a transmitir considerando que se verá afectado por algunas componentes de ISI, sin embargo, dado que a priori sabemos que componentes de ISI son las que no se quitaron, en el proceso de recepción estas son eliminadas con la ayuda de filtros. Es decir, la ISI que se introduce deja de ser aleatoria pues se sabe que componentes serán afectadas y por lo tanto estamos hablando de una ISI determinística que puede ser eliminada en el receptor.

1.5.3. La igualación de canal como solución al problema de ISI.

Las comunicaciones modernas, plantean un problema de gran importancia que consiste en diseñar "filtros igualadores" que compensen los efectos de la distorsión y del ruido que son introducidos por el sistema de comunicaciones a la señal que viaja a través del canal. Una característica de los canales físicos es que tienen un ancho de banda finito, esto distorsiona a la señales en fase y en amplitud, además el factor ruido que producen los elementos físicos provoca que la señal portadora de información llegue atenuada y con distorsión. Los filtros en general sirven como una etapa de preprocesamiento para la cual la señal portadora de información, o sus parámetros, pueden identificarse de una manera más clara que a la entrada del filtro.

El método de diseño de filtros que es empleado en las comunicaciones modernas para su diseño utiliza las características estadísticas de la señal y del ruido. [1-4].

Es común encontrarnos con canales cuya respuesta en frecuencia no es conocida o es variante con el tiempo. Por ejemplo, en telefonía el canal de comunicación cambia sus características frecuenciales cada vez que marcamos un número, esto se debe al cambio de ruta de la señal. Una vez que la conexión se realiza, el canal será invariante con el tiempo salvo por pequeñas variaciones debido principalmente a cambios de temperatura. Por otra parte, los sistemas móviles son un ejemplo clásico de canales variantes con el tiempo, en dichos canales los filtros en el transmisor y receptor no pueden ser establecidos de antemano y es necesario utilizar igualadores adaptables.

Los filtros igualadores pueden ser clasificados de acuerdo al criterio de optimización con el que se actualizan sus parámetros y también de acuerdo a su estructura, ya sea transversal o "lattice".

IGUALADORES ADAPTABLES

Los sistemas de comunicaciones actuales utilizan los igualadores adaptables como una pieza clave del receptor. El surgimiento de estos igualadores se dió rápidamente sobre todo en aplicaciones donde, debido a limitantes físicas del canal de transmisión, económicas o de regulación, se limitaba el ancho de banda a utilizar para la transmisión.

Previo a la transmisión es necesario calibrar el igualador de acuerdo a las características del canal, esto se realiza durante un periodo llamado *entrenamiento*. En algunas aplicaciones en tiempo real el periodo de entrenamiento es suprimido y la calibración del canal se hace con la propia señal transmitida, a este tipo de igualadores se les conoce como igualadores ciegos.

Durante el *periodo de entrenamiento*, una señal conocida es transmitida y una señal idéntica y sincronizada es generada en el receptor para que de esta forma se puedan obtener las características del canal. Esta señal debe de ser por lo menos tan larga como el igualador, para que se tenga la información suficiente para que el igualador pueda inicializarse correctamente.

Después del *periodo de entrenamiento*, con el objetivo de seguir pequeñas variaciones en el canal, los coeficientes del igualador adaptable se deben de estar ajustando continuamente. Para esto, una vez que se ha terminado el *periodo de entrenamiento*, el igualador adaptable cambia la retroalimentación del dispositivo generador de la secuencia aleatoria al dispositivo de decisión, de esta forma la señal de error es derivada de la señal estimada por el receptor $\hat{d}(k)$, la cual no es necesariamente correcta, figura 1.11. Si el funcionamiento del igualador es bajo condiciones normales, se tiene una alta probabilidad de que las decisiones tomadas por el igualador sean correctas. Es importante señalar que el igualador puede seguir variaciones lentas del canal, así como inestabilidades de fase en el receptor.

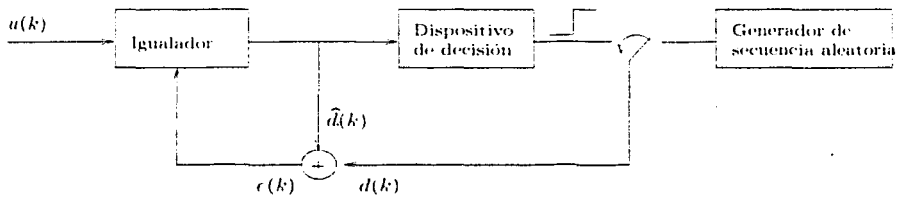


Figura 1.11: Esquema de un igualador adaptable

Igualador de pasos descendentes

En las implementaciones prácticas, la solución para los coeficientes óptimos se obtiene empleando un procedimiento recursivo que evita la inversión de la matriz de covarianza \mathbf{R} en forma directa. El procedimiento más simple es el *método de pasos descendentes* basado en el criterio del error cuadrático promedio, que consiste en escoger un vector de parámetros arbitrario W_0 , el cual corresponde a un punto sobre la curva de desempeño del error. El vector gradiente $g(k)$ es entonces calculado en este punto, pero como $g(k)$ apunta en la dirección de más rápido crecimiento y el punto mínimo buscado está en la dirección opuesta a éste, es necesario cambiar la dirección del vector $g(k)$ en cada iteración, obteniéndose así :

$$g(k) = \mathbf{R}W(k) - r, \quad k = 0, 1, 2, \dots \quad (1.8)$$

donde R y r son respectivamente la matriz y vector de covarianza del proceso, y el vector $W(k)$ que contiene los parámetros del filtro igualador se actualiza mediante la siguiente relación:

$$W(k+1) = W(k) - \mu g(k) \quad (1.9)$$

donde μ es un parámetro de *paso de convergencia* o *factor de ponderación* para el procedimiento iterativo. Para asegurar la convergencia del algoritmo, μ debe de escogerse pequeño y positivo, de tal forma que $g(k)$ tenga una convergencia hacia cero, e.g. $g(k) \rightarrow 0$ cuando $k \rightarrow \infty$, y el vector de coeficientes $W(k) \rightarrow W_{\text{optimo}}$. Existe un compromiso entre el tiempo de convergencia y μ , ya que entre más pequeño sea μ , menor será el MSE, sin embargo, el tiempo de convergencia será mucho mayor, por otro lado si μ es mayor, el tiempo de convergencia es mucho menor pero el MSE mínimo será mayor que en el primer caso mencionado. [14].

Este método presenta el inconveniente de no llegar a los W_{opt} en un número finito de iteraciones, sin embargo se puede obtener una buena aproximación con algunos cientos de iteraciones.

Igualador de gradiente estocástico

Con el objetivo de que el algoritmo de pasos descendentes siga las variaciones del canal se hace uso de un gradiente estocástico en lugar de un gradiente determinístico, esto es:

$$W(k+1) = W(k) - \mu \hat{g}(k) \quad (1.10)$$

donde $\hat{g}(k)$ es el estimado del vector de gradiente $g(k)$ que se calcula como:

$$\hat{g}(k) = -e(k)u(k) \quad (1.11)$$

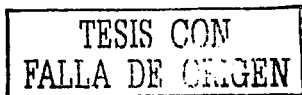
Sustituyendo la ecuación 1.11 en 1.10 tenemos:

$$W(k+1) = W(k) + \mu e(k)u(k) \quad (1.12)$$

Este algoritmo es conocido como **algoritmo LMS** o **algoritmo de gradiente estocástico**. Al igual que el *algoritmo de pasos descendentes*, es necesario tener un compromiso entre velocidad de convergencia y el valor mínimo del MSE. [14].

Igualador zero-forcing ZF

En un canal las distorsiones introducidas pueden ser compensadas empleando un filtro lineal que posea parámetros que puedan ser ajustados. Es decir, si se tiene



un canal invariante con el tiempo con respuesta en frecuencia desconocida, se miden las características del canal y se ajustan los parámetros del igualador. Una vez hecho esto, los parámetros se conservarán durante toda la transmisión de datos. [24].

De acuerdo a su estructura estos filtros pueden ser transversales o "lattice". La más simple es la estructura transversal. En la cual los valores presentes y pasados de la señal recibida son ponderados por los coeficientes del igualador y sumados para obtener la salida. Los coeficientes del igualador deben escogerse de tal forma que hagan que las muestras del canal y la respuesta al impulso del igualador sean cero para todos los valores excepto en uno de los N instantes. Lucky en 1965 propuso el algoritmo "zero-forcing" para ajustar los coeficientes de un filtro transversal: se emplea el concepto de "distorsión pico" el cual está directamente relacionado con el valor máximo de ISI que pueda ocurrir: los coeficientes del filtro se ajustan de modo que se reduzca la distorsión pico. Entonces se presenta el efecto de "forzar" la ISI debida a los pulsos del filtro transversal hasta "cero": de ahí el nombre del algoritmo. [14].

Si dejamos que el número de coeficientes del igualador **ZF** crezca de manera ilimitada, se obtendrá un igualador de longitud infinita con cero ISI a su salida. La respuesta de tal igualador será periódica e igual a la tasa de símbolos. Es posible demostrar que un igualador de longitud infinita con cero ISI es simplemente un filtro inverso de la respuesta en frecuencia del canal. Sin embargo, tal tipo de igualador amplificará el ruido en las frecuencias para las cuales el espectro del canal tiene grandes atenuaciones. [25].

Igualador fraccionalmente espaciado

El ISI en canales reales sólo afecta un número finito de muestras. Ésto trae como consecuencia que se utilice un filtro a respuesta impulsional finita (FIR) o filtro transversal, con coeficientes variables. El tiempo de retardo τ se debe escoger tan largo como el intervalo entre símbolos T . Si τ se escoge de tal forma que $1/\tau \geq 2W > 1/T$, donde W es el ancho de banda del canal, no ocurre traspase entre los símbolos y por lo tanto el igualador inverso al canal compensa la distorsión del mismo. Como $\tau < T$ el canal es llamado *fraccionalmente espaciado* y de ahí su nombre de *igualador fraccionalmente espaciado (FSE)*. El valor que más se utiliza en la práctica es $\tau = T/2$.

Los igualador **FSE**, gracias a su frecuencia de muestreo, pueden obtener las mejores características de un filtro acoplado adaptable, dentro de las limitaciones de su periodo. Un igualador con periodo T no puede ser acoplado. Un **FSE** puede compensar distorsiones de retardo más severas con menos amplificación del ruido.

Es posible demostrar que con un diseño adecuado de la característica del filtro, la

tasa de símbolos a la salida del *igualador fraccionalmente espaciado*, puede utilizarse sin dejar de ser óptima, para cualquier receptor lineal o no lineal sin importar el criterio de optimización empleado, [14].

1.6. Conclusiones

El canal de propagación es la pieza clave de la operación de un sistema de comunicación. Sus propiedades determinan tanto la capacidad del sistema de llevar información como la calidad del servicio ofrecido. Podemos clasificar los canales de comunicación de diferentes maneras. [14].

En este capítulo hemos estudiado diversos tipos de canales enfatizando en el modelo de los canales inalámbricos puesto que en un canal de este tipo podemos observar con más claridad la característica de variancia con el tiempo. Los canales fijos como líneas telefónicas por par de cobre, líneas de cable coaxial o fibra óptica también presentan variación con el tiempo, por esta razón la utilización de filtros adaptables no es exclusiva de los canales inalámbricos.

Finalmente, hemos discutido diferentes maneras de solucionar el problema de la interferencia entre símbolos dada por el ancho de banda limitado de los canales fijos o la propagación multitrayectoria en canales inalámbricos, terminando con la introducción de diferentes esquemas de igualación de canal. La utilización de uno u otro esquema dependerá de la aplicación¹. Dado que nuestro objetivo no es hacer un análisis exhaustivo de los diferentes esquemas de igualación, nos enfocaremos en estudiar las propiedades numéricas de los algoritmos, en los capítulos 3 y 5 utilizaremos un esquema de igualación lineal para comparar los diferentes algoritmos.

En el capítulo 2 se presentan las bases teóricas para el estudio de los algoritmos de filtrado adaptable basados en dos de los criterios de optimización más usados, el de los mínimos cuadrados (**LS**) y el del error cuadrático promedio (**MSE**).

¹Si se desea un análisis detallado de los diferentes esquemas de igualación se puede ver [22].

Capítulo 2

TEORÍA DE FILTROS LINEALES ÓPTIMOS

2.1. Introducción

En este capítulo presentamos las bases del filtrado. Comenzamos con el estudio de los procesos estocásticos modelados a través de modelos AR, MA o ARMA, esto es de gran importancia pues una gran variedad de señales reales pueden ser modeladas por modelos estocásticos.

Se presentan además los tres problemas básicos del filtrado. (filtrado, predicción y suavizado), y las principales características de los algoritmos de filtrado adaptables. En la sección 2.4 se presenta el problema de filtrado planteado por Wiener mediante la representación de un modelo de ecuación en diferencias (filtro transversal) y se resuelve aplicando el criterio del error cuadrático promedio. Por último, en la sección 2.5 se presenta el problema del filtrado de Kalman el cual utiliza un modelo representado en variables de estado y se obtiene su solución mediante un método conocido como proceso de innovación en donde se supone que la nueva información referente al estado actual del filtro esta dada por el error de filtrado actual.

2.2. Modelado de procesos estocásticos

El término "modelo" se utiliza para cualquier hipótesis que puede ser aplicada con el fin de explicar o describir las leyes ocultas que se supone gobiernan o limitan la generación de los datos de interés. La representación de procesos estocásticos mediante modelos data de una idea de Yule (1927). [13].

Una secuencia $u(k)$ constituida por observaciones altamente correlacionadas, puede ser generada a partir de un proceso aleatorio estadísticamente independiente apli-

cado a la entrada de un filtro como se muestra en la figura 2.1

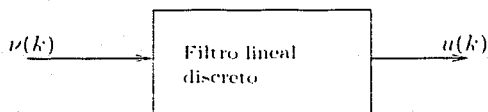


Figura 2.1: Modelo de un filtro generador de señales estocásticas.

El proceso de entrada comúnmente se supone gaussiano con media cero y varianza constante. El término *gaussiano* se refiere a su función de densidad de probabilidad, mientras que sus características estadísticas se expresan como:

$$E[\nu(k)] = 0; \forall k$$

$$E[\nu(k)\nu^*(j)] = \begin{cases} \sigma_\nu^2; & j = k \\ 0; & \text{otro caso} \end{cases}$$

En general, la descripción en el dominio del tiempo de la relación entrada-salida para el modelo estocástico de la figura 2.1 se puede escribir como se muestra en la figura 2.2:

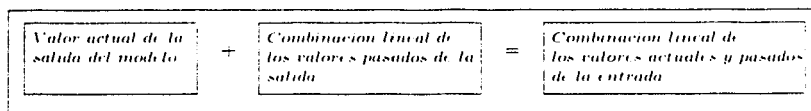


Figura 2.2: Descripción del filtro generador de señales estocásticas.

Con base en la figura 2.2 podemos identificar tres tipos de modelos estocásticos.[13]:

1. Modelos autoregresivos (AR). En los que no se utilizan valores pasados de la entrada del modelo.
2. Modelos de promedio móvil (MA). En los que no se utilizan valores pasados de la salida del modelo.
3. Modelos autoregresivos de promedio móvil (ARMA). En los que la descripción de la figura 2.2 aplica en su totalidad.

MODELOS AUTOREGRESIVOS

Una secuencia $U(k) = u(k), u(k-1), \dots, u(k-M)$ que proviene de un modelo autoregresivo de orden M satisface la ecuación en diferencias:

$$u(k) + a_1^* u(k-1) + \dots + a_M^* u(k-M) = v(k) \quad (2.1)$$

Donde a_1, a_2, \dots, a_M ¹ son los parámetros del modelo y $v(k)$ es un proceso de ruido blanco, es decir, es la secuencia de variables aleatorias independientes que se encuentran a la entrada de la figura 2.1.

Si despejamos $u(k)$ de la ecuación 2.1 podemos apreciar que el valor actual del proceso $U(k)$ es igual a una combinación lineal de los valores pasados más un término de error $v(k)$, de aquí el nombre de modelo **autoregresivo**.

Podemos escribir en forma de sumatoria de convolución la ecuación 2.1 como:

$$\sum_{j=0}^M a_j^* u(k-j) = v(k) \quad (2.2)$$

Si consideramos a los parámetros como una señal de tamaño M , donde M es el orden del filtro generador, entonces podemos denotar a las transformadas Z de los parámetros del filtro, la secuencia de salida y el ruido blanco como $H_A(z), U(z)$ y $V(z)$ respectivamente, y podemos escribir el polinomio característico asociado a la relación de la ecuación 2.2, como:

$$H_A(z)U(z) = V(z) \quad (2.3)$$

Dada la ecuación 2.3 podemos obtener la función de transferencia para dos situaciones: $u(k)$ como entrada para obtener un ruido blanco a la salida y $v(k)$ como entrada para obtener la secuencia $u(k)$ a la salida. En particular nos interesa la segunda situación dada por la función de transferencia:

$$H_G(z) = \frac{U(z)}{V(z)} = \frac{1}{H_A(z)} = \frac{1}{\sum_{k=0}^M a_k^* Z^{-k}} \quad (2.4)$$

El generador por lo tanto, tiene una respuesta al impulso infinita y corresponde a un filtro todo polo ya que su función de transferencia se define completamente especificando la ubicación de los polos.

Un modelo autoregresivo de este tipo tiene una estructura como la que se muestra en la figura 2.3.

Para que el generador del proceso AR sea estable debemos asegurar que los polos se encuentren dentro del círculo unitario, lo cual es a su vez una condición necesaria y

¹El * denota complejo conjugado en el caso general de señales complejas.

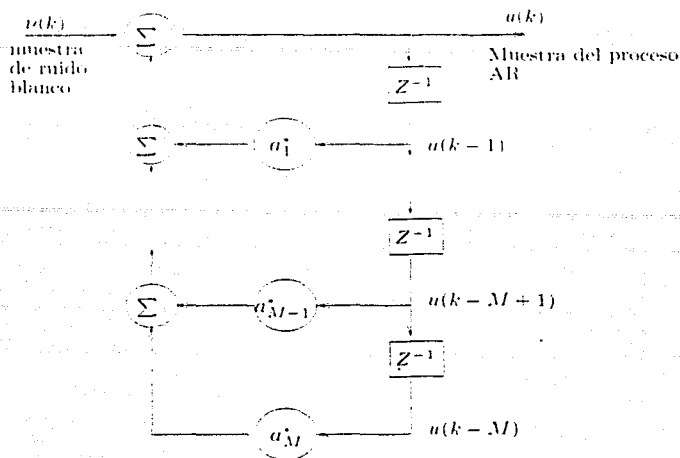


Figura 2.3: Estructura generadora de un proceso AR $u(k)$.

suficiente para estacionariedad en sentido amplio del proceso. [13].

MODELOS DE PROMEDIO MÓVIL

En este caso, el proceso resultante a la salida del filtro se describe por la ecuación en diferencias:

$$u(k) = \nu(k) + b_1^* \nu(k-1) + \dots + b_N^* \nu(k-N) \quad (2.5)$$

Donde b_1, b_2, \dots, b_N son los parámetros del modelo y $\nu(k)$ es un proceso de ruido blanco con las características estadísticas ya mencionadas. En este caso el orden del proceso de promedio móvil es igual a N y el filtro tiene una respuesta al impulso finita puesto que su función de transferencia está dada por:

$$\frac{U(z)}{V(z)} = H_B(z) = \sum_{k=0}^N b_k^* Z^{-k} \quad (2.6)$$

La función de transferencia de la ecuación 2.6 corresponde a un modelo todo cero. Entonces, podemos concluir que dada una realización del proceso $\nu(k)$ podemos obtener nuestra secuencia $u(k)$ mediante una combinación lineal de las muestras $\nu(k), \nu(k-1), \dots, \nu(k-N)$. La estructura para un modelo de promedio móvil se muestra en la figura 2.4.

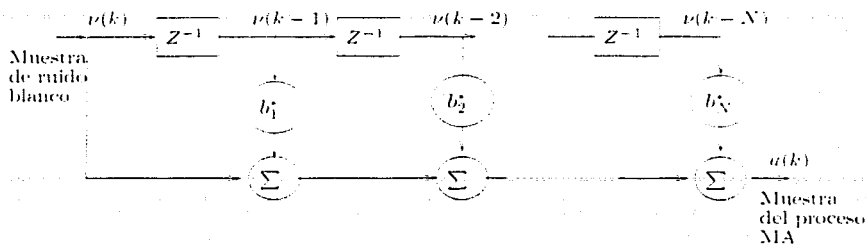


Figura 2.4: Estructura generadora de un proceso MA $u(k)$.

MODELOS AUTOREGRESIVOS DE PROMEDIO MÓVIL

En este caso la función de transferencia incluye tanto polos como ceros y su estructura es la unión de una estructura AR, figura 2.3 y una estructura MA, figura 2.4. Un proceso derivado de un modelo autoregresivo de promedio móvil es descrito por la siguiente ecuación en diferencias:

$$u(k) + a_1^* u(k-1) + \dots + a_M^* u(k-M) = v(k) + b_1^* v(k-1) + \dots + b_N^* v(k-N) \quad (2.7)$$

Donde a_1, a_2, \dots, a_M y b_1, b_2, \dots, b_N se denominan parámetros ARMA y el orden del proceso es igual a (M, N) .

Desde un punto de vista computacional el modelo autoregresivo tiene una ventaja sobre los modelos MA y ARMA ya que el cálculo de los coeficientes en el modelo AR involucra un sistema de ecuaciones lineales conocidas como ecuaciones de Yule-Waker, el cual será resuelto de una manera eficiente en una sección más adelante haciendo uso de las propiedades de la matriz de correlación del proceso $U(k)$. Por otra parte, la obtención de los coeficientes de los modelos involucra sistemas de ecuaciones no lineales y por consecuencia más difíciles de resolver, [27].

La importancia de representar procesos estocásticos o señales aleatorias mediante modelos AR, MA y ARMA es importante dado que un gran número de señales reales se pueden modelar mediante estos modelos (filtrado, predicción y suavizado), [27].

2.3. Aplicaciones de los filtros digitales.

En esta sección se describirán las tres operaciones básicas de un filtro, haciendo mayor énfasis en el filtrado y la predicción. Se discutirá sobre el proceso de optimización de un filtro a partir de las suposiciones sobre las señales de entrada y los

criterios de optimización empleados. Así mismo, se describirá el funcionamiento y las características de un filtro adaptable.

2.3.1. Funciones de los filtros.

Llamamos filtro a todo dispositivo, ya sea de software o hardware, que es aplicado a un grupo de datos ruidosos con el propósito de extraer información acerca de una cantidad de interés. En general un filtro puede realizar tres operaciones básicas de procesamiento de información:

1. **Filtrado.** Consiste en extraer la información alrededor de una cantidad de interés hasta el tiempo t utilizando los datos medidos hasta el tiempo t .

2. **Suavizado.** La información alrededor de la cantidad de interés no necesita estar disponible en el tiempo t y los datos medidos después del tiempo t pueden utilizarse para obtener esta información. La idea del suavizado es obtener información adicional entre las muestras (interpolación) con el propósito de tener una mejor resolución (mayor número de muestras), lo cual tendrá el efecto de que la señal se vea más "suave".

3. **Predicción.** La idea es obtener información acerca de la posible cantidad de interés al tiempo $t + \tau$ en el futuro, $\forall \tau > 0$, utilizando los datos medidos hasta el instante t .

En general podemos clasificar los filtros en lineales y no lineales, se dice que un filtro es lineal si la cantidad filtrada, suavizada o estimada a la salida del dispositivo es una función lineal de los datos a la entrada, en cualquier otro caso el filtro se considera no lineal.

En la solución del problema de filtrado lineal se utiliza una aproximación estadística de los datos y se asume que se dispone de ciertos parámetros como media y funciones de correlación tanto de la señal deseada como del ruido aditivo. Tal solución se basa en la minimización de los efectos del ruido a la salida del filtro de acuerdo con algún criterio estadístico.

Una aproximación utilizada para la solución del problema de filtrado es minimizar el valor del error cuadrático promedio definido como el valor esperado de la señal de error, frecuentemente definida como la diferencia entre la respuesta deseada y la salida del filtro, con el fin de tener un filtro óptimo. Para entradas estacionarias la solución resultante se conoce como filtro Wiener, óptimo en el sentido del error cuadrático promedio, por el contrario, si la señal de entrada es no estacionaria el filtro óptimo tiene que asumir una forma variante con el tiempo. Una solución exitosa

para este problema es el filtro de Kalman. [13].

2.3.2. Filtros adaptables

Un filtro adaptable es un dispositivo autodiseñable que opera con un algoritmo recursivo, esto hace posible que el filtro se comporte satisfactoriamente en un ambiente donde no se disponga completamente de las características de la señal relevante. El algoritmo debe iniciar con un grupo de condiciones iniciales con base en las características del entorno conocido.

Una gran variedad de algoritmos recursivos han sido diseñados para la operación de filtros lineales adaptables, la elección de uno u otro se puede determinar por uno o más de los siguientes factores. [13]:

Tasa de Convergencia. Se define como el número de iteraciones requeridas por el algoritmo para converger a la solución óptima en el sentido de algún criterio de optimización. Una tasa rápida de convergencia permite al algoritmo adaptarse rápidamente a un ambiente de estadísticas desconocidas.

Desajuste. Este parámetro provee una medida cuantitativa de la cantidad por la cual el valor final del error se desvía del error mínimo producido por el filtro de Wiener.

Seguimiento. Cuando un algoritmo opera en un medio no estacionario, se requiere que este de seguimiento a las variaciones estadísticas en el medio. La capacidad de seguimiento del algoritmo es influenciada por dos características que van de la mano: Tasa de convergencia y fluctuaciones debido al ruido del algoritmo.

Robustez. Para que un filtro sea robusto, pequeñas alteraciones (alteraciones de poca energía) solo deben ocasionar errores de estimación pequeños. Tales alteraciones pueden ser internas ó externas al filtro.

Requerimientos de cómputo. Incluyen número de operaciones requeridas para completar una iteración, memoria requerida para almacenar los datos y la inversión para programar el algoritmo.

Estructura. El flujo de la información en el algoritmo determina la manera en la que este es implementado en hardware. Un algoritmo cuya estructura exhibe alta modularidad, paralelismo ó concurrencia es adecuado para implementaciones en microprocesadores.



Propiedades numéricas. Las imprecisiones son producidas debido a errores de cuantización, estas ocasionan básicamente dos problemas: estabilidad y precisión numérica. La estabilidad es una característica inherente del algoritmo y la precisión se determina por el número de bits utilizados para representar las muestras y los coeficientes del filtro.

2.3.3. La predicción lineal

Como se mencionó al inicio de esta sección, una de las operaciones de los filtros puede ser la de predicción. Cuando nos referimos a la predicción esta puede ser predicción "forward" o bien predicción "backward", en el primer caso el problema consiste en estimar la señal deseada al tiempo k cuando se conoce un conjunto de p muestras pasadas $U(k-1) = [u(k-1), u(k-2), \dots, u(k-p)]$, por el contrario; en el caso de la predicción backward el problema es el de estimar el valor de la señal deseada al instante anterior p cuando se conocen p futuras muestras de la señal $U(k-p) = [u(k), u(k-1), \dots, u(k-p+1)]$.

Para el caso de la predicción forward se tiene el sistema de la figura 2.5. En este caso, la señal deseada es una versión adelantada de la señal de entrada al filtro. En el momento que el filtro logre converger, este último representa un modelo para la señal de entrada y puede ser usado como un modelo predictor de la señal de entrada. [7].

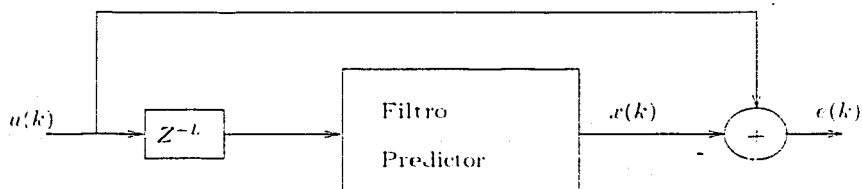


Figura 2.5: Sistema para la predicción de señales.

Con base en la figura 2.5 podemos escribir la siguiente ecuación que representa el error de predicción:

$$e(k) = u(k) - x(k) \quad (2.8)$$

Donde la salida del filtro predictor esta dada por una combinación lineal de la señal de entrada y los parámetros del filtro, de la siguiente forma:

$$x(k) = - \sum_{j=1}^p a_j^*(k) u(k-j)$$

Por lo tanto queda definido el error como¹:

$$\varepsilon_p^f[k] = u(k) + \sum_{j=1}^p a_j^T(k) u(k-j) \quad (2.9)$$

O en su forma vectorial:

$$\varepsilon_p^f[k] = u(k) + A_p^T(k) U_p(k-1) \quad (2.10)$$

donde:

$$A_p^T(k) = [a_1(k), a_2(k), \dots, a_p(k)]$$

$$U_p^T(k-1) = [u(k-1), u(k-2), \dots, u(k-p)]$$

El problema en la predicción consiste en optimizar (minimizar) el error a la salida del filtro predictor dado por la ecuación 2.8 de acuerdo con algún criterio de optimización. En el apéndice B se obtiene la solución del filtro predictor utilizando dos de los criterios más usados, el del error cuadrático promedio y el de los mínimos cuadrados.

2.3.4. El filtrado

El problema fundamental del filtrado se ilustra en la figura 2.6. Básicamente, este problema consiste en obtener un buen estimador para una señal deseada $d(k)$ a la salida del filtro caracterizado por los parámetros $\omega_0, \omega_1, \omega_2, \dots$ cuya entrada es la señal $u(k)$. Dado que el filtro proporciona una salida $\hat{d}(k)$ diferente a la señal deseada, podemos cuantificar un error de estimación $e(k)$ dado por la ecuación 2.11. El propósito de un filtro es minimizar este error de estimación al máximo en algún sentido estadístico.

$$e(k) = d(k) - \hat{d}(k) \quad (2.11)$$

La solución para el filtrado se presenta en el capítulo 3 a partir del criterio de los mínimos cuadrados, mientras que en la sección 2.4 se presenta la solución bajo el criterio del error cuadrático promedio, **MSE**.

¹ ε_p^f define el error de predicción de igual forma que $e(k)$, sin embargo en la deducción de los algoritmos se utiliza la notación ε_p^f donde p indica el orden del predictor y f indica que se trata de predicción forward.

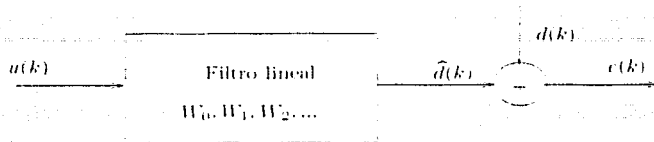


Figura 2.6: Problema fundamental del filtrado.

2.4. El filtro de Wiener

El filtro de Wiener representa una solución para el problema de filtrado basada en el criterio del error cuadrático promedio y es la base para el diseño de filtros adaptables, pues aunque Wiener plantea el problema para un filtro fijo, considerando que se dispone de las señal de entrada para todo tiempo, en el filtro de Wiener tanto la entrada al filtro como la respuesta deseada son estacionarias en sentido amplio con media cero, por lo que el error también es estacionario en sentido amplio y con media cero. Para un filtro adaptable se desea llegar a la solución de Wiener de manera recursiva conforme se obtienen los datos de la señal de entrada.

El filtro puede ser a respuesta impulsional finita (FIR) ó a respuesta impulsional infinita (IIR) dependiendo de consideraciones prácticas, sin embargo para aplicaciones de filtrado adaptable es más frecuente el uso de filtros FIR, que por definición son estables, es decir no presentan retroalimentación que los pueda llevar a alguna situación de inestabilidad. Por otra parte, en el problema de filtrado adaptable se presentan por definición lazos de retroalimentación para actualizar los parámetros, esto ocasiona que de manera inherente se presenten problemas de estabilidad y convergencia en los algoritmos. Por esta razón, se utilizan más frecuentemente filtros FIR que aseguran la estabilidad del filtro, reduciéndose el problema de estabilidad del algoritmo.

El criterio de optimización del error cuadrático promedio como veremos más adelante, nos indica una relación de la función a optimizar que a su vez involucra en forma cuadrática a los parámetros óptimos del filtro. Además se puede demostrar que la solución óptima para el filtro de Wiener se presenta cuando el error de estimación y la señal de salida del entrada son ortogonales [11].

$$E[u(k-n)c_{optimo}^*(k)] = 0, \quad \forall n = 0, 1, 2, \dots \quad (2.12)$$

ECUACIÓN DE WIENER-HOPF

La salida del filtro la podemos expresar como la convolución de la señal de entrada con los parámetros del filtro, esto es:

$$\hat{d}(k) = \sum_{n=0}^{\infty} \omega(n)^* u(k-n), \quad \forall k = 0, 1, 2, \dots \quad (2.13)$$

donde el asterisco denota complejo conjugado para el caso de un filtro con coeficientes complejos. Basados en el principio de ortogonalidad (ec. 2.12) podemos escribir:

$$E[u(k-n)(d^*(k) - \sum_{i=0}^{\infty} \omega_{optimo}^*(i)u(k-i))] = 0, \quad \forall n = 0, 1, 2, \dots \quad (2.14)$$

Realizando los productos e intercambiando el operador esperanza y la sumatoria (por ser operadores lineales):

$$\sum_{i=0}^{\infty} \omega_{optimo}^*(i)E[u(k-n)u(k-i)] = E[u(k-n)d^*(k)], \quad \forall n = 0, 1, 2, \dots \quad (2.15)$$

donde:

$E[u(k-n)u^*(k-i)] = R(i-n)$ es la autocorrelación de la señal de entrada del filtro valuada en $i-n$.

$E[u(k-n)d^*(k)] = r(-n)$ es la correlación cruzada entre la señal deseada y la entrada del filtro valuada en $-n$.

$\omega_{optimo}(i)$ son los parámetros óptimos del filtro en el sentido del error cuadrático promedio.

Podemos reescribir la ecuación 2.15 en términos de las funciones de correlación:

$$\sum_{i=0}^{\infty} \omega_{optimo}(i)R(i-n) = r(-n) \quad \forall n = 0, 1, 2, \dots \quad (2.16)$$

La ecuación 2.16 se conoce como ecuación de Wiener-Hopf [12] y representa un sistema de ecuaciones simultáneas para obtener los parámetros óptimos.

Como se mencionó al principio de esta sección, es de especial interés obtener la solución para un filtro transversal del tipo FIR. La estructura para un filtro transversal FIR se muestra en la figura 2.7.

En este caso la respuesta al impulso está limitada a las etapas de ganancia del filtro $\omega_0, \omega_1, \dots, \omega_{M-1}$, por lo tanto la ecuación de Wiener-Hopf se puede escribir de la siguiente manera:

$$\sum_{i=0}^{M-1} \omega_i(i)R(i-n) = r(-n) \quad \forall n = 0, 1, 2, \dots, M-1 \quad (2.17)$$



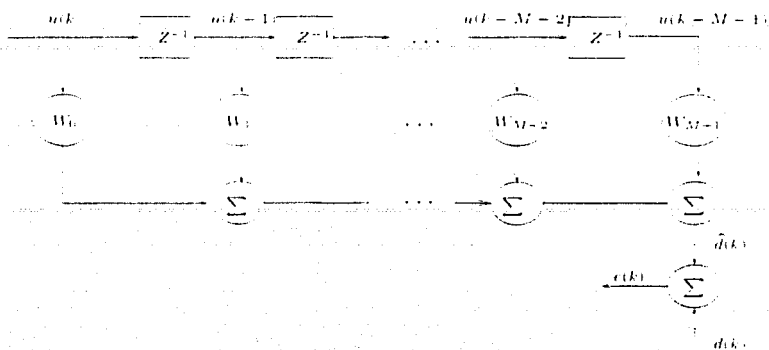


Figura 2.7: Filtro FIR transversal.

Entonces la matriz de correlación de la señal de entrada a las etapas del filtro debe ser de dimensión $M \times M$ y esta dada por:

$$R = E[U^*(k)U^T(k)]$$

Donde $U^*(k)$ es el vector de entrada a las etapas del filtro y es igual a: $U^*(k) = [u(k), u(k-1), \dots, u(k-M+1)]^T$, por lo tanto la matriz de correlación tiene la forma:

$$R = \begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r^*(1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \dots & r(0) \end{bmatrix} \quad (2.18)$$

Y el vector de correlación cruzada entre la señal de entrada y la señal deseada:

$$r = E[U^*(k)d^*(k)]$$

Es decir:

$$r = [r(0), r(-1), \dots, r(1-M)]^T \quad (2.19)$$

La ecuación de Wiener-Hopf para un filtro transversal se reduce la ecuación matricial:

$$RW_o = r \quad (2.20)$$

Por lo tanto para encontrar la solución de los parámetros óptimos solo requerimos invertir la matriz de autocorrelación R .

$$W_o = R^{-1}r \quad (2.21)$$

De la ecuación 2.21 podemos observar que para obtener los parámetros óptimos de un filtro de Wiener solo requerimos conocer la matriz de autocorrelación de la señal de entrada y el vector de correlación de la señal deseada con la señal de entrada. Hemos obtenido la ecuación de Wiener-Hopf a partir del principio de ortogonalidad, existe otra manera de obtener esta ecuación, esta es observando la dependencia de la función costo con respecto a los parámetros del filtro y haciendo uso de las propiedades de correlación para procesos estacionarios en sentido amplio, es decir suponiendo que tanto la señal de entrada como la deseada son estacionarias, además se deben de considerar con media cero. [12].

ERROR ÓPTIMO E INTERPRETACIÓN GEOMÉTRICA DEL ERROR.

Es de interés conocer el valor óptimo del error, ya que de esta manera podemos comparar diferentes algoritmos y encontrar cual es el que presenta un mejor desempeño con respecto al error al que converge y de esta forma evaluar nuestro filtro. Para obtener el valor óptimo hacemos uso de la definición de la función costo¹ y de la definición del error obteniendo:

$$J(w) = E[d^2(k)] - 2 \sum_{n=0}^{\infty} w(n) E[d(k)u(k-n)] + \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} w(n)w(m) E[u(k-n)u(k-m)] \quad (2.22)$$

Asumiendo que la señal de entrada y la señal deseada son conjuntamente estacionarias en sentido amplio, lo cual significa que sus funciones de correlación cruzada son independientes de n , podemos escribir la ecuación 2.22 en forma matricial de la siguiente manera:

$$J(w) = \sigma_d^2 - W^* r - r^* W + W^* R W \quad (2.23)$$

El valor del error cuadrático promedio óptimo es el mínimo posible pues este representa la energía de la señal de error. Para obtener el error óptimo debemos sustituir en la ecuación 2.23 los parámetros óptimos dados por la ecuación de Wiener-Hopf, dando como resultado:

$$J(w)_{min} = \sigma_d^2 - r^* R^{-1} r \quad (2.24)$$

Por lo tanto, para cuantificar el error necesitamos conocer:

1) El valor cuadrático promedio o varianza de la respuesta deseada, $r_d(0) = \sigma_d^2$.

¹Llamamos función costo $J(w) = E\{e(k)^2\}$ a la función a optimizar en el sentido del error cuadrático promedio.

2) La correlación cruzada entre la señal deseada y la señal de entrada al filtro, $r(-n)$.

3) La inversa de la matriz de correlación de los datos de entrada R^{-1} .

Para obtener una interpretación geométrica del error podemos escribir la función costo de la forma:

$$J(W) = J_{min} + (W - W_{optimo})^* R (W - W_{optimo})$$

Esta última ecuación nos permite observar la dependencia en forma cuadrática de la función costo con respecto a los parámetros del filtro. La superficie que se muestra en la figura 2.8 muestra la superficie del error para un filtro de orden 2, donde las coordenadas en el plano (x, y) corresponden a los parámetros del filtro. Cuando el filtro converge al valor óptimo de sus parámetros, se obtiene el mínimo valor para la función costo el cual corresponde al punto mínimo en la superficie de la figura 2.8.

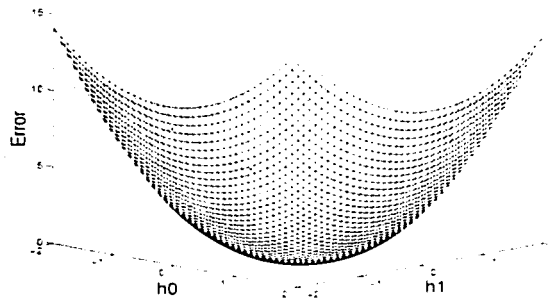


Figura 2.8: Superficie del error

2.5. El filtro de Kalman

El filtro de Kalman es un proceso de estimación de estado óptimo aplicado a sistemas dinámicos que involucren perturbaciones aleatorias. Este nos lleva a la obtención de algoritmos recursivos lineales sin sesgo y varianza de error mínima para estimar de manera óptima el estado desconocido de un sistema dinámico a partir de la medición de datos ruidosos. Ha sido usado en muchas áreas de la industria tales como sistemas de rastreo por video y laser, estimación de trayectorias de misiles, radares,

etc., [10].

Como hemos dicho, el filtro de Kalman trabaja con sistemas representados mediante variables de estado, es decir en lugar de que su objetivo sea estimar los parámetros del filtro, este se ocupa de calcular el estado actual del filtro. A continuación se realizara un análisis del filtro de Kalman.

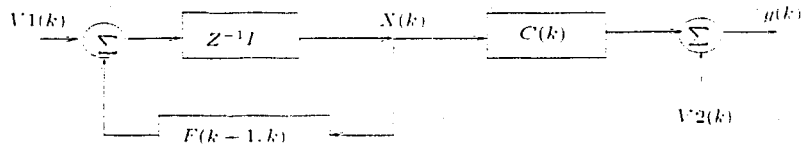


Figura 2.9: Problema fundamental de un filtro Kalman.

Considerando el sistema descrito por el diagrama en la figura 2.9, el vector de estado denotado por $X(k)$ se interpreta como un grupo de cantidades suficientes para describir únicamente el comportamiento dinámico no forzado del sistema, este se asume de dimensión M desconocido. Para estimar $X(k)$ nos basamos en la observación de un grupo de muestras contenidas en el vector $y(k)$ el cual se asume de dimensión N . Este sistema involucra básicamente dos pares de ecuaciones:

1. Una ecuación del proceso dada por:

$$X(k+1) = F(k-1, k)X(k) + V_1(k) \quad (2.25)$$

donde:

$F(k+1, k)$ es una matriz conocida de transición de estado de dimensión $M \times M$, relacionando el estado del sistema con los tiempos $k+1$ y k .

$V_1(k)$ es un proceso de ruido blanco, modelado con media cero y matriz de correlación definida por:

$$E[V_1(k)V_1^*(n)] = \begin{cases} Q_1(k); & k = n. \\ 0; & k \neq n. \end{cases}$$

2. Una ecuación de medición descrita por el vector de observaciones como:

$$y(k) = C(k)X(k) + V_2(k) \quad (2.26)$$

donde:

$C(k)$ Es la matriz de medición de $N \times M$.

$V_2(k)$, Vector de ruido de medición de dimensión $N \times 1$, el cual es tomado como ruido blanco con media cero y matriz de correlación:

$$E[V_2(k)V_2^*(n)] = \begin{cases} Q_2(k): & k = n. \\ 0: & k \neq n. \end{cases}$$

Asumimos que $X(0)$, el valor del estado inicial, es descorrelacionado tanto con $V_1(k)$ como con $V_2(k) \forall k \geq 0$. También consideramos que los vectores de ruido son estadísticamente independientes, esto es:

$$E[V_1(k)V_2^*(n)] = 0 \quad \forall k, n.$$

El problema del filtrado de Kalman consiste en obtener el estimador de los mínimos cuadrados promedio de $X(i)$, mediante el uso de los datos observados, vectores $y(1), y(2), \dots, y(k)$, para cada $n \geq 1$. Entonces, se pueden apreciar tres situaciones interesantes, esto es: si $i = k$ el problema se denomina filtrado, si $i > k$ predicción y suavizado si $1 \leq i < k$. A menudo se utiliza la aproximación de inovaciones para resolver el problema. [13].

El proceso de inovación.

En este método definimos el vector $\hat{y}(k|y_{k-1})$ que denota el estimador de los mínimos cuadrados promedio de la señal $y(k)$ en el tiempo k , dados los valores pasados de la señal observada a partir de $k = 1$ y hasta $k = k - 1$, estos valores contenidos en los vectores $y(1), y(2), \dots, y(k - 1)$ se representan por el vector y_{k-1} . Entonces podemos definir el proceso de inovación asociado al vector de muestras $y(k)$ como:

$$\alpha(k) = y(k) - \hat{y}(k|y_{k-1}); \quad k = 1, 2, \dots \quad (2.27)$$

El vector $\alpha(k)$ de $M \times 1$ representa la nueva información en los datos observados $y(k)$ y presenta las siguientes propiedades. [13]:

1. El proceso de inovación $\alpha(k)$, asociado a los datos observados al tiempo n , es ortogonal a todas las observaciones pasadas:

$$E[\alpha(k)y^*(n)] = 0; \quad 1 \leq n \leq k - 1 \quad (2.28)$$

2. El proceso de inovación consiste en una secuencia de vectores de variables aleatorias que son ortogonales entre sí:

$$E[\alpha(k)\alpha^*(n)] = 0; \quad 1 \leq n \leq k - 1 \quad (2.29)$$

3. Existe una correspondencia uno a uno entre la secuencia de variables aleatorias $y(1), y(2), \dots, y(k)$ representando las muestras observadas de datos y la secuencia de variables aleatorias $\alpha(1), \alpha(2), \dots, \alpha(k)$ representando el proceso de inovación. Dada esta correspondencia, una de estas secuencias puede ser obtenida a partir de la otra

por medio de operadores lineales estables sin pérdida de información. Esto es:

$$y(1), y(2), \dots, y(k) \iff \alpha(1), \alpha(2), \dots, \alpha(k)$$

La matriz de correlación del proceso de innovación se define como, [13]:

$$R(k) = C'(k)K(k, k-1)C''(k) + Q_2(k) \quad (2.30)$$

Donde:

$Q_2(k)$ es la matriz de correlación del vector de ruido $V_2(k)$

$K(k, k-1)$ Es la matriz de correlación de estado del error de predicción.

$$K(k, k-1) = E[\epsilon(k, k-1)\epsilon^*(k, k-1)]$$

$\epsilon(k, k-1)$ es el vector del error de predicción de estado.

Estimación del estado usando el proceso de innovación

El problema consiste en obtener el estimador de los mínimos cuadrados promedio del estado $X(i)$. Este estimador lo definimos como:

$$\hat{x}(i|y_k) = \sum_{n=1}^k B_i(n)\alpha(n) \quad (2.31)$$

Donde, $B_i(n)$, $k = 1, \dots, k$ es un grupo de matrices de $M \times N$ que se deben determinar. De acuerdo con el principio de ortogonalidad, el vector del error predictor de estado es ortogonal con el proceso de innovación:

$$E[\epsilon(i, k)\alpha^*(m)] = E[X(i) - \hat{x}(i|y_k)]\alpha^*(m) = 0, m = 1, 2, \dots, k. \quad (2.32)$$

Sustituyendo la ecuación 2.31 en la ecuación 2.32 y utilizando la ecuación 2.29:

$$E[X(i)\alpha^*(m)] = B_i(m)E[\alpha(m)\alpha^*(m)] = B_i(m)R(m) \quad (2.33)$$

De la ecuación 2.33:

$$B_i(m) = E[\alpha(m)\alpha^*(m)]R^{-1}(m) \quad (2.34)$$

Finalmente substituyendo la ecuación 2.34 en la ecuación 2.31 para $i = k+1$ y definiendo $G(k) = E[X(k+1)\alpha^*(k)]R^{-1}(k)$, podemos escribir el estimador como:

$$\hat{x}(k+1|y_k) = F(k+1, k)\hat{x}(k|y_{k-1}) + G(k)\alpha(k) \quad (2.35)$$

La ecuación 2.35 muestra que podemos calcular el estimador de los mínimos cuadrados promedio $\hat{x}(k+1|y_k)$ del estado del sistema sumando al estimador previo pre-multiplicado por la matriz de transición $F(k+1, k)$ un término de corrección igual a

$G(k)\alpha(k)$.

A $G(k)$ se le denomina la ganancia de Kalman y puede redefinirse de la siguiente forma. [13]:

$$G(k) = F(k-1, k)K(k, k-1)C^H(k)R^{-1}(k)$$

En general, el hecho de poder calcular el estimador de los mínimos cuadrados promedio actual del sistema en función del estimador anterior y un factor de corrección, es la característica más importante de la teoría de Kalman, dando como consecuencia que los algoritmos basados en esta teoría tengan convergencias más rápidas que los basados en la teoría de Wiener en donde se requiere acumular las muestras pasadas de la señal para obtener el estimado actual.

TESIS CON
FALLA DE ORIGEN

2.6. Conclusiones

En este capítulo hemos introducido una serie de elementos que nos permitirán obtener una mejor comprensión del problema de filtrado adaptable para la igualación de canal. Conceptos como el modelado de procesos estocásticos y el de la predicción lineal son de gran importancia en el problema de filtrado adaptable. En particular, la comprensión del problema de la predicción lineal es fundamental para el filtrado, ya que cuando hablamos de filtrado, la única diferencia es la señal de referencia que se usa.

Por otra parte, hemos comentado acerca de las características de los algoritmos de filtrado, las cuales no siempre se pueden tener para un mismo algoritmo, sin embargo; dependiendo de la aplicación, algunas características serán más importantes que otras. Así mismo, hemos estudiado el filtro de Wiener basado en el criterio del error cuadrático promedio, aunque este filtro no es adaptable; en el caso de los filtros adaptables se busca llegar a la solución de Wiener de manera iterativa conforme se recibe la información.

Por último, en la sección 2.5 hemos incluido el problema del filtro de Kalman, un estudio en variables de estado, y hemos obtenido su solución óptima por medio del proceso de innovación.

Capítulo 3

ALGORITMOS DE FILTRADO ADAPTABLE PARA LA IGUALACIÓN DE CANAL

3.1. Introducción

En este capítulo se estudiarán dos de los criterios de optimización más utilizados para el desarrollo de algoritmos de filtrado adaptable. En primer lugar, el criterio del error cuadrático promedio (MSE) utilizado en los algoritmos de paso descendente y LMS clásico, este último como una aproximación a una situación real en donde se desconocen tanto la matriz de correlación del proceso como el vector de correlación cruzada. En segundo lugar, tenemos a los algoritmos basados en los mínimos cuadrados LS. Se presentará la deducción para el algoritmo clásico, CLS, con estructura transversal. Además, como una alternativa para disminuir el tiempo de cálculo de los parámetros del filtro se obtendrá el algoritmo rápido de Kalman.

Finalmente, con el objetivo de observar las características de cada uno de los algoritmos, éstos se implementarán en un esquema de igualación lineal en aritmética de punto flotante con precisión doble.[13].

3.2. Algoritmos basados en el error cuadrático promedio

3.2.1. El método de paso descendente

Para estudiar este método consideramos el filtro de la figura 3.1, con vector de entrada $U(k)$ representando a las muestras $u(k), u(k-1), \dots, u(k-p+1)$ y vector de

ganancias $W(k)$ representando los parámetros del filtro $w_0(k), w_1(k), \dots, w_{p-1}(k)$.

$U(k)$ se compone de las muestra de un proceso estocástico WSS (que es estacionario en sentido amplio), de media cero con su correspondiente matriz de correlación R . El filtro es provisto de una respuesta deseada $d(k)$, también conocida como secuencia de entrenamiento, con características estadísticas iguales a la señal de entrada.

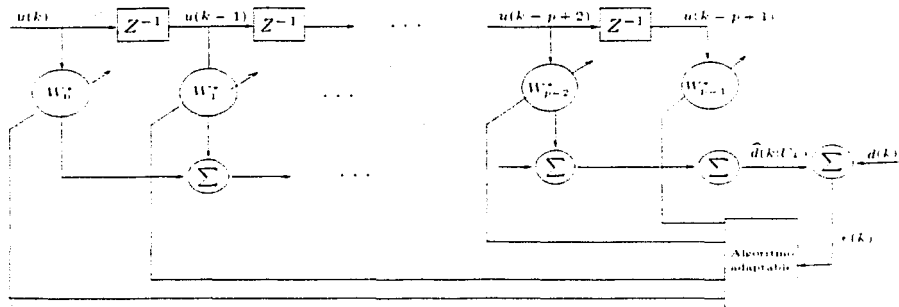


Figura 3.1: Filtro transversal adaptable.

El estimado de la respuesta deseada se denota por $\hat{d}(k|U_k)$, donde $U_k = U(k)$.

Si comparamos el estimado con la respuesta deseada $d(k)$ podemos obtener el error de estimación dado por:

$$e(k) = d(k) - \hat{d}(k|U_k) = d(k) - W^*(k)U'(k) \quad (3.1)$$

Como se demostró en el capítulo 2, si $U'(k)$ y la respuesta deseada $d(k)$ son conjuntamente estacionarios, entonces la función costo ($J(k)$) al tiempo k es una función cuadrática del vector de ganancias y podemos escribir:

$$J(k) = \sigma_d^2 - W^*(k)r - r^*W(k) + W^*(k)RW(k) \quad (3.2)$$

Donde:

σ_d^2 es la variancia de la respuesta deseada $d(k)$.

r es el vector de correlación cruzada entre $U'(k)$ y la respuesta deseada.

R es la matriz de autocorrelación de $U(k)$.

TESIS CON
FALLA DE ORIGEN

La ecuación 3.2 define el error cuadrático promedio que resultaría si los parámetros del filtro $W(k)$ fueran fijados. Dado que $W(k)$ varía con el tiempo, el error medio cuadrático lo hace de igual manera. La variación del error cuadrático promedio $J(k)$ con respecto al tiempo significa que el error de estimación (proceso $e(k)$) es no estacionario.

Como se vio en el capítulo 2 la superficie que representa $J(k)$ es en forma de un tazón con un mínimo global y se denomina para el caso de un filtro adaptable *superficie de desempeño del error del filtro adaptable*, [13]. El proceso adaptable tiene la tarea de buscar continuamente el punto mínimo de esta superficie, en el cual el vector de ganancias $W(k)$ toma su valor óptimo W_0 que se define por la ecuación de Wiener-Hopf (ecuación 2.20).

Sustituyendo la solución de Wiener en la ecuación 3.2 podemos obtener el mínimo error cuadrático promedio definido como:

$$J_{min} = \sigma_d^2 - r^* W_0 \quad (3.3)$$

En la siguiente sección desarrollaremos las ecuaciones fundamentales para el algoritmo de paso descendente.

El algoritmo de paso descendente

El requisito que un filtro adaptable debe satisfacer es el de encontrar una solución para su vector de ganancias ó vector de parámetros que satisfaga la ecuación de Wiener-Hopf.

Una forma de encontrar esta solución es resolver el sistema de ecuaciones dado en forma analítica por 2.20. En general, este método es sencillo, sin embargo presenta serias dificultades de cómputo, especialmente cuando el filtro contiene muchas etapas y la tasa de llegada de los datos es elevada.

Un procedimiento alternativo es utilizar el método de paso descendente, uno de los métodos más antiguos de optimización. El algoritmo de paso descendente se puede resumir en los siguientes puntos:

1. Se inicia con un valor inicial $W(0)$, el cual provee una suposición inicial para el punto mínimo en la superficie del error. A menos que se disponga de algún conocimiento previo, usualmente es puesto a cero.
2. Usando la suposición inicial, se calcula el vector gradiente como la derivada del error cuadrático promedio $J(k)$ evaluado con respecto a la parte real e imaginaria del vector de parámetros $W(k)$ al tiempo k (k -ésima iteración).

3. Se calcula la siguiente suposición para $W(k)$ haciendo un cambio en el valor actual supuesto, en una dirección opuesta al vector gradiente,

4. Regresamos al paso dos y repetimos el proceso.

Se puede intuir que las correcciones sucesivas a los parámetros en la dirección negativa del vector gradiente debe eventualmente dirigirse al error cuadrático promedio mínimo J_{min} y en tal punto el vector de parámetros asume su valor óptimo W_0 .

Si $\nabla J(k)$ denota el valor del vector gradiente al tiempo k y $W(k)$ denota el valor del vector de parámetros al tiempo k . De acuerdo al método de paso descendente el valor actualizado del vector de parámetros al tiempo $k+1$ es calculado como:

$$W(k+1) = W(k) + \frac{1}{2}\mu[-\nabla J(k)] \quad (3.4)$$

Donde μ es el parámetro de paso, constante real positiva, y el factor de $\frac{1}{2}$ se utiliza únicamente para cancelar un factor de 2 que aparece en el cálculo del gradiente $\nabla J(k)$ como se muestra a continuación.

$$\nabla J(k) = \begin{bmatrix} \frac{\partial J(k)}{\partial a_0(k)} + j \frac{\partial J(k)}{\partial b_0(k)} \\ \frac{\partial J(k)}{\partial a_1(k)} + j \frac{\partial J(k)}{\partial b_1(k)} \\ \vdots \\ \frac{\partial J(k)}{\partial a_{p-1}(k)} + j \frac{\partial J(k)}{\partial b_{p-1}(k)} \end{bmatrix} = -2r + 2RW(k) \quad (3.5)$$

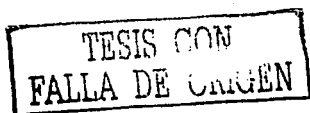
Donde el vector columna $\frac{\partial J(k)}{\partial a_m(k)} + j \frac{\partial J(k)}{\partial b_m(k)}$ corresponde a las derivadas parciales de la función costo $J(k)$ con respecto a la parte real $a_m(k)$ y a la parte imaginaria $b_m(k)$ de la m -ésima etapa de ganancia $W_m(k)$. $\forall m = 0, 1, 2, \dots, p-1$.

Para la aplicación del algoritmo de paso descendente, asumimos que en la ecuación 3.5 la matriz y el vector de correlación R y r son conocidos de tal forma que se puede calcular el vector gradiente $\nabla J(k)$ para un valor dado del vector de parámetros $W(k)$. Sustituyendo la ecuación 3.5 en la ecuación 3.4 encontramos la actualización de los parámetros en términos de la matriz R y el vector r , la cual describe el algoritmo de paso descendente:

$$W(k+1) = W(k) + \mu[r - RW(k)]; \quad k = 0, 1, 2, \dots \quad (3.6)$$

En la ecuación 3.6, μ controla el tamaño de la corrección aplicado al vector de parámetros $W(k)$ de una iteración a otra. Por lo tanto, a menudo nos referimos a μ como el parámetro del tamaño del paso.

La corrección aplicada al vector de parámetros al tiempo $k+1$ es igual a $\mu[r - RW(k)]$, esta corrección se puede expresar como μ veces el valor esperado del producto interno



del vector de entrada $U(k)$ y el error de estimación $e(k)$ [13]. para hacerlo se puede utilizar el banco de correladores de la figura 3.2. En la figura podemos observar como en cada paso se realiza una correlación y se estima el error de modo que es posible estimar el vector de corrección a aplicar en el algoritmo de paso descendente.

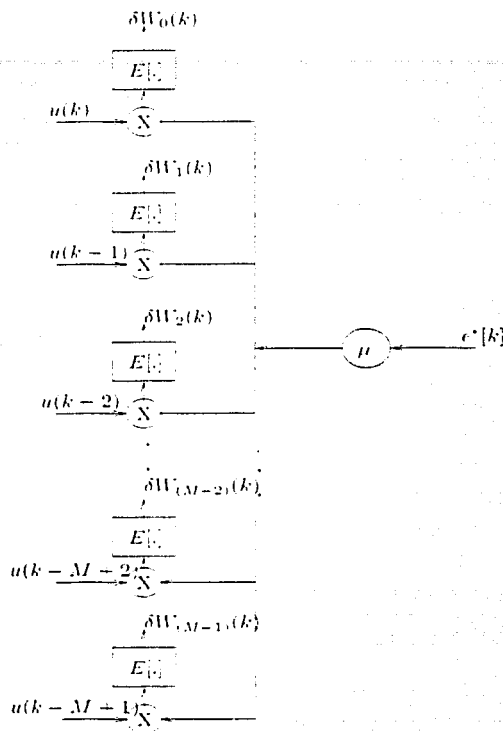


Figura 3.2: Banco de correladores para estimar la corrección en el algoritmo de paso descendente.

Dado que el algoritmo de paso descendente, como todos los algoritmos de filtrado adaptable, involucra la presencia de retroalimentación, éste está sujeto a la posibilidad de volverse inestable.

La estabilidad de los algoritmos es determinada por dos factores:

1. El parámetro de paso, μ .
2. La matriz de correlación R del vector de entrada en las etapas, $U(k)$.

La condición para la estabilidad y convergencia del algoritmo esta dada por, [13].:

$$0 < \mu < \frac{2}{\lambda_{max}}; \quad (3.7)$$

donde λ_{max} es el máximo valor característico de la matriz de correlación.

3.2.2. Algoritmo LMS transversal

Widrow y Hoff desarrollaron en 1960 este algoritmo utilizado ampliamente en la actualidad, cuya diferencia básica con el algoritmo de paso descendente es que, este se basa en el cálculo de un gradiente estocástico, mientras que el de paso descendente, en el cálculo de un gradiente determinístico para entradas estocásticas. Una característica significativa del algoritmo LMS es su simplicidad al no requerir ningún tipo de inversión de matrices.

GENERALIDADES DEL ALGORITMO LMS

Consiste en dos procesos básicos:

1. Un proceso de filtrado en el cual se calcula la salida del filtro transversal que se produce por un conjunto de muestras en la entrada y se genera un error de estimación comparando esta salida con una respuesta deseada.
2. Un proceso adaptable en el cual se involucra un ajuste automático de las ganancias del filtro de acuerdo al error de estimación.

El valor calculado para las ganancias $\hat{W}(k)$ utilizando el algoritmo LMS representa un estimado cuyo valor esperado se aproxima a la solución de Wiener, W_0 , para un proceso WSS, conforme el número de iteraciones se aproxima a infinito.

Durante el proceso de filtrado, se suministra la respuesta deseada $d(k)$ para ser procesada junto con el vector de entrada $u(k)$. Dada esta entrada, el filtro produce una salida $\hat{d}(k|U_k)$ utilizada como un estimado de la señal deseada y definiendo un error de estimación $e(k)$ como la diferencia entre la respuesta deseada y la salida actual del filtro, el error de estimación $e(k)$ y el vector de entrada $u(k)$ se aplican al

mecanismo de control para realizar la retroalimentación sobre las ganancias.

En la figura 3.3, el algoritmo adaptable se obtiene mediante el producto interno del error de estimación y el vector de entrada $u(k)$, obteniendo la corrección $\delta \hat{w}_k(k)$ para la ganancia $\hat{w}_k(k)$ en cada iteración. La diferencia con el algoritmo de paso descendente es que el algoritmo LMS no obtiene el valor esperado del producto interno, figura 3.2. En consecuencia, el cálculo recursivo de cada etapa de ganancias sufre de un ruido de gradiente.

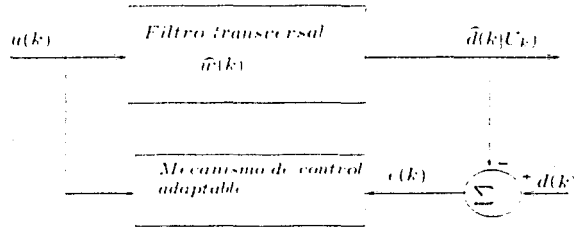


Figura 3.3: Esquema de igualación del algoritmo LMS.

Suponiendo que la señal de entrada $u(k)$ y la respuesta deseada $d(k)$ se derivan de un proceso conjuntamente estacionario en sentido amplio, el algoritmo de paso descendente calcula la ganancia $w(k)$ que se mueve hacia el mínimo de la superficie del error a lo largo de una trayectoria determinística, terminando en la solución de Wiener W_0 . Por su parte cuando hablamos del algoritmo LMS nos referimos al cálculo de un gradiente estocástico, dado el ruido de gradiente involucrado, por esta razón el comportamiento no es del todo igual al del algoritmo de paso descendente, como se verá a continuación.

En lugar de terminar en la solución de Wiener, el vector de ganancia $\hat{W}(k)$ calculado por el algoritmo LMS realiza un movimiento aleatorio alrededor del punto mínimo de la superficie del error. Al igual que en el algoritmo de paso descendente, la presencia de retroalimentación involucra el riesgo de inestabilidad buscando siempre que $J(k) \rightarrow J_{min}$ conforme $k \rightarrow \infty$.

Al igual que en el algoritmo de paso descendente el error cuadrático promedio $J(k)$ debe tender a una constante, para lograrlo se debe escoger μ de tal forma que se satisfagan con ciertas condiciones relacionadas con la estructura de la matriz R . [13].

OBTENCIÓN DEL ALGORITMO LMS

En la práctica no es posible realizar una medición exacta del vector gradiente $\nabla J(k)$ en cada iteración, por lo tanto; aún utilizando el algoritmo de paso descendente no es posible converger a la solución de Wiener, esto se debe a que se requeriría de un conocimiento previo de la matriz de autocorrelación R y del vector de correlación cruzada r . Por esta razón el vector gradiente debe ser estimado de los datos disponibles y para hacerlo debemos estimar la matriz R y el vector r . Como vimos en el algoritmo de paso descendente el gradiente está dado por la ecuación 3.5:

$$\nabla J(k) = -2r + 2RW(k) \quad (3.8)$$

Una elección simple para estimar R y r es utilizar estimadores instantáneos basados en los valores actuales de $u(k)$ y de la señal deseada $d(k)$ en cada iteración, esto es:

$$\hat{R}(k) = U(k)U^*(k) \quad (3.9)$$

$$\hat{r}(k) = U(k)d^*(k) \quad (3.10)$$

Sustituyendo las ecuaciones 3.9 y 3.10 en la ecuación 3.8 obtenemos un estimado instantáneo del vector gradiente:

$$\hat{\nabla} J(k) = -2U(k)d^*(k) + 2U(k)U^*(k)\hat{W}(k) \quad (3.11)$$

En general, este estimador es sesgado pues depende de los valores del vector de entrada $U(k)$. Teniendo el estimado para el gradiente podemos obtener el vector de parámetros estimado de forma recursiva sustituyendo la ecuación 3.11 en 3.6, obteniendo:

$$\hat{W}(k+1) = \hat{W}(k) - \mu U^H(k)[d^*(k) - U^H(k)\hat{W}(k)] \quad (3.12)$$

El algoritmo LMS puede escribirse como se muestra en el cuadro. 3.1

El algoritmo LMS requiere de $2P+1$ multiplicaciones y $2P$ sumas por iteración, donde P es el orden del filtro. Los estimados instantáneos de R y r tienen varianzas relativamente grandes, por lo que podría pensarse que es incapaz de tener un buen rendimiento, sin embargo; este algoritmo es por naturaleza recursivo dando como resultado que el algoritmo por sí solo promedia efectivamente estos estimados a lo largo del proceso de adaptación.

TESIS CON
FALLA DE ORIGEN

ALGORITMO DE FILTRADO LMS

Condiciones iniciales

$$\hat{W}(k-1) = 0$$

Filtrado

$$\hat{d}(k|U_k) = W^H(k-1)U(k)$$

$$e(k) = d(k) - \hat{d}(k|U_k)$$

$$\hat{W}(k) = \hat{W}(k-1) - \mu U(k)e^*(k)$$

Cuadro 3.1: Algoritmo de Filtrado LMS

COMPARACIÓN DEL ALGORITMO LMS CON EL DE PASO DESCENDENTE

Tanto el algoritmo de paso descendente como el algoritmo LMS son diseñados para entradas estocásticas estacionarias en sentido amplio.

De manera ideal, el mínimo error cuadrático promedio J_{min} se logra cuando el vector de coeficientes $\hat{W}(k)$ del filtro se aproxima al valor óptimo W_0 definido por la ecuación de Wiener-Hopf. El algoritmo de paso descendente alcanza esta condición ideal conforme el número de iteraciones se aproxima a infinito, este último tiene la capacidad de hacerlo porque realiza una medición exacta del vector gradiente en cada iteración, por otra parte, el algoritmo LMS depende de un estimado ruidoso para el vector gradiente y por tanto el vector de ganancias estimado $\hat{W}(k)$ solo se aproxima al valor óptimo, como consecuencia, después de un número grande de iteraciones el error cuadrático promedio obtenido $J(\infty)$ es mayor que J_{min} .

3.3. Algoritmos basados en los mínimos cuadrados

Los algoritmos que se basan en el criterio del error cuadrático promedio restringen las señales de entrada a ser estacionarias en sentido amplio además de que convergen a la solución óptima muy lentamente, por esta razón es de interés el estudio de los algoritmos basados en los mínimos cuadrados dado que estos se pueden utilizar para todo tipo de señales además de converger mucho más rápido a la solución óptima. En esta sección se deduce el algoritmo clásico RLS, conocido como CLS, basado en el criterio de los mínimos cuadrados así como su versión rápida conocida como filtro

Rápido de Kalman (Fast Kalman).

3.3.1. El algoritmo RLS transversal

Se tiene la señal $\hat{d}(k|U_k)$ que en el caso más general es un vector complejo de dimensión $N \times 1$. Esta señal es la salida del filtro que se muestra en la figura 2.6, cuya entrada es la señal $u(k)$ que también en el caso más general es un vector complejo de dimensión $M \times 1$.

Entonces:

$$\hat{d}(k|U_k) = - \sum_{j=1}^k W_j^T(k) u(k-j-1) \quad (3.13)$$

Retomando la ecuación 2.11 que nos define el error de filtrado podemos escribir de manera vectorial:

$$\varepsilon_p(k) = d(k) + W_p^T(k) U_p(k) \quad (3.14)$$

EL CRITERIO DE LOS MÍNIMOS CUADRADOS

El criterio de los mínimos cuadrados consiste en minimizar la función costo definida como:

$$E_p(k) = \sum_{j=N}^k \varepsilon_p(j) \varepsilon_p^T(j) \quad (3.15)$$

Los límites de la sumatoria nos definen una ventana de análisis. En este caso consideraremos el caso prewindowed donde se consideran que los datos de entrada para $j < N$ son cero, lo cual equivale a limitar la sumatoria en el intervalo de $[1, k]$.

Sustituyendo la ecuación 3.14 en la ecuación 3.15:

$$E_p(k) = Tr \sum_{j=1}^k \lambda^{k-j} \{ [d(j) + W_p^T(k) U_p(j)] [d^T(j) + U_p^T(j) W_p(k)] \}$$

$$E_p(k) = Tr \sum_{j=1}^k \lambda^{k-j} \{ d(j) d^T(j) + W_p^T(k) U_p(j) d^T(j) + d(j) U_p^T(j) W_p(k) + W_p^T(k) U_p(j) U_p^T(j) W_p(k) \}$$

En estas últimas ecuaciones se ha introducido el factor de olvido λ el cual tiene la finalidad de ponderar las muestras que salen del filtro, dándoles un mayor peso a las muestras más actuales y un menor peso a las muestras más viejas. Por esta razón se debe escoger $\lambda < 1$.

Definimos entonces:

$$\sum_{j=1}^k \lambda^{k-j} U_p(j) U_p^T(j) = R_p(k) \quad (3.16)$$

$$\sum_{j=1}^k \lambda^{k-j} U_p(j) d^T(j) = r_p(k) \quad (3.17)$$

$$\sum_{j=1}^k \lambda^{k-j} d(j) d^T(j) = r_{p0}(k) \quad (3.18)$$

A la ecuación 3.17 se le conoce como vector de covarianza mientras que a la ecuación 3.16 se le conoce como matriz de covarianza.

$$E_p(k) = r_{p0}(k) + W_p^T(k) r_p(k) + r_p^T(k) W_p(k) + W_p^T(k) R_p(k) W_p(k) \quad (3.19)$$

Para obtener los parámetros óptimos, derivamos el criterio con respecto a W_p , $\frac{\partial E_p(k)}{\partial W_p}$:

$$\frac{\partial E_p(k)}{\partial W_p} = r_p(k) + r_p^T(k) + [R_p(k) + R_p^T(k)] W_p(k)$$

Como $R_p(k)$ es simétrica:

$$R_p(k) = R_p^T(k).$$

e igualando a cero:

$$\frac{\partial E_p(k)}{\partial W_p} = 0$$

Obtenemos entonces la solución óptima:

$$R_p(k) W_p(k) = -r_p(k) \quad (3.20)$$

En esta última expresión habrá que invertir $R_p(k)$, para obtener W_p .

RECURSIVIDAD PARA $R_p(k)$ Y $r_p(k)$.

$$\begin{aligned} R_p(k) &= \sum_{j=1}^k \lambda^{k-j} U_p(j) U_p^T(j) \\ R_p(k-1) &= \sum_{j=1}^{k-1} \lambda^{k-j} U_p(j) U_p^T(j) + U_p(k) U_p^T(k) \\ &= \lambda \sum_{j=1}^{k-1} \lambda^{k-j-1} U_p(j) U_p^T(j) + U_p(k) U_p^T(k) \end{aligned}$$

$$R_p(k) = \lambda R_p(k-1) + U_p(k)U_p^T(k) \quad (3.21)$$

Similar para $r_p(k)$:

$$\begin{aligned} r_p(k) &= \sum_{j=1}^k \lambda^{k-j} U_p(j) d^T(j) \\ r_p(k-1) &= \sum_{j=1}^{k-1} \lambda^{k-j} U_p(j) d^T(j) + U_p(k) d^T(k) \\ &= \lambda \sum_{j=1}^{k-1} \lambda^{k-j-1} U_p(j) d^T(j) + U_p(k) d^T(k) \\ r_p(k) &= \lambda r_p(k-1) + U_p(k) d^T(k) \end{aligned} \quad (3.22)$$

RECURSIVIDAD PARA $W_p(k)$.

Sustituyendo 3.22 en la ecuación 3.20:

$$\begin{aligned} R_p(k)W_p(k) &= -\lambda r_p(k-1) - U_p(k) d^T(k) \\ &= \lambda R_p(k-1)W_p(k-1) - U_p(k) d^T(k) \end{aligned}$$

Si sumamos y restamos $U_p(k)U_p^T(k)W_p(k-1)$:

$$= \{\lambda R_p(k-1) + U_p(k)U_p^T(k)\}W_p(k-1) - U_p(k)\{d^T(k) + U_p^T(k)W_p(k-1)\}$$

Si definimos el error a priori, utilizando los parámetros al instante $k-1$ como:

$$e_p(k) = d(k) + W_p^T(k-1)U_p(k) \quad (3.23)$$

Entonces tenemos:

$$\begin{aligned} R_p(k)W_p(k) &= R_p(k)W_p(k-1) - U_p(k)e_p^T(k) \\ R_p(k)[W_p(k) - W_p(k-1)] &= -U_p(k)e_p^T(k) \end{aligned}$$

Despejando $W_p(k)$:

$$W_p(k) = W_p(k-1) - R_p^{-1}(k)U_p(k)e_p^T(k) \quad (3.24)$$

Donde, podemos definir a la relación:

$$K_p(k) = -R_p^{-1}(k)U_p(k) \quad (3.25)$$

A la ecuación 3.25 se le conoce como ganancia de Kalman. Podemos entonces escribir la ecuación 3.24 en términos de la ganancia de Kalman como:

$$W_p(k) = W_p(k-1) + K_p(k)u_p^T(k) \quad (3.26)$$

Como podemos observar en la ecuación anterior, para obtener los parámetros óptimos del filtro se requiere evaluar $R_p^{-1}(k)$ en cada iteración lo cual representa un gran número de operaciones. Por esta razón, a continuación se presenta un método recursivo para evaluar la inversa de la matriz de covarianza $R_p(k)$.

RECURSIVIDAD PARA $R_p^{-1}(k)$.

Para poder evaluar $R_p^{-1}(k)$ calculamos la inversa de la ecuación 3.21:

$$R_p^{-1}(k) = [\lambda R_p(k-1) + U_p(k)U_p^T(k)]^{-1}$$

Entonces aplicando el lema de inversión matricial¹,

$$R_p^{-1}(k) = \lambda^{-1}R_p^{-1}(k-1) +$$

$$\lambda^{-1}R_p^{-1}U_p(k)[1 + U_p^T(k)\lambda^{-1}R_p^{-1}(k-1)U_p(k)]^{-1}U_p^T(k)\lambda^{-1}R_p^{-1}(k-1)$$

Por último:

$$R_p^{-1}(k) = \lambda^{-1}R_p^{-1}(k-1) + \frac{\lambda^{-1}R_p^{-1}(k-1)U_p(k)U_p^T(k)R_p^{-1}(k-1)}{\lambda + U_p^T(k)R_p^{-1}(k-1)U_p(k)} \quad (3.27)$$

Con esta última expresión solo necesitamos calcular la inversa de $R_p(k)$ una sola vez y después evaluar $[R_p(k)]^{-1}$ en forma recursiva. El algoritmo de filtrado RLS clásico se da en el cuadro 3.2.

Existen otros tipos de estructuras de filtros que presentan algunas ventajas como dinámica limitada y modularidad, tales estructuras se denominan "lattice". Los algoritmos utilizados para la obtención de los parámetros de los filtros "lattice" involucran tanto recursiones en orden como en tiempo, lo que los hace más lentos, sin embargo debido a la dinámica limitada estos son más estables que los algoritmos RLS y la utilización de uno u otro tipo de estructura dependerá de la aplicación.

3.3.2. El algoritmo FRLS "Fast Kalman"

En aplicaciones en donde se requiere utilizar filtros de ordenes muy grandes, los algoritmos RLS clásicos limitan la velocidad de transmisión dado que requieren del orden de p^2 operaciones aritméticas por iteración y por tanto un gran tiempo de procesamiento para cada muestra recibida. La idea de los algoritmos rápidos (o Fast RLS, FRLS), es obtener los parámetros del filtro de una manera más eficiente, es decir:

¹Ver apéndice A.



| ALGORITMO RLS [27] | |
|--|------|
| Condiciones iniciales | |
| $R^{-1}[k-1] = I_N^{\frac{1}{\lambda}}$ | |
| $\delta = 1 \times 10^{-2}, W_p[k-1] = 0$ | |
| $U_p[k-1] = 0$ | |
| ECUACIONES | REF. |
| $e_p[k] = d[k] + W_p[k-1]U_p[k]$ | 3.23 |
| $R^{-1}[k] = \frac{1}{\lambda}R^{-1}[k-1] - \frac{\frac{1}{\lambda}R^{-1}[k-1]U_p[k]U_p[k]^TR^{-1}[k-1]}{\lambda - U_p[k]^TR^{-1}[k-1]U_p[k]}$ | 3.27 |
| $K_p(k) = -R^{-1}(k)U_p(k)$ | 3.25 |
| $W_p[k] = W_p[k-1] + K_p[k]e(k)^T$ | 3.26 |

Cuadro 3.2: Algoritmo de filtrado RLS

con el mínimo de operaciones aritméticas pero conservando las buenas propiedades de velocidad de convergencia de los algoritmos RLS clásicos. En particular, los algoritmos FRLS reducen el número de operaciones aritméticas al orden de p . Como veremos a continuación, estos algoritmos utilizan tanto parámetros de predicción como parámetros de filtrado y se basan fundamentalmente en el hecho de que podemos escribir al vector de entrada como:

$$U_{p-1}[k] = \begin{bmatrix} U_p[k] \\ u[k-p] \end{bmatrix} = \begin{bmatrix} U_p[k] & u[k] \\ U_p[k-1] & u[k-1] \end{bmatrix} \quad (3.28)$$

La ecuación 3.28 nos permite escribir la matriz de covarianza (ecuación 3.16) al orden $p+1$ como:

$$R_{p-1}(k) = \sum_{j=N}^k \lambda^{k-j} U_{p-1}(j)U_{p-1}^T(j) \quad (3.29)$$

Donde si tomamos de la ecuación 3.28:

TESIS CON
FALLA DE ORIGEN

$$U_{p+1}[k] = \begin{bmatrix} U_p[k] \\ u[k - p] \end{bmatrix}$$

y la sustituimos en 3.29 obtenemos:

$$R_{p+1}(k) = \begin{bmatrix} R_p(k) & r_p^b(k) \\ r_p^{bT}(k) & r_o^b(k) \end{bmatrix} \quad (3.30)$$

De igual forma si tomamos de la ecuación 3.28:

$$U_{p+1}[k] = \begin{bmatrix} u[k] \\ U_p[k - 1] \end{bmatrix}$$

Y la sustituimos en 3.29 obtenemos:

$$R_{p+1}(k) = \begin{bmatrix} r_o^f(k) & r_p^{fT}(k) \\ r_p^f(k) & R_p(k - 1) \end{bmatrix} \quad (3.31)$$

Tomando la ganancia de Kalman (ecuación 3.25) al orden $p + 1$:

$$K_{p+1}(k) = -R_{p+1}^{-1}(k)U_{p+1}(k) \quad (3.32)$$

En la ecuación 3.32 tenemos que evaluar de manera recursiva $R_{p+1}^{-1}(k)$, para realizar tal tarea hacemos uso del lema de inversión matricial para una matriz particionada ¹.

Si trabajamos con parametros forward (ecuación 3.31) podemos obtener:

$$R_{p+1}^{-1}(k) = \begin{bmatrix} 0 & 0 \\ 0 & R_p^{-1}(k - 1) \end{bmatrix} + \begin{bmatrix} I \\ -R_p^{-1}(k - 1)r_p^f \end{bmatrix} \quad (3.33)$$

$$\begin{bmatrix} r_o(k)^f - r_p^{fT}(k)R_p^{-1}(k - 1)r_p^f(k) \end{bmatrix}^{-1} \begin{bmatrix} I & -r_p^{fT}(k)R_p^{-1}(k - 1) \end{bmatrix}$$

Y si sustituimos la solución óptima para la predicción forward en el caso prewindowed ($R_p^f(k) = R_p(k - 1)$) ².

$$A_p(k) = -R_p^{-1}(k - 1)r_p^f(k)$$

¹Ver apendice A.

²Ver apendice B.



$$R_{p-1}(k) = \begin{bmatrix} 0 & 0 \\ 0 & R_p^{-1}(k-1) \end{bmatrix} - \begin{bmatrix} I \\ A_p(k) \end{bmatrix} \quad (3.34)$$

$$\left[r_o(k)^f - r_p^{fT}(k)A_p(k) \right]^{-1} \begin{bmatrix} I & A_p^T(k) \end{bmatrix}$$

Al término $r_o(k)^f - r_p^{fT}(k)A_p(k)$ en la última ecuación se le conoce como energía residual y se representa como:

$$\alpha_p^f(k) = r_o(k)^f - r_p^{fT}(k)A_p(k) \quad (3.35)$$

Sustituyendo las ecuaciones en 3.35, 3.34 y 3.28 en la ecuación 3.32 tenemos:

$$K_{p-1}(k) = \begin{bmatrix} 0 \\ K_p(k-1) \end{bmatrix} - \begin{bmatrix} I \\ A_p(k) \end{bmatrix} \left[\alpha_p(k)^{-f} \right] \left[\varepsilon_p^f[k] \right] \quad (3.36)$$

De igual forma si utilizamos los parámetros backward, en la ecuación 3.30 podemos obtener:

$$K_{p-1}(k) = \begin{bmatrix} K_p(k) \\ 0 \end{bmatrix} - \begin{bmatrix} B_p(k) \\ I \end{bmatrix} \left[\alpha_p(k)^{-b} \right] \left[\varepsilon_p^b[k] \right] \quad (3.37)$$

Podemos escribir la ecuación 3.37 como:

$$K_{p-1}(k) = \begin{bmatrix} M_p(k) \\ \mu(k) \end{bmatrix} \quad (3.38)$$

Donde:

$$\mu(k) = \alpha_p(k)^{-b} \varepsilon_p^b(k)$$

Y resolviendo para $K_p(k)$, las ecuaciones 3.36, 3.37 y 3.38 :

$$K_p(k) = M_p(k) - B_p(k)\mu(k) \quad (3.39)$$

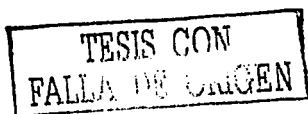
Al igual que para los parámetros de predicción forward $A_p(k)$ ¹ podemos escribir los parámetros de predicción backward $B_p(k)$ de manera recursiva:

$$B_p(k) = B_p(k-1) - K_p(k) \varepsilon_p^{bT}(k) \quad (3.40)$$

Sustituyendo la ecuación 3.40 en la ecuación 3.39:

$$K_p(k) = \frac{M_p(k) - B_p(k-1)\mu(k)}{1 - \varepsilon_p^{bT}(k)\mu(k)} \quad (3.41)$$

¹Ver apéndice B.



Donde $e_p^b(k)$ es el error de predicción backward a priori y se define como:

$$e_p^b(k) = u(k-p) - B_p(k-1)U_p(k) \quad (3.42)$$

Para que la ecuación 3.36 sea recursiva y completar la deducción del algoritmo necesitamos evaluar $\alpha_p^f(k)$.

RECURSIVIDAD PARA $\alpha_p^f(k)$.

De la definición:

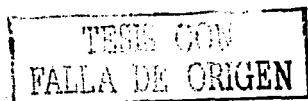
$$\alpha_p^f(k) = r_n(k)^f + r_p^{fT}(k)A_p(k)$$

Utilizando las definiciones B.16, B.17 y B.22 ¹, se puede demostrar que:

$$\alpha_p^f(k) = \lambda \alpha_p^f(k-1) + \varepsilon_p^f(k) e_p^{fT}(k) \quad (3.43)$$

El algoritmo "Fast Kalman" se muestra en el cuadro 3.3.

¹Ver apéndice B



| ALGORITMO FAST KALMAN | |
|--|------|
| Condiciones Iniciales | |
| $\alpha_p^f[k-1] = \delta, \quad \delta > 0$ $A_p[k-1] = B_p[k-1] = W_p[k-1] = K_p[k-1] = U_p[k-1] = 0$ | |
| Actualización de la ganancia de Kalman | |
| $e_p^f(k) = u(k) - A_p^T(k-1)U_p(k-1)$ | B.21 |
| $A_p(k) = A_p(k-1) - K_p(k-1)e_p^{fT}(k)$ | B.24 |
| $\varepsilon_p^f(k) = u(k) + A_p^T(k)U_p(k-1)$ | B.11 |
| $\alpha_p^f(k) = \lambda \alpha_p^f(k-1) + \varepsilon_p^f(k)e_p^{fT}(k)$ | 3.43 |
| $K_{p-1}(k) = \begin{bmatrix} 0 \\ K_p(k-1) \end{bmatrix} - \begin{bmatrix} I \\ A_p(k) \end{bmatrix} \left[\alpha_p(k)^{-1} \right] \left[\varepsilon_p^f(k) \right]$ | 3.36 |
| $e_p^b(k) = u(k-p) + B_p(k-1)U_p(k)$ | 3.42 |
| $K_p(k) = \frac{M_p(k) - B_p(k-1)U_p(k)}{1 - e_p^{fT}(k)U_p(k)}$ | 3.41 |
| $B_p(k) = B_p(k-1) - K_p(k)e_p^{bT}$ | 3.40 |
| Actualización para el Filtrado | |
| $e_p(k) = d(k) + W_p^T(k-1)U_p(k)$ | 3.23 |
| $W_p(k) = W_p(k-1) - K_p(k)e_p^T(k)$ | 3.26 |

Cuadro 3.3: Algoritmo FRLS "Fast Kalman".

3.4. Evaluación de algoritmos

Los resultados presentados en esta sección fueron evaluados bajo el esquema de un igualador lineal como el que se muestra en la figura 3.4.

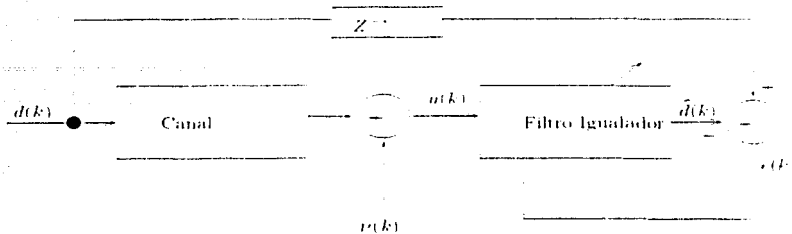


Figura 3.4: Esquema de igualación de canal.

Las señales utilizadas son generadas con la función `randn` de Matlab y luego cuantizadas. Los resultados se presentan en forma gráfica mediante la respuesta en frecuencia del canal y de su respectivo filtro igualador, así también se presenta una gráfica del error de filtrado contra el número de iteraciones para evaluar la velocidad de convergencia de cada algoritmo así como su desempeño con respecto al error al que convergen. Cabe destacar que para poder observar de manera más fácil la convergencia de los algoritmos se presentan las gráficas del error correspondientes a un promedio de 100 realizaciones para cada algoritmo.

Se presentarán los resultados de tres algoritmos de filtrado. Sólo se presenta una gráfica de respuesta en frecuencia puesto que los tres algoritmos presentan las mismas características frecuenciales. Sin embargo se presentan las gráficas correspondientes al error en donde se puede apreciar la velocidad de convergencia así como el error al que converge cada algoritmo.

Con respecto a las características del canal, este tiene una función de transferencia que corresponde a un canal de fase mínima:

$$H_c(z) = 1 - 1.1314z^{-1} + 0.6401z^{-2}$$

En la figura 3.5, es posible observar la respuesta en frecuencia de filtro LMS y la respuesta en frecuencia del canal, podemos apreciar que la respuesta del filtro corresponde precisamente a la inversa de la respuesta del filtro y por tanto compensa las distorsiones introducidas por el canal.

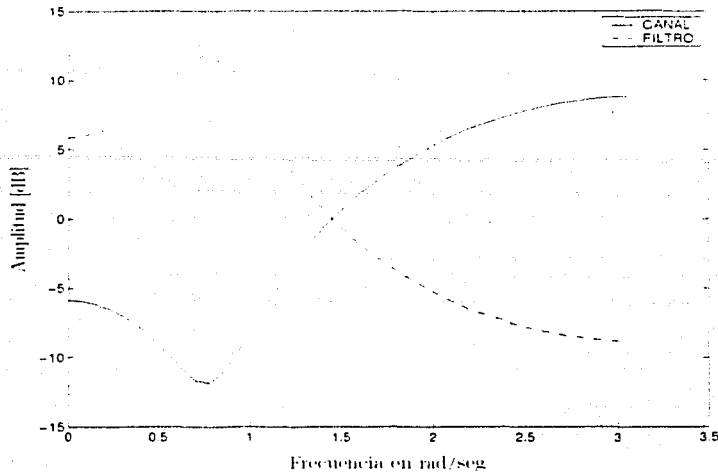


Figura 3.5: Respuesta es frecuencia del algoritmo LMS.

Con respecto al comportamiento del error de filtrado, en la figura 3.6 podemos apreciar el comportamiento del error para el algoritmo LMS. En este caso el error al que converge el algoritmo es menor que para los algoritmos RLS (figuras 3.7 y 3.8), sin embargo el tiempo requerido para converger a tal valor es mucho mayor que para los algoritmos RLS. En el caso de los algoritmos RLS y FRLS el comportamiento de ambos es idéntico pues la única diferencia es el número de iteraciones aritméticas que requieren para calcular los parámetros, sin embargo tanto el algoritmo RLS como el FRLS convergen a un valor mayor del error de filtrado comparado con el error al que converge el LMS.

TESIS CON
FALLA DE ORIGEN

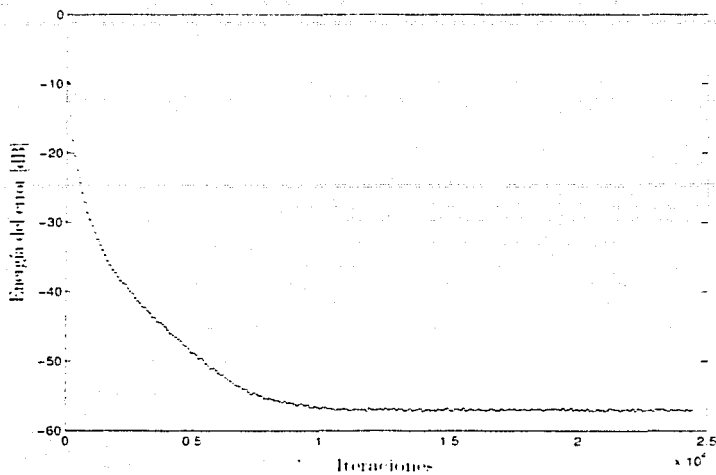


Figura 3.6: Error en el algoritmo de filtrado LMS.

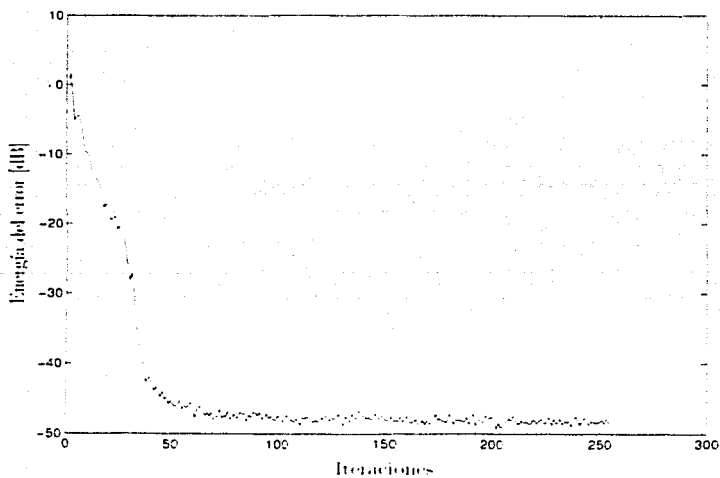


Figura 3.7: Error en el algoritmo de filtrado RLS.

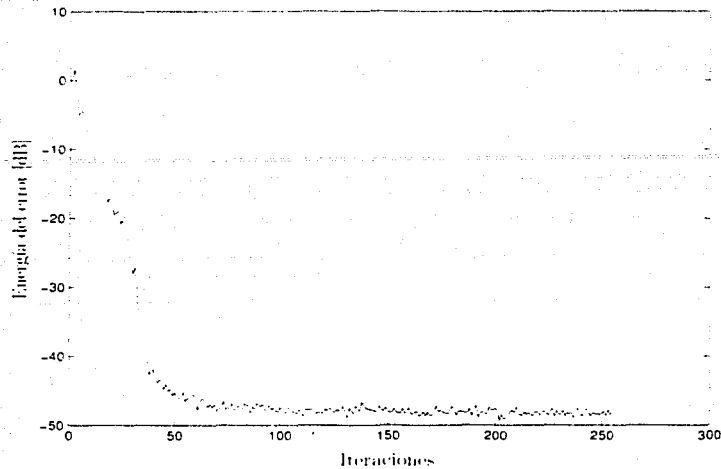


Figura 3.8: Error en el algoritmo de filtrado rápido RLS. Fast Kalman

3.5. Conclusiones

A lo largo de este capítulo hemos estudiado básicamente dos familias de algoritmos, los basados en el error cuadrático promedio y los basados en los mínimos cuadrados. En cuanto a los primeros, podemos decir que son muy fáciles de implementar con el inconveniente de que las características estadísticas de los procesos de entrada se ven restringidas, es decir estos algoritmos se reducen a trabajar en ambientes estacionarios, sin embargo sus características de convergencia pueden ser mejores que las de los algoritmos LS si se utilizan parámetros de paso suficientemente pequeños aunque también presentan el inconveniente del tiempo de convergencia el cual es mucho mayor que en los algoritmos LS. Por tanto, el empleo de uno u otro algoritmo dependerá de la aplicación deseada y de las características de las señales utilizadas.

Hemos comprendido la importancia de trabajar con matrices de correlación bien condicionadas, lo cual nos permitirá asegurar la estabilidad de los algoritmos, por otra parte; es necesario asegurar, sea cual sea el algoritmo utilizado, que este algoritmo converja más rápido que las variaciones que se presentan en el medio, por esta razón se han diseñado algoritmos rápidos como el presentado en la parte final de este capítulo aunque este tipo de algoritmos presentan problemas de divergencia debido a errores de redondeo los cuales se analizarán en el siguiente capítulo.

TESIS CON
FALLA DE ORIGEN

Hasta este momento hemos presentado diferentes algoritmos de filtrado y los resultados demuestran la efectividad de tales, sin embargo en situaciones reales estos algoritmos se deben implementar en procesadores que disponen de una precisión finita para almacenar los datos y resultados de operaciones aritméticas, tal limitación ocasiona errores de cuantización (truncamiento y/o redondeo), que muchas veces hacen diverger al algoritmo. Además, no solo la representación en precisión finita ocasiona que los errores de redondeo crezcan, también la utilización de factores de olvido $\lambda \ll 1$ ocasionan divergencia en los algoritmos, por estas razones en el siguiente capítulo se presenta un estudio de los errores de numéricos.

TESIS CON
FALLA DE ORIGEN

Capítulo 4

ANÁLISIS DE ROBUSTEZ NUMÉRICA EN ALGORITMOS DE FILTRADO ADAPTABLE

4.1. Introducción

La ventaja de los filtros digitales es que estos pueden ser implementados en dispositivos de procesamiento de señales digitales o DSP's, los cuales permiten realizar un gran número de operaciones en muy poco tiempo y por esta razón podemos utilizarlos en operaciones en tiempo real como telefonía o redes de datos.

Sin embargo, la representación de las cantidades utilizadas en dichos microprocesadores, tales como los datos de entrada, coeficientes del filtro y resultados de operaciones aritméticas, esta limitada por un tamaño de palabra finito lo cual ocasiona que los resultados obtenidos difieran de los datos teóricos. Estos errores, definidos como la diferencia entre la cantidad calculada y la cantidad teórica pueden ocasionar que los algoritmos no lleguen a su solución óptima y en muchos casos presenten divergencia.

Por estas razones, después de haber estudiado los algoritmos de filtrado adaptable bajo una precisión infinita de cálculo en el capítulo anterior, ahora nos proponemos desarrollar un análisis de los errores numéricos, truncatura y/o redondeo, debido a una longitud de palabra finita.

En este capítulo estudiaremos la propagación de errores en los algoritmos RLS y FRLS puesto que se ha demostrado que es precisamente en estas implementaciones donde se presentan un mayor número de problemas relacionados con los errores numéricos. Así mismo, en los análisis realizados se considera que no existen errores de cuantización de la señal de entrada.

En la última parte de este capítulo se estudiarán dos algoritmos rápidos estabilizados robustos con respecto a errores de redondeo.

4.2. Errores Numéricos

4.2.1. Tipos de errores

Es de gran importancia que en aplicaciones numéricas se analicen las fuentes y las magnitudes de los diferentes tipos de errores, y a su vez emplear algoritmos que minimicen el error. La figura 4.1 muestra la propagación del error en un algoritmo, donde el error de salida es el resultado del error de entrada y del error inherente al algoritmo.

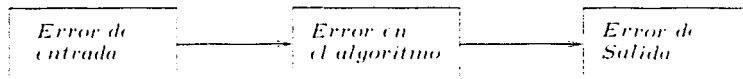


Figura 4.1: Errores en los algoritmos

Los errores de entrada surgen por lo general de la imperfección en la medición del fenómeno, por diferentes tipos de aproximaciones y por errores aleatorios en un sentido estadístico, así mismo por errores de cuantización de las señales.

Los errores en los algoritmos son básicamente de dos tipos: de truncamiento y de redondeo. Como objetivo, es necesario encontrar algoritmos que minimicen el crecimiento del error cuando aumenta el número de cálculos.

Para el cálculo de errores es necesario distinguir entre el error actual y el relativo:

$$\text{Error actual} = \text{Salida real} - \text{Salida computacional}$$

$$\text{Error relativo} = \frac{\text{Salida real} - \text{Salida computacional}}{\text{Salida real}}$$

4.2.2. Errores por redondeo

El error por redondeo es un problema que surge de aproximar cantidades debido a que para representarlas, un número finito de decimales es usado en vez de un número infinito.

Dígitos significativos: Son aquellos que contienen la información real de la magnitud del número. Para ser significativo, el último dígito deberá ser igual a la mitad de la unidad en el último lugar decimal.

Es importante distinguir entre los términos *aritmética de punto fijo*, en el cual se maneja un número fijo de lugares decimales; y *aritmética de punto flotante*, el cual se emplea en trabajo científico y en diversos lenguajes de programación. Los números en aritmética de punto flotante son expresados de la forma $a \cdot 10^b$. Si a se encuentra entre $0.1 < a \leq 1$, entonces se dice que el número está normalizado. [17]

Una expresión matemática que proporciona una mejor aproximación del error por redondeo es la conocida como *intervalo aritmético*. Para esta aproximación el número redondeado se reemplazará por los dos números que representan los posibles valores máximo y mínimo. Los dos cálculos son desarrollados, de modo que el valor verdadero deberá estar dentro del intervalo. Si nombramos al valor real como x_i y al valor aproximado (redondeado) como x_i^* ; entonces el error actual E_i y el error relativo E_i^R son respectivamente:

$$E_i = x_i - x_i^* \quad (4.1)$$

$$E_i^R = \frac{x_i - x_i^*}{x_i} = \frac{E_i}{x_i} \quad (4.2)$$

4.2.3. Errores en operaciones aritméticas

Si consideramos el proceso de adición en donde es necesario encontrar $\sum_{i=1}^n x_i$ y dado que solo conocemos los números redondeados, calculamos $\sum_{i=1}^n x_i^*$. De la fórmula 4.1 podemos suponer un error por redondeo dado por, [17]:

$$\text{Error actual por redondeo} = \sum_{i=1}^n E_i$$

En el análisis del error por redondeo es necesario considerar los valores absolutos. Si suponemos que $E_i \leq E$ para $i = 1, 2, \dots, n$. Entonces el máximo error por redondeo puede ser obtenido mediante:

$$\left| \sum_{i=1}^n E_i \right| \leq \sum_{i=1}^n |E_i| \leq nE \quad (4.3)$$

En la expresión anterior la respuesta es la que se produciría empleando el máximo (o mínimo) valor de cada error. Esta fórmula también se aplica a sustracciones, ya

que puede ser manejada como la suma de las cantidades negativas correspondientes.

Para el caso de la multiplicación es un poco más complejo, en este caso el uso del error relativo 4.2 es más útil. Si consideramos el error relativo del producto de dos números x_1 y x_2 como $E_{x_1 x_2}^R$, entonces tenemos:[17]:

$$\begin{aligned} E_{x_1 x_2}^R &= \frac{x_1 x_2 - x_1^* x_2^*}{x_1 x_2} \\ &= \frac{(x_1^* - E_1)(x_2^* - E_2) - x_1^* x_2^*}{x_1 x_2} \\ &= \frac{x_1^* E_2 + x_2^* E_1 - E_1 E_2}{x_1 x_2} \end{aligned}$$

Si suponemos que E_1 y E_2 son pequeños comparados con x_1 y x_2 , tenemos que $x_1^* \approx x_1$, $x_2^* \approx x_2$ y $\frac{E_1 E_2}{x_1 x_2} \approx 0$, por lo tanto:

$$E_{x_1 x_2}^R \approx \frac{E_1}{x_2} + \frac{E_2}{x_1} = E_1^R + E_2^R \quad (4.4)$$

De forma similar, para la división tenemos:

$$E_{x_1 \div x_2}^R \approx E_1^R - E_2^R \quad (4.5)$$

4.2.4. Propagación del Error

La resolución de un problema matemático por métodos numéricos da como solución una aproximación numérica del resultado exacto. Es importante, por tanto, tratar de conocer el efecto que tiene sobre el resultado final cada una de las operaciones realizadas con números aproximados.

El efecto de los errores en el resultado final de operaciones básicas aplicadas a dos números x_1 y x_2 se obtuvo en las ecuaciones 4.3, 4.4 y 4.5.

Cuando el problema consiste en calcular el resultado $y = f(x)$ tenemos la siguiente fórmula aproximada de propagación del error:[6]:

$$E(y) = |f'(x)|E(x)$$

En el caso más general, en que una función depende de más de una variable $y = f(x_1, x_2, \dots, x_n)$ la fórmula aproximada de propagación del error es:[6]:



$$E(y) = \sum_{i=1}^n \left(\frac{\partial}{\partial x_i} f(x_1, x_2, \dots, x_n) \right) E(x_i)$$

4.3. Análisis general de la propagación de errores en algoritmos recursivos

Los algoritmos adaptables son recursivos por naturaleza y podemos definir el estado actual del algoritmo, $\Theta(k)$, en función de un estado anterior y de las secuencias de entrada $U_p(k)$ y $d(k)$. Esto es:

$$\Theta(k) = f(\Theta(k-1), d(k), U_p(k)) \quad (4.6)$$

Cualquier implementación numérica del algoritmo generará imprecisiones debido a errores de redondeo y de representación en el microprocesador, por lo tanto el algoritmo correrá con cantidades perturbadas, $\hat{\Theta}(k)$, y puede describirse como:

$$\hat{\Theta}(k) = f(\hat{\Theta}(k-1), d(k), U_p(k)) + V(k) \quad (4.7)$$

Donde $V(k)$ representa el ruido de redondeo.¹ Podemos entonces expresar la magnitud del error como $\Delta\Theta(k) = \hat{\Theta}(k) - \Theta(k)$, la cual se puede expresar como:[15]:

$$\Delta\Theta(k) = \Delta\Theta(k-1)F(k) + V(k) \quad (4.8)$$

Donde $F(k)$ es función de $\Theta(k-1)$, $d(k)$ y $U_p(k)$ y depende de la implementación que se utilice.[15].

La ecuación 4.8 nos permite realizar el estudio de los errores de redondeo a través de los siguientes pasos:

1. **Generación del error:** Describir a $V(k)$ tal como es generado al instante de recursión k .
2. **Propagación del error:** Cómo los errores presentados al instante k_0 se propagan asumiendo que en las subsecuentes recursiones no se generan nuevos errores.
3. **Acumulación de errores:** Equivale a sumar los pasos 2 y 3.

El paso 1 depende en gran manera de la implementación utilizada ya sea punto flotante, punto fijo, normalización, formas de raíz cuadrada, etc., por esta razón no

¹ Llamamos ruido de redondeo a los errores generados al instante k debido a errores de redondeo.

estudiaremos este primer paso que ha sido estudiado para diferentes implementaciones en [8], [32], [2], [1].

El comportamiento de la ecuación 4.8 nos define la estabilidad del algoritmo con respecto a los errores de redondeo. Por tanto si ésta es inestable, es decir un error generado a un instante $k - 1$ crece de manera infinita al propagarse al instante k , entonces tal implementación es inestable y no se podrá utilizar.

En particular, si el primer término de la suma en la ecuación 4.8 converge entonces la convergencia de tal implementación dependerá de que los errores generados al instante k no crezcan de manera indefinida lo cual se puede lograr utilizando una longitud de palabra adecuada.

El paso 2 del método de análisis se reduce al estudio de estabilidad de la ecuación:

$$\Delta\Theta(k) = \Delta\Theta(k-1)F(k) \quad (4.9)$$

La cual puede escribirse de manera general como,[15]:

$$\Delta\Theta(k) = \Delta\Theta(k_0)\Psi(k_0, k) \quad (4.10)$$

donde:

$$\Psi(k_0, k) = F(k_0+1)F(k_0+2)\dots F(k) \quad (4.11)$$

La ecuación 4.10 nos muestra como se propaga un error generado al instante $k_0 < k$ hasta el instante k . Podemos observar como tal propagación del error depende básicamente de la matriz $\Psi(k_0, k)$ conocida como matriz de transición, la cual juega un rol muy importante en el análisis del error.

La matriz de transición, ecuación 4.11, es la que nos define la propagación de los errores desde el instante en que ocurren hasta el instante de análisis. Por esta razón, es importante un estudio detallado de su comportamiento cuando se pretende analizar el comportamiento numérico de alguna implementación en particular. En la siguiente sección veremos como se define esta matriz así como su comportamiento para los algoritmos RLS clásicos.



4.4. Análisis de los errores de redondeo en los algoritmos RLS

Aplicaciones de los filtros RLS han demostrado que estos presentan problemas de divergencia [32], la cual principalmente se presenta por la pérdida de simetría de la inversa de la matriz de covarianza. A continuación se presentará un análisis de errores de redondeo para dos implementaciones de filtros RLS.

Dada la solución para el algoritmo RLS convencional obtenida en el capítulo 3:

$$W_p(k) = W_p(k-1) - R_p^{-1}(k)U_p^T(k)e_p^T(k) \quad (4.12)$$

$$R_p(k) = \lambda R_p(k-1) + U_p(k)U_p^T(k) \quad (4.13)$$

donde:

$$e_p(k) = d(k) - W_p^T(k)U_p(k)$$

El problema en la ecuación 4.12 consiste en evaluar $K_p(k) = -R_p^{-1}(k)U_p^T(k)$, la cual para el algoritmo RLS a priori se puede escribir como [20], [32]:

$$K_p(k) = \frac{R_p^{-1}(k-1)U_p^T(k)}{\lambda + U_p^T(k)R_p^{-1}(k-1)U_p(k)} \quad (4.14)$$

Y de acuerdo al lema de inversión matricial¹:

$$R_p^{-1}(k) = \frac{1}{\lambda} [R_p^{-1}(k-1) - \frac{R_p^{-1}(k-1)U_p^T(k)U_p^T(k)R_p^{-1}(k-1)}{\lambda + U_p^T(k)R_p^{-1}(k-1)U_p(k)}] \quad (4.15)$$

Para la solución del problema definimos las siguientes variables:

$$o(k) = U_p^T(k)R_p^{-1}(k-1) \quad (4.16)$$

$$\rho(k) = \lambda + U_p^T(k)R_p^{-1}(k-1)U_p(k) \quad (4.17)$$

$$\overline{R}_p^{-1}(k-1) = \frac{R_p^{-1}(k-1)U_p^T(k)U_p^T(k)R_p^{-1}(k-1)}{\lambda + U_p^T(k)R_p^{-1}(k-1)U_p(k)} \quad (4.18)$$

Dos implementaciones de este algoritmo conocido como algoritmo convencional RLS a menudo utilizado se muestra en la tabla 4.1.

En el algoritmo CLS1 claramente se explota la simetría de la matriz $R_p^{-1}(k-1)$ para calcular $K_p(k)$. La única razón de hacerlo es una ligera reducción en la complejidad de cálculo. Además, la operación para calcular $\overline{R}_p^{-1}(k-1)$ es "aparentemente"

¹Ver apéndice A.

| ALGORITMO RLS [32] | | |
|----------------------------|---|---|
| Variable | CLS1 | CLS2 |
| $\phi(k)$ | $U_p^T(k)R_p^{-1}(k-1)$ | $U_p^T(k)R_p^{-1}(k-1)$ |
| $\rho(k)$ | $\lambda - \phi(k)U_p(k)$ | $\lambda - \phi(k)U_p(k)$ |
| $K_p(k)$ | $\frac{\phi^T(k)}{\rho(k)}$ | $\frac{R_p^{-1}(k-1)U_p(k)}{\rho(k)}$ |
| $\overline{R}_p^{-1}(k-1)$ | $K_p(k)\phi(k)$ | $K_p(k)\phi(k)$ |
| $R_p^{-1}(k)$ | $\frac{1}{\lambda}[R_p^{-1}(k-1) - \overline{R}_p^{-1}(k-1)]$ | $\frac{1}{\lambda}[R_p^{-1}(k-1) - \overline{R}_p^{-1}(k-1)]$ |

Cuadro 4.1: Algoritmos RLS

simétrica dejando que $R_p^{-1}(k-1)$ permanezca simétrica. Sin embargo, el análisis teórico del error demuestra que precisamente ocurre lo contrario, presentándose así problemas de divergencia.

Existe otra implementación para este algoritmo la cual no considera simetría de $R_p^{-1}(k-1)$ y por tanto el algoritmo no depende de la simetría, el único cambio con respecto al algoritmo anterior es la forma de calcular la ganancia de Kalman como se ilustra en el cuadro 4.1.

4.4.1. Análisis teórico del error

PROPAGACIÓN DE UN ERROR SIMPLE EN RECURSIONES SUBSECUENTES. [32].

Consideraremos la propagación de un solo error al instante $k-1$ con subsecuentes recursiones.

Teorema:

Definimos las cantidades erróneas al instante $k-1$ denotadas por “ $\tilde{\cdot}$ ” como:

$$\tilde{R}_p^{-1}(k-1) = R_p^{-1}(k-1) - \delta R_p^{-1}(k-1) \quad (4.19)$$

$$\tilde{W}_p(k-1) = W_p(k-1) - \delta W_p(k-1) \quad (4.20)$$

$$\tilde{\rho}(k-1) = \rho(k-1) - \delta \rho(k-1) \quad (4.21)$$

TESIS CON
FALLA DE ORIGEN

Para observar la manera en que intervienen estas cantidades erróneas calculamos $\delta K_p(k)$ y su influencia en $\delta R_p^{-1}(k)$. Considerando la ecuación 4.21 podemos escribir:

$$\tilde{\rho}(k) = \lambda - U_p^T(k)R_p^{-1}(k-1)U_p(k) - \delta[\lambda - U_p^T(k)R_p^{-1}(k-1)U_p(k)]$$

Puesto que λ es una constante no existe un error en esta si suponemos que la aritmética utilizada puede representar su valor sin error y por lo tanto no existe una propagación del error sobre λ , entonces el error $\delta\rho(k)$ se reduce a:

$$\delta\rho(k) = U_p^T(k)\delta R_p^{-1}(k-1)U_p(k)$$

Y podemos escribir a $\tilde{\rho}^{-1}(k)$ como:

$$\tilde{\rho}^{-1}(k) = \frac{1}{\lambda - U_p^T(k)R_p^{-1}(k-1)U_p(k) - U_p^T(k)\delta R_p^{-1}(k-1)U_p(k)} \quad (4.22)$$

y el error en $\tilde{\rho}^{-1}(k)$ está dado por:

$$\begin{aligned} \delta\rho^{-1}(k) &= \tilde{\rho}^{-1}(k) - \rho^{-1}(k) = \\ &= \frac{1}{\lambda - U_p^T(k)R_p^{-1}(k-1)U_p(k) - U_p^T(k)\delta R_p^{-1}(k-1)U_p(k)} - \\ &= \frac{1}{\lambda - U_p^T(k)R_p^{-1}(k-1)U_p(k)} \end{aligned} \quad (4.23)$$

Finalmente podemos escribir:

$$\begin{aligned} \rho^{-1}(k) &= \tilde{\rho}^{-1}(k) + \delta\rho^{-1}(k) = \\ &= \frac{1}{\lambda - U_p^T(k)R_p^{-1}(k-1)U_p(k)} \left[1 - \frac{U_p^T(k)\delta R_p^{-1}(k-1)U_p(k)}{\lambda - U_p^T(k)R_p^{-1}(k-1)U_p(k)} \right] + O(\delta^2) \end{aligned} \quad (4.24)$$

Donde $O(\delta^2)$ indica el orden de la magnitud $\|\delta R_p^{-1}(k-1)\|^2$, generalmente se utiliza la norma de Frobenius¹, y $\tilde{c}(k)$:

$$\tilde{c}(k) = c(k) - \delta c(k)$$

donde:

$$\delta c(k) = U_p^T(k)\delta R_p^{-1}(k-1)$$

¹Ver apéndice E.

Por lo tanto $\widetilde{K}_p(k)$ está dada por:

$$\widetilde{K}_p(k) = \widetilde{c}(k)^T \rho^{-1}(k)$$

Entonces el error $\delta K_p(k)$ está dado por:

$$\delta K_p(k) = \frac{c^T(k)}{\delta \rho(k)} - \frac{\delta c^T(k)}{\rho(k)} - O(\delta^2) \quad (4.25)$$

El primer término en la ecuación 4.25 corresponde al error sobre $K_p(k)$ debido a $\rho(k)$ mientras que el segundo término corresponde al error debido a $R_p^{-1}(k-1)$. Sustituyendo:

$$\delta K_p(k) = \frac{\delta R_p^{-T}(k-1)U_p(k)}{\rho(k)} - \frac{R_p^{-T}(k-1)U_p(k)U_p^T(k)\delta R_p^{-1}(k-1)U_p(k)}{\rho(k)} - O(\delta^2) \quad (4.26)$$

Y el error en la matriz de correlación esta dado por:

$$\delta R_p^{-1}(k) = \frac{1}{\lambda} [\delta R_p^{-1}(k-1) - K_p(k)U_p^T(k)\delta R_p^{-1}(k-1) - \delta K_p(k)U_p^T(k)R_p^{-1}(k-1)] \quad (4.27)$$

Los primeros dos términos en la ecuación 4.27 corresponden al error sobre $R_p^{-1}(k)$ debido a $R_p^{-1}(k-1)$ mientras que el tercer término corresponde al error debido a $K_p(k)$.

Y sustituyendo la ecuación 4.26 en 4.27:

$$\begin{aligned} \delta R_p^{-1}(k) = & \frac{1}{\lambda} [\delta R_p^{-1}(k-1) - K_p(k)U_p^T(k)\delta R_p^{-1}(k-1) - \\ & \left\{ \frac{\delta R_p^{-T}(k-1)U_p(k)}{\rho(k)} - \frac{R_p^{-T}(k-1)U_p(k)U_p^T(k)\delta R_p^{-1}(k-1)U_p(k)}{\rho(k)} \right\} \\ & [U_p^T(k)R_p^{-1}(k-1)]] - O(\delta^2) \end{aligned} \quad (4.28)$$

O bien:

$$\begin{aligned} \delta R_p^{-1}(k) = & \frac{1}{\lambda} [\delta R_p^{-1}(k-1) - K_p(k)U_p^T(k)\delta R_p^{-1}(k-1) - \\ & \delta R_p^{-T}(k-1)U_p(k)K_p^T(k) + K_p(k)U_p^T(k)\delta R_p^{-1}(k-1)U_p(k)K_p^T(k) + O(\delta^2)] \end{aligned} \quad (4.29)$$

Acomodando términos podemos escribir, [32]:

$$\delta R_p^{-1}(k) = \frac{1}{\lambda} (I - K_p(k)U_p^T(k))\delta R_p^{-1}(k-1)(I - K_p(k)U_p^T(k))^T -$$

$$\frac{1}{\lambda}(\delta R_p^{-1}(k-1) - \delta R_p^{-1}(k-1)U_p^T(k)K_p^T(k) + O(\delta^2)) \quad (4.30)$$

Haciendo un análisis similar podemos obtener el error correspondiente al instante k debido a errores al instante $k-1$ para los parámetros $W_p(k)$. Este error está dado por, [32]:

$$\begin{aligned} \delta W_p(k) = (I - K_p(k)U_p^T(k))[\delta W_p(k-1) + \frac{\delta R_p^{-1}(k-1)U_p^T(k)}{\lambda - U_p^T(k)R_p^{-1}(k-1)U_p^T(k)} \\ (dk) - U_p^T(k)W_p(k-1)] + O(\delta^2) \end{aligned} \quad (4.31)$$

En la ecuación 4.30 podemos apreciar que el error en $R_p^{-1}(k)$ depende de la simetría de la matriz de covarianza. Por lo tanto, si la matriz se conserva simétrica durante todo el proceso, el segundo término de la ecuación 4.30 desaparece y existe menos probabilidad de que los errores crezcan demasiado. Sin embargo, si la matriz de covarianza no se conserva simétrica el error puede crecer a cada instante y ocasionar la divergencia del algoritmo.

Un análisis similar para el algoritmo CLS2 nos lleva a los siguientes resultados para el error en $R_p^{-1}(k)$, simplemente considerando que para el cálculo de la ganancia $K_p(k)$ no se hace uso de $\phi^T(k)$ pues no se considera $R_p^{-1}(k-1)$ simétrica. Obtenemos por lo tanto, [32]:

$$\delta R_p^{-1}(k) = \frac{1}{\lambda}(I - K_p(k)U_p^T(k))\delta R_p^{-1}(k-1)(I - K_p(k)U_p^T(k))^T + O(\delta^2) \quad (4.32)$$

Mientras que para el error en los parámetros del filtro se obtiene la misma expresión que para el caso de la implementación CLS1, ecuación 4.31.

En la ecuación 4.32, a diferencia de la ecuación 4.30, se observa que la propagación del error no depende de la simetría de la matriz de correlación, por esta razón, la convergencia de los errores depende básicamente de la matriz $(I - K_p(k)U_p^T(k))$ conocida como matriz de transición.

ANÁLISIS DE LA MATRIZ DE TRANSICIÓN.

A continuación se presentará el análisis de la propagación de un error generado al instante k_0 donde $k_0 < k$. Para ello es necesario basarnos en las siguientes definiciones:

Definición 1: El vector de regresión $U_p(i)$ es persistentemente excitado sobre el intervalo de observación $k_0 \leq i \leq k$ cuando se usa un factor de olvido exponencial $\lambda \leq 1$, si la siguiente condición se cumple, [32]:

$$\alpha I \leq \sum_{i=k}^k X^T U_p(i) U_p^T(i) \leq \beta I, \quad \forall \alpha, \beta > 0 \quad (4.33)$$

La condición de excitación persistente se cumple cuando la señal recibida se compone de valores diferentes de cero durante el proceso de filtrado. Cuando esta condición no se cumple los algoritmos divergen debido a la generación de matrices de covarianza singulares. La influencia de la falta de excitación persistente se puede observar en [32], donde observamos que, incluso para una implementación robusta, la falta de excitación persistente ocasiona divergencia en el algoritmo.

Definición 2: Una matriz M es una contracción si $\|M\|_2$ es estrictamente menor a 1.

Definición 3: Para una terna de matrices cuadradas (M, N, E) de la misma dimensión, la matriz $M = N - E$ es una perturbación aguda de la matriz M cuando $\|N^{-1}\|_2 \|E\|_2 < 1$, lo cual asegura que la matriz E no modifica el rango de N . [30].

Basados en las definiciones anteriores podemos escribir el siguiente lema:

LEMA: Cuando el vector de regresión $U_p(k)$ es persistentemente excitado y $\tilde{R}_p^{-1}(k) = R_p^{-1}(k) - \delta R_p^{-1}(k)$ es una perturbación aguda de $R_p^{-1}(k)$, entonces $R_p^{-1}(k) \delta R_p^{-1}(k)$ es una contracción.

Si consideramos el efecto de un error inducido al instante de tiempo k_o en las cantidades $R_p^{-1}(k)$ y $W_p(k)$. Cuando $R_p^{-1}(k)$ permanece simétrica o en la implementación CLS2, podemos escribir la ecuación 4.32 como:

$$\delta R_p^{-1}(k) = \frac{1}{\lambda^{k-k_o}} c(k, k_o) \delta R_p^{-1}(k_o) c(k, k_o)^T + O(\delta^2) \quad (4.34)$$

Mientras que de acuerdo con la ecuación 4.31 el error sobre los parámetros está dado por:

$$\delta W_p(k) = c(k, k_o) \delta W_p(k_o) + \sum_{j=0}^{k-k_o-1} c(k, k_o + j) \mu_{k_o-1-j} + O(\delta^2) \quad (4.35)$$

donde:

$$\mu_i = \frac{\delta R_p^{-1}(i-1) U_p(i)}{\lambda + U_p^T(i) R_p^{-1}(i-1) U_p(i)} (d(i) - U_p^T(i) W_p(i-1))$$

$$c(k, k_o + j) = \prod_{i=k_o+j}^k (I - K_p(i) U_p^T(i))$$

Y se puede demostrar que $c(k, k_0)$, matriz de transición, se puede expresar como, [20]:

$$c(k, k_0) = \lambda^{k-k_0} R_p^{-1}(k) R_p(k_0)$$

Si suponemos que $d(i) = U_p^T(i) W_p(i - 1)$ es una variable aleatoria con media cero, entonces el segundo término en la ecuación 4.35 desaparece cuando consideramos solo la media de $\delta W_p(k)$. Bajo esta suposición, en adelante, analizamos solo la propagación de $E[\delta W_p(k)]$.

Los errores $\delta R_p^{-1}(k_0)$ y $E[\delta W_p(k)]$ son atenuados si la matriz de transición $c(k, k_0)$ es una contracción. Esto se puede verificar en el siguiente Lema.

LEMA: Cuando el vector de regresión $U_p(k)$ es persistentemente excitado sobre el intervalo (k_0, k) , la matriz de transición $c(k, k_0)$ es una contracción.

Prueba: La matriz $R_p(k)$ definida como:

$$R_p(k) = \sum_{i=1}^k \lambda^{k-i} Y_p(i) Y_p^T(i)$$

Dado que $k_0 < k$, podemos escribir:

$$\begin{aligned} R_p(k) &= \sum_{i=1}^{k_0} \lambda^{k-i} Y_p(i) Y_p^T(i) + \sum_{i=k_0}^k \lambda^{k-i} Y_p(i) Y_p^T(i) \\ &= \lambda^{k-k_0} \sum_{i=1}^{k_0} \lambda^{k_0-i} Y_p(i) Y_p^T(i) + \sum_{i=k_0}^k \lambda^{k-i} Y_p(i) Y_p^T(i) \end{aligned}$$

Por lo tanto:

$$R_p(k) = \lambda^{k-k_0} R_p(k_0) + \sum_{i=k_0}^k \lambda^{k-i} Y_p(i) Y_p^T(i)$$

Cuando la ecuación 4.33 se cumple, la matriz:

$$R_p(k) - \lambda^{k-k_0} R_p(k_0)$$

se define positiva¹ y entonces:

$$\lambda^{k-k_0} R_p(k_0) < R_p(k)$$

¹Ver apéndice E.

En consecuencia:

$$\| \lambda^{k-k_0} R_p^{-1}(k) R_p(k) \|_2 < 1$$

Con lo que se completa la prueba. Por lo tanto podemos decir que si la matriz de covarianza se conserva simétrica o se utiliza la implementación CLS2 y además se cumple la condición de excitación persistente, entonces esta es una implementación robusta con respecto a errores numéricos.

ERRORES QUE SE GENERAN EN UNA RECURSIÓN

El siguiente teorema especifica los límites para el error en las cantidades actualizadas en cada recursión tanto para la implementación CLS1 como para la CLS2.

Si $\Delta_{R_p^{-1}}$ y Δ_{W_p} denotan las normas de los errores absolutos al calcular R_p^{-1} y W_p , respectivamente, entonces las siguientes expresiones nos definen los límites para tales errores en cada recursión.[32]:

$$\Delta_{R_p^{-1}} \leq \epsilon_1 \| R_p^{-1}(k) \|$$

$$\Delta_{W_p} \leq \epsilon_2 (\| H_p(k) \| + \| K_p(k) \| \| y(k) \|)$$

Donde ϵ_1 y ϵ_2 son constantes que dependen de la precisión utilizada.

En general estos errores son limitados de la misma manera para las implementaciones del tipo Cholesky o raíz cuadrada de la covarianza. En ambos casos proporcionales a $\| R_p(k) \|$. [32].

ACUMULACIÓN E INTERACCIÓN DE LOS ERRORES DE REDONDEO

El error total se denota por el prefijo δ_{tot} en lugar de δ para el caso de la propagación de una recursión a otra.

Suposición: El error total en ciertos instantes de recursión k es la suma de la propagación del error previo más los errores locales en el tiempo de la recursión.

Teorema: Cuando el vector de regresión $U_p(k)$ es persistentemente excitado los errores sobre $\tilde{R}_p^{-1}(k)$ y $E[\tilde{W}_p(k)]$ son limitados de la misma forma tanto para la implementación CLS1 y CLS2.

Prueba: Relacionando la ecuación 4.33 con las fronteras dadas en la subsección 4.4.1. de acuerdo a la suposición 1 se obtiene el límite superior para el error total.[32]:

$$\| \delta_{tot} R_p^{-1}(k) \| \leq \frac{1}{\lambda^{k-i_m}} \| c(k, k_m) \|^2 \| \delta_{tot} R_p^{-1}(k_m) \| + \overline{\Delta}_{R_p^*}^m \quad (4.36)$$

En esta ecuación k_m es el instante de tiempo mas cercano, menor que k , para el cual $c(k, k_m)$ es una contracción y $\overline{\Delta}_{R_p^*}^m$ es dada por:

$$\overline{\Delta}_{R_p^*}^m = \sum_{i=k_m}^k \epsilon_1 \| R_p^{-1}(i) \|^2$$

Esta última expresión representa la acumulación de los errores locales hechos en cada recursión en el cálculo de $R_p^{-1}(i)$ en el intervalo (k_m, k) .

Cuando $R_p^{-1}(i)$ permanece limitada, asegurado por la condición de excitación persistente, $\overline{\Delta}_{R_p^*}^m$ también permanece limitado.

De manera similar, la media del error total en los parámetros estimados.[32]:

$$\| E(\delta_{tot} W_p(k)) \| \leq \| c(k, k_m) \| \| E(\delta_{tot} W_p(k_m)) \| + \overline{\Delta}_{W_p}^m \quad (4.37)$$

Las ecuaciones 4.36 y 4.37 representan los modelos de propagación del error total en la implementación CLS2 cuando la señal de entrada es limitada y persistentemente excitada.[32]. Podemos apreciar que ambos modelos dependen de la matriz de transición, por tanto es importante garantizar que ésta sea una contracción durante el proceso, lo cual se puede asegurar con la condición de excitación persistente.

4.5. Generalidades de los algoritmos rápidos.

Todos los algoritmos rápidos actualizan la ganancia de Kalman de manera eficiente. Las versiones más rápidas lo hacen con $7p$ operaciones por iteración y utilizan la ganancia de Kalman dual $C_p(k)$.[21].

Dado que para obtener los algoritmos podemos utilizar tanto errores de predicción a priori como errores de predicción a posteriori, existen dos familias de algoritmos: los algoritmos a priori y los algoritmos a posteriori.

El algoritmo rápido de Kalman dado en el capítulo 3 es un algoritmo del tipo a priori y en consecuencia utiliza la ganancia de Kalman directa $K_p(k)$ y errores a priori para la obtención de los parámetros del filtro. Este algoritmo presenta buenas propiedades de estabilidad como se mostrará en el siguiente capítulo, sin embargo; se ha demostrado que los algoritmos más rápidos son los que utilizan la ganancia de

Kalman dual $C_p(k)$, tales algoritmos se conocen como algoritmos **a posteriori** pues utilizan errores a posteriori para el cálculo de los parámetros del filtro.

Las propiedades de estabilidad de los algoritmos **a posteriori** no son tan buenas como en el caso del filtro de Kalman, por tal razón es necesario obtener algoritmos estabilizados de tales versiones.

En la sección 4.6.3 se presentarán dos versiones estabilizadas de algoritmos correspondientes a los algoritmos rápidos del apéndice D. En particular, la versión de Slock introduce la variable de verosimilitud $\gamma_p(k)$ para relacionar de manera escalar los errores de predicción a priori y a posteriori y reducir de esta manera el tiempo de cálculo. Además, la actualización de $\gamma_p(k)$ se realiza a través de su inversa mediante la expresión:

$$\gamma_p^{-1}(k) = \gamma_{p-1}^{-1}(k) - c_p^{bT}(k)\delta(k)$$

Donde $\delta(k)$ es el último elemento de la ganancia de Kalman dual al orden $p-1$ y:

$$\gamma_{p-1}^{-1}(k) = \gamma_{p-1}^{-1}(k-1) - C_{p-1}^1(k)c_p^f(k)$$

El cálculo del error de predicción backward a priori $c_p^b(k)$ se realiza mediante la expresión:

$$c_p^b(k) = -\lambda\alpha_p^b(k-1)\delta(k)$$

Esta última expresión solo requiere de una multiplicación entre escalares y utilizando la variable $\gamma_p(k)$ podemos obtener $\varepsilon_p^b(k)$ mediante otra multiplicación entre escalares como:

$$\varepsilon_p^b(k) = c_p^b(k)\gamma_p(k)$$

El algoritmo rápido de **Slock** requiere $7p$ operaciones aritméticas por iteración para la actualización del filtro. La disminución de las operaciones se basa en el hecho de que se pueden calcular los errores de predicción a partir de multiplicaciones entre escalares mediante la introducción de la variable $\gamma_p(k)$ en lugar de obtener tales errores a partir de su definición, lo cual implica un producto escalar entre vectores.

En el caso de la versión **Cioffi**, a diferencia de la versión de **Slock** se actualiza directamente $\gamma_p(k)$ y no su inversa como:

$$\gamma_p(k) = [1 + \gamma_{p-1}(k)c_p^{bT}(k)\delta(k)]^{-1}\gamma_{p-1}(k)$$

Donde:

$$\gamma_{p-1}(k) = \lambda \gamma_p(k-1) \alpha_p^f(k-1) \alpha_p^f(k)$$

El cálculo de los errores y la actualización del filtro se realiza de la misma forma que en la versión **Stoick**.

En particular el algoritmo de **Stoick** actualiza de manera separada el último elemento de la ganancia de Kalman dual al orden $p-1$, $C_{p-1}^{\mu-1}(k)$, y actualiza directamente la inversa de la energía residual forward $\alpha_p^f(k)$, evitando así una división que podría introducir errores numéricos. Como se verá en el siguiente capítulo el algoritmo de **Stoick** presenta mejores propiedades numéricas que el algoritmo de **Cioffi** aun cuando se trabaja con la versión no estabilizada.

4.6. Errores de redondeo en algoritmos rápidos (FRLS)

Los algoritmos RLS a pesar de presentar propiedades bastante buenas con respecto a velocidad de convergencia, facilidad de implementación y no presentar limitaciones sobre el tipo de señales procesadas, presentan el inconveniente de que necesitan del orden de p^2 operaciones aritméticas por iteración para obtener los parámetros del filtro, donde p es el orden del filtro, por esta razón para aplicaciones en tiempo real se ve limitada la velocidad de transmisión debido a el tiempo que le lleva al algoritmo calcular los parámetros. Sin embargo, se han desarrollado algoritmos denominados como **rápidos**, los cuales presentan la mismas ventajas que los RLS convencionales pero eficientando el cálculo de las operaciones aritméticas por iteración al orden de p como en el caso de los algoritmos LMS.

El empleo de los algoritmos rápidos se justifica, desafortunadamente se sabe que todos los algoritmos rápidos ponderados exponencialmente ($\lambda < 1$) exhiben un comportamiento inestable y una divergencia repentina debido a errores de redondeo cuando la precisión de cálculo es limitada, por tanto, tal inestabilidad numérica es la principal limitante de la utilización en problemas reales. [21].

Se han propuesto diferentes esquemas para la utilización de estos algoritmos. Por ejemplo en [16] el algoritmo se implementa en paralelo con un proceso de detección, el cual al detectar inestabilidad del algoritmo lo reinicializa, algunas otras técnicas consisten en técnicas de regularización que se usan para estabilizar el algoritmo en situaciones donde se presentan matrices de covarianza mal condicionadas, sin embargo estos algoritmos tienen una limitada capacidad de rastreo, puesto que la ganancia calculada difiere mucho de la óptima. [21].

4.6.1. Algoritmos rápidos estabilizados

Con el propósito de aprovechar las ventajas de los algoritmos rápidos pero evitando los problemas de divergencia se presentan los algoritmos estabilizados, los cuales se basan en un cálculo redundante de cantidades dentro del algoritmo y una retroalimentación del error calculado.

Los algoritmos rápidos no estabilizados más eficientes con respecto al número de operaciones aritméticas presentan $7p$ operaciones por iteración para resolver el problema de filtrado ($5p$ para predicción), en los algoritmos de filtrado estabilizados es posible obtener hasta $8p$ operaciones aritméticas por iteración.[15]. Este aumento es debido a la redundancia introducida, sin embargo este número sigue siendo pequeño con respecto a los algoritmos RLS y por esa razón, es válida la utilización de los algoritmos estabilizados.

4.6.2. Redundancia y retroalimentación del error

Por redundancia nos referimos al cálculo de una misma cantidad de dos maneras diferentes, esto nos permite hacer mediciones de los errores que se generan simplemente obteniendo la diferencia de la misma cantidad calculada de manera diferente.

Las diferencias entre las mismas variables calculadas de maneras diferentes pueden verse como señales de la propagación del error en el algoritmo y podemos usarlas en un mecanismo de retroalimentación con el objetivo de disminuir la propagación del error. La estructura de retroalimentación debe de retroalimentar las señales de error y usar tal retroalimentación para calcular las variables corregidas asociadas al error correspondiente, es decir: tomar el valor final de tales cantidades como una combinación de las dos maneras de calcularlas, e.g:

$$cb(k) = cbs(k) + K(cb(k) - cbs(k)) \quad (4.38)$$

Donde $cbs(k)$ y $cbf(k)$ representan las dos maneras diferentes de calcular el error de predicción backward $cb(k)$ y K es una constante de retroalimentación.[15].

De la ecuación 4.38 podemos decir que si no existen errores de redondeo, entonces las dos maneras de calcular el error backward arrojarán el mismo resultado y por tanto no habrá necesidad de corregir el valor calculado para el error backward. Además, K , constante de retroalimentación, es diferente para cada variable a retroalimentar lo que nos da mayor libertad en afectar la propagación del error y por otra parte, si $K = 0$ entonces no existe retroalimentación y no habrá estabilización alguna.

Un esquema general de la retroalimentación utilizada para la estabilización de los algoritmos rápidos se muestra en la figura 4.2.[15].

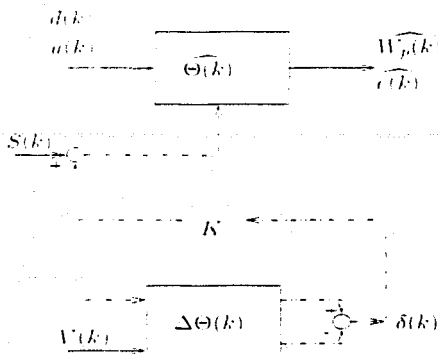


Figura 4.2: Retroalimentación para los algoritmos SFRLS

En el esquema de la figura 4.2, K es un vector que contiene los parámetros de estabilización del algoritmo y cuya dimension depende del numero de señales a retroalimentar. La señal o señales de error se retroalimentan ponderadas por K al sistema representado por $\Delta\Theta(k)$ al cual también se le agrega el error generado en la recursión actual, así es calculada una nueva señal o señales de error. El producto de dicha señal o señales retroalimentadas por el parámetro de estabilización más su correspondiente variable calculada sirve junto con los datos de entrada para calcular un estimado del estado del algoritmo, el cual a su vez proporciona un estimado del error y de los parámetros del filtro.

VARIABLES SENSIBLES A ERRORES NUMÉRICOS

Las variables que han sido identificadas como las más sensibles con respecto a los errores de redondeo y por tanto como las más útiles para medir los errores son.[9]:

- El error de predicción backward $e_p^b(k)$.
- La variable de verosimilitud $\tau_p^b(k)$.
- El último elemento de la ganancia de Kalman al orden $p + 1$, $C_{p-1}^{p-1}(k)$.

Tales variables y las diferentes maneras de calcularse en los algoritmos rápidos se muestran en el cuadro 4.2.

| Variables sensibles a errores numéricos en algoritmos FRLS |
|--|
| <p>Error de predicción backward e_p^b</p> |
| $e_{p1}^b(k) = a^T(k-p) - B_p(k-1)U_p(k)$ $e_{p2}^b(k) = -\lambda\alpha_n(k-1)C_{p-1}^{p-1}(k)$ |
| <p>Variable de verosimilitud γ_p^{-1}</p> |
| $\gamma_p^{-1}(k) = \gamma_{p-1}^{-1}(k) - C_{p-1}^{p-1}(k)e_p^b(k)$ $\gamma_{p-1}^{-1} = 1 - C_p^p(k)U_p(k)$ |
| <p>Ultimo elemento del vector C_{p-1}^{p-1}</p> |
| $C_{p-1}^{p-1}(k) = C_p^p(k-1) - C_{p-1}^1(k)A_p^p(k-1)$ $C_{p-1}^{p-1}(k) = -\lambda^{-1}\alpha_n^{-1}(k-1)e_{p1}^b(k)$ |

Cuadro 4.2: Variables sensibles a errores de redondeo y utilizadas para la estabilización de los algoritmos FRLS.

TESIS CON
FALLA DE ORIGEN

4.6.3. Obtención de algoritmos rápidos estabilizados

ALGORITMO ESTABILIZADO VERSION CIOFFI

En el cuadro 4.4 se presenta la versión estabilizada del algoritmo rápido de Cioffi dado en el apéndice D. La estabilización de este algoritmo utiliza solo una retroalimentación del error de redondeo medido utilizando el calculo redundante del error de predicción backward a priori. La primera forma de calcular el error $e_p^b(k)$ es tal y como se hace en el algoritmo de Cioffi y se requiere de una sola multiplicación para hacerlo, la otra manera de calcularlo es a partir de su definición, esto es:

$$e_p^b(k) = u(k-p) - B_p^T(k-1)U_p(k)$$

Se puede demostrar que la diferencia entre las dos maneras de calcular $e_p^b(k)$ es una función de los parámetros de predicción $A_p(k-1)$ y $B_p(k-1)$. [21].

Con la diferencia de estas dos cantidades podemos obtener una medida del error de redondeo y retroalimentarla de tal forma que se minimice tal error corrigiendo los parámetros de predicción sin que su valor cambie mucho con respecto a los valores reales.

El algoritmo entonces obtiene una medición del error a partir de los parámetros de predicción corregidos como:

$$\overline{\xi_p(k)} = f(\overline{A_p(k-1)}, \overline{B_p(k-1)})$$

Donde " $\overline{}$ " denota cantidades corregidas. $\overline{\xi_p(k)}$ se utiliza para calcular $e_p^b(k)$ corregido y posteriormente calcular con esta variable corregida tanto la ganancia de Kalman Dual al orden $p-1$ como los parámetros de predicción backward, pues se ha observado que para estabilizar el algoritmo no se requiere corregir los parámetros de predicción forward.[21].

El algoritmo se divide básicamente en cuatro partes: Primero se calcula la variable corregida de medición del error $\overline{\xi_p(k)}$, después se calculan las variables corregidas utilizando la medición del error $\overline{\xi_p(k)}$. La tercera parte consiste en calcular las variables que se ocupan en el algoritmo rápido de Cioffi pero utilizando las variables corregidas $\overline{\tau_p(k)}$ y $\overline{C_p(k)}$. Por último se obtienen los parámetros del filtro utilizando $\overline{\tau_p(k)}$ y $\overline{C_p(k)}$ corregidas.

El parámetro ρ , en el algoritmo estabilizado, cuadros 4.3 y 4.4, se relaciona directamente con la corrección hecha a los parámetros de predicción backward y su valor debe ser $\rho \geq 1$ dependiendo del orden del filtro.[21].

ALGORITMO SFRLS VERSIÓN CIOFFI [21]

Condiciones iniciales

$$\alpha_p^l(k-1) = \mu\lambda^p; \quad \alpha_p^h(k-1) = \mu$$

$$\gamma_p(k-1) = 1; \quad A_p(k-1) = B_p(k-1) = W_p(k-1) = C_p(k-1) = U_p(k-1) = 0$$

Cálculo de las variables de corrección $\xi_p(k)$ $\bar{\xi}_p(k)$

$$e_p^l(k) = w(k) - A_p^l(k-1)U_p(k-1)$$

$$e_p^h(k) = u(k-p) - B_p^h(k-1)U_p(k)$$

$$\gamma_{p-1}(k) = \frac{\lambda\alpha_p^l(k-1)}{\lambda\alpha_p^l(k-1) + \gamma_p(k-1)(\alpha_p^l(k))^2} \gamma_p(k-1)$$

$$\gamma_p(k) = \frac{\gamma_{p-1}(k)}{1 + \gamma_{p-1}(k)\alpha_p^h(k)C_p^h(k-1) + \frac{e_p^l(k)\alpha_p^h(k-1)}{\lambda\alpha_p^l(k-1)}}$$

$$k_p(k-1) = \lambda^{-p}\gamma_p(k-1)$$

$$\xi_p(k) = e_p^h(k) - A_p^h(k-1)k_p(k-1)e_p^l(k) - \lambda\alpha_p^h(k-1)C_p^h(k-1)$$

$$\bar{\xi}_p(k) = [1 - \rho(\frac{1}{\gamma_p(k)} - 1) - \rho(A_p^h(k-1)k_p(k-1))^2(\frac{1}{\gamma_p(k-1)} - 1) - 2\rho A_p^h(k-1)k_p(k-1)C_p^h(k-1)e_p^l(k)]^{-1}\xi_p(k)$$

Corrección del filtro $B_p(k-1)$

$$\overline{C_p^l(k)} = (1 - \rho k_p(k-1)C_p^h(k-1)\bar{\xi}_p(k))\overline{C_p^l(k)} - \rho^{\lambda-p}A_p^h(k-1)(1 - \gamma_p(k-1))\bar{\xi}_p(k)$$

$$C_{p-1}(k) = \begin{bmatrix} 0 \\ C_p(k-1) \end{bmatrix} - \begin{bmatrix} 1 \\ A_p(k-1) \end{bmatrix} \lambda^{-1}\alpha_p^h(k-1)e_p^l(k) = \begin{bmatrix} D_{p-1}(k) \\ \mu b(k) \end{bmatrix}$$

$$\begin{bmatrix} \overline{B_p(k-1)} \\ 1 \end{bmatrix} = \frac{1}{1 - \rho\alpha_p^h(k)\bar{\xi}_p(k)} \begin{bmatrix} B_p(k-1) \\ 1 \end{bmatrix} - \rho\bar{\xi}_p(k)C_{p-1}(k)$$

Cuadro 4.3: Algoritmo SFRLS Versión Cioffi

ALGORITMO SFRLS VERSIÓN CIOFFI (CONTINUACIÓN)

$$\overline{c_p^i(k)} = c_p^i(k) - \rho \frac{1}{\tau_p(k)} - 1) \overline{\xi_p(k)}$$

Actualización de la ganancia dual de Kalman $C_p(k)$

$$\varepsilon_p^f(k) = \overline{c_p^f(k)} \tau_p(k-1)$$

$$\alpha_p^f(k) = \lambda \alpha_p^f(k-1) - \varepsilon_p^f(k) \overline{c_p^{fT}(k)}$$

$$A_p(k) = A_p(k-1) - C_p(k-1) \varepsilon_p^{fT}(k)$$

$$\varepsilon_p^h(k) = \overline{c_p^h(k)} \tau_p(k)$$

$$\alpha_p^h(k) = \lambda \alpha_p^h(k-1) - \varepsilon_p^h(k) \overline{c_p^{hT}(k)}$$

$$C_p(k) = D_{\mu}(k) - B_p(k-1) \mu_h(k)$$

$$B_p(k) = \overline{B_p(k-1)} - C_p(k) \varepsilon_p^{hT}(k)$$

Actualización del filtro

$$c_p(k) = c_p(k) - W_p^T(k-1) U_p^i(k)$$

$$\varepsilon_p(k) = c_p(k) \tau_p(k)$$

$$W_p(k) = W_p(k-1) - C_p(k) \varepsilon_p^T(k)$$

Cuadro 4.4: Continuación algoritmo SFRLS Versión Cioffi

ALGORITMO ESTABILIZADO VERSION SLOCK

En los cuadros 4.5 y 4.6 se presenta la versión estabilizada del algoritmo rápido de Slock dado en el apéndice D. Este algoritmo utiliza diversas retroalimentaciones sobre algunas de las variables más sensibles. La primera retroalimentación que utiliza es sobre el último elemento de la ganancia de Kalman dual al orden $p - 1$, $C_{p-1}^{(2)}(k)$. Otra de las variables que se utilizan para la retroalimentación del error es el error de predicción backward a priori $e_p^-(k)$ así como la inversa de la variable de verosimilitud $\gamma_p^{-1}(k)$.

Con estas variables corregidas se obtienen el resto de las variables utilizadas para calcular la ganancia de Kalman dual y los parámetros del filtro, tal y como se hace en el algoritmo de Slock no estabilizado.

Esta versión estabilizada del algoritmo de Slock utiliza diferentes constantes K para cada retroalimentación lo que nos permite tener más grados de libertad para estabilizar el algoritmo en aplicaciones específicas.

TESIS CON
FALLA DE ORIGEN

ALGORITMO SFRLS VERSIÓN SLOCK [15]

Condiciones Iniciales

$$\alpha_p^f(k-1) = \mu\lambda^p; \quad \alpha_p^b(k-1) = \mu$$

$$\gamma_p(k-1) = 1; \quad A_p(k-1) = B_p(k-1) = W_p(k-1) = C_p(k-1) = U_p(k-1) = 0$$

Actualización en orden "Update"

$$e_p^f(k) = u(k) + A_p^T(k-1)U_p(k-1)$$

$$C_{p+1}^f(k) = -\lambda^{-1}\alpha_p^{-f}(k-1)e_p^f(k)$$

$$C_{p+1}[k] = \begin{bmatrix} 0 \\ C_p(k-1) \end{bmatrix} - \begin{bmatrix} 1 \\ A_p(k-1) \end{bmatrix} C_{p+1}^f(k) = \begin{bmatrix} M_p(k) \\ \mu(k) \end{bmatrix}$$

$$\gamma_{p+1}^{-1}(k) = \gamma_p^{-1}(k-1) - C_{p+1}^f(k)e_p^f(k)$$

Predicción Forward

$$\varepsilon_p^f(k) = e_p^f(k)\gamma_p(k-1)$$

$$A_p(k) = A_p(k-1) - C_p(k-1)\varepsilon_p^{fT}(k)$$

$$\alpha_p^{-f} = \lambda^{-1}\alpha_p^{-f}(k-1) - [C_{p+1}^f(k)]^T \gamma_{p+1}^{-1}(k) C_{p+1}^f(k)$$

Actualización en orden "Downdate"

$$e_p^b(k) = -\lambda\alpha_p^b(k-1)C_{p+1}^{fT}(k)$$

$$e_p^{br}(k) = u(k-p) - B(k-1)^T U_p(k)$$

$$[C_{p+1}^{fT}(k)]^* = -\lambda^{-1}\alpha_p^{-b}(k-1)e_p^{br}(k)$$

$$[C_{p+1}^{fT}(k)]^{*r} = C_p^r(k-1) - C_{p+1}^f(k)A_p^r(k-1)$$

Cuadro 4.5: Algoritmo SFRLS Versión Slock.

ALGORITMO SFRLS VERSIÓN SLOCK (CONTINUACIÓN)

$$C_{p-1}^{p-1}(k) = [C_{p-1}^{p-1}(k)]^r - K_4([C_{p-1}^{p-1}(k)]^r - [C_{p-1}^{p-1}(k)]^s)$$

$$[e_p^b(k)]^{k_i} = e_p^b(k) - K_i(e_p^b(k) - e_p^b(k)); \quad i = 1, 2, 5$$

$$C_p(k) = M_p(k) - C_{p-1}^{p-1}(k)B_p(k-1)$$

$$\gamma_p^{-r}(k) = \gamma_{p-1}^{-1}(k) - [C_{p-1}^{p-1}(k)]^r \times [e_p^b(k)]^{k_5}$$

$$\gamma_p^{-f}(k) = 1 - C_p(k)^T U_p(k)$$

$$\gamma_p^{-l}(k) = \gamma_p^{-r}(k) - k_3(\gamma_p^{-f}(k) - \gamma_p^{-r}(k))$$

Predicción Backward

$$[\varepsilon_p^b(k)]^{k_i} = [e_p^b(k)]^{k_i} \gamma_p^{-l}(k); \quad i = 1, 2$$

$$B_p(k) = B_p(k-1) + [\varepsilon_p^b(k)]^{k_1} C_p(k)$$

$$\alpha_p^b(k) = \lambda^{-1} \alpha_p^b(k-1) + [\varepsilon_p^b(k)]^{k_2} [e_p^{bT}(k)]^{k_2}$$

$$\gamma_p(k) = \gamma_p^{-l}(k) - K_5(N \text{ orden}) \alpha_p(k) \alpha_p^{-l}(k) - \gamma_p^{-l}(k)$$

Filtrado

$$e_p(k) = d(k) - W_p^T(k-1)U_p(k)$$

$$\varepsilon_p(k) = e_p(k) \gamma_p(k)$$

$$W_p(k) = W_p(k-1) - C_p(k) \varepsilon_p^T(k)$$

Cuadro 4.6: Continuación algoritmo SFRLS Versión Slock



4.7. Conclusiones

En este capítulo hemos estudiado dos problemas fundamentales que ocasionan divergencia en los algoritmos RLS. El primero de ellos es la pérdida de simetría en la matriz de covarianza, por lo que las implementaciones que suponen simetría sobre la matriz de covarianza, aún implementadas bajo un número grande de bits de precisión, presentarían problemas de divergencia como es el caso de la implementación CLS1. Un análisis teórico de los errores de redondeo nos permite llegar a la conclusión que la implementación CLS2 es suficientemente robusta con respecto a errores de redondeo sin importar la simetría de la matriz de covarianza, siempre que esta se mantenga bien condicionada¹. El otro problema son los errores de redondeo los cuales pueden hacer que un algoritmo diverja incluso si la matriz de covarianza esta bien condicionada.

El empleo de los algoritmos FRLS para eficientar el rendimiento del algoritmo reduciendo el número de operaciones aritméticas para calcular los parámetros del filtro se justifica dada la necesidad de aumentar la velocidad de transmisión. Sin embargo, los algoritmos FRLS son muy sensibles con respecto a los errores de redondeo y presentan problemas de divergencia, por esta razón es necesario estabilizarlos con respecto a estos errores. La estrategia de estabilización presentada aquí y basada en [15] puede ser aplicada a cualquier algoritmo rápido una vez que se han identificado las variables más sensibles en la implementación en estudio.

Por otra parte, un algoritmo estabilizado se podrá utilizar solo si la introducción de redundancia no es tal que el número de operaciones aritméticas aumente tanto que se compare con las operaciones requeridas por los algoritmos RLS clásicos. Además, dado que en estos algoritmos se corrigen los valores de las variables, es importante asegurar que las variables corregidas no difieran mucho de sus valores teóricos pues podría presentarse el caso de obtener un algoritmo estabilizado que convergiera a un valor muy diferente del teórico.

En este capítulo hemos presentado las versiones estabilizadas de los algoritmos FRLS versiones de **Cioffi**, [27] y **Stock**, [15]. Ambas versiones estabilizadas presentan una complejidad de programación mayor a la de sus versiones no estabilizadas puesto que se calculan de manera redundante algunas variables. Sin embargo, ambas presentan una complejidad de $10p$ operaciones por iteración, [21], [15], aumentando solo en 3 operaciones sobre sus versiones no estabilizadas. El precio que se paga para obtener las versiones estabilizadas realmente es pequeño considerando que se obtienen versiones mucho más estables de los algoritmos. Estas versiones nos permitirán incluso implementar tales algoritmos en dispositivos de procesamiento con longitud de palabra finita.

¹Ver apéndice C.

La evaluación de los algoritmos presentados en este capítulo se realizará en el capítulo 5 con el fin de comprobar lo que aquí se ha mencionado.

Capítulo 5

EVALUACIÓN Y COMPARACIÓN DE ALGORITMOS ROBUSTOS DE FILTRADO ADAPTABLE

En este capítulo se muestran las simulaciones que demuestran los problemas de divergencia de los algoritmos basados en el criterio de los mínimos cuadrados debido a los errores de redondeo y a la utilización de factores de olvido $\lambda < 1$, lo que puede ocasionar problemas más severos con respecto a errores numéricos y matrices de covarianza mal condicionadas.

En una primera parte se muestran los problemas de divergencia de los algoritmos RLS clásicos mediante el estudio de dos implementaciones de este tipo de algoritmos y se observa como uno de los principales problemas con estos algoritmos es la pérdida de simetría de la matriz de covarianza.

En la segunda parte de este capítulo se simularon tres versiones de algoritmos rápidos así como versiones estabilizadas de aquellos que son más sensibles a problemas de inestabilidad. Estas simulaciones son hechas en aritmética de punto flotante con precisión sencilla y doble.

En una última parte se presentan las simulaciones correspondientes al algoritmo estabilizado de Slock en aritmética de punto fijo pues fue precisamente este algoritmo el que resultó ser más robusto con base en las simulaciones hechas. Así mismo, se simula el algoritmo Fast Kalman en aritmética de 16 y 32 bits para ciertos valores del factor de olvido λ .

5.1. Algoritmos RLS clásicos

En esta sección se evalúan los algoritmos que se dieron en el capítulo 4 en el cuadro 4.1 y se muestra como un mismo algoritmo puede tener propiedades numéricas muy diferentes dependiendo de la forma en que se implementen las ecuaciones. Siendo algunas implementaciones más robustas que otras.

5.1.1. Implementación CLS1

La implementación CLS1 corresponde al algoritmo clásico recursivo RLS. De acuerdo al análisis de propagación del error hecho en el capítulo 4, para algoritmos clásicos recursivos llegamos a la conclusión que los errores en la implementación CLS1 podían crecer de manera infinita si la matriz de covarianza no se conserva simétrica durante todo el proceso.

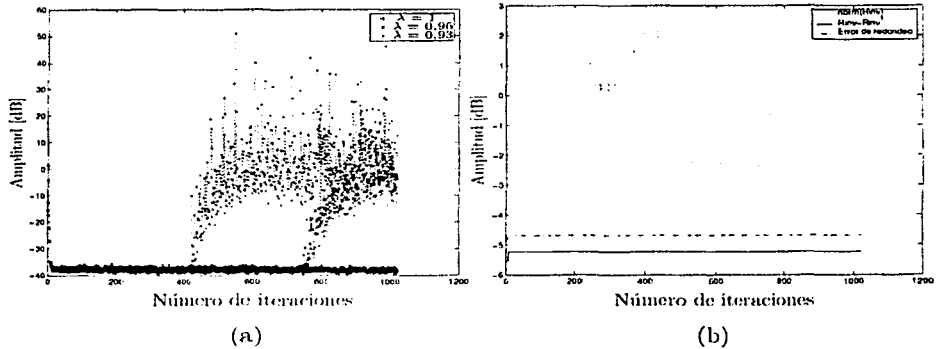


Figura 5.1: Implementación CLS1. (a) Comportamiento del error de filtrado, $\lambda = 1, 0.96, 0.93$. (b) Comportamiento de la norma de la matriz de covarianza $R_p^{-1}(k)$ para $\lambda = 1$.

La figura 5.1(a) muestra el error de filtrado para distintos factores de olvido λ . Con base en esta figura podemos apreciar el comportamiento sumamente inestable de esta implementación cuando se utilizan factores de olvido $\lambda < 1$. Podemos apreciar que entre mas pequeño sea el factor de olvido los problemas de divergencia son aún mayores.

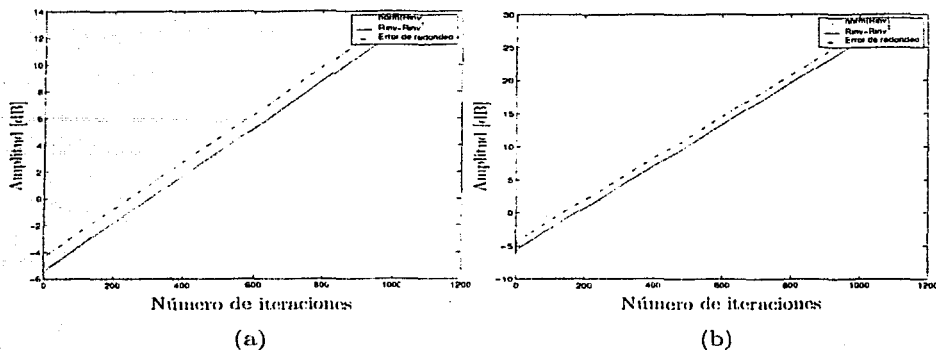


Figura 5.2: Comportamiento de la norma de la matriz de covarianza $R_p^{-1}(k)$ para la implementación CLS1. (a) $\lambda = 0,96$. (b) $\lambda = 0,93$.

Para comprender a detalle lo que sucede con esta implementación en las figuras 5.1(b) y 5.2 se muestra el comportamiento de la norma de la matriz de covarianza a lo largo del proceso así como el error acumulado.

En la figura 5.1(b) podemos apreciar como para $\lambda = 1$ el error de redondeo no crece durante el proceso, esto es debido a que la matriz de covarianza se mantiene casi simétrica, ($R_p^{-1}(k) - R_p^T(k) \approx 0$). esto da como resultado que el algoritmo converja a su valor óptimo. Para verificar que efectivamente el algoritmo converge a su valor óptimo en la figura 5.3 (a) y (b) se muestra la respuesta en frecuencia y la convolución de la respuesta al impulso del canal con el filtro respectivamente del algoritmo. En la figura 5.3 (a) podemos apreciar como la respuesta en frecuencia de nuestro algoritmo es exactamente igual a la respuesta en frecuencia del canal, de manera equivalente en la figura 5.3 (b), observamos que la respuesta al impulso es precisamente un impulso lo que nos muestra que se ha eliminado la interferencia entre símbolos.

Para realizar la medición del error de redondeo, el algoritmo se corrió en aritmética flotante de precisión sencilla y de precisión doble, considerando las variables calculadas con doble precisión como libres de errores y de esta manera se calcula un error de redondeo como la diferencia de las variables calculadas con doble precisión menos las variables calculadas con precisión sencilla. En el caso especial de las figuras 5.1 (b) y 5.2 se muestra el error con respecto a la matriz de covarianza.

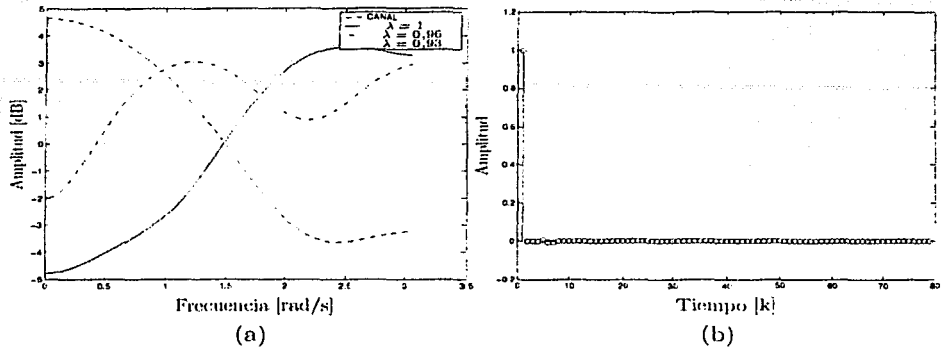


Figura 5.3: Implementación CLS1. (a) Respuesta en frecuencia. (b) Convolución $h_c * h_f$ para $\lambda = 1$.

La figura 5.3 (a) muestra la respuesta en frecuencia de esta implementación para los diferentes λ 's, claramente se reafirma lo observado en la figura 5.1 (a) y podemos decir que salvo para $\lambda = 1$ esta implementación es inestable con respecto a errores de redondeo.

El hecho que ocasiona la divergencia es claramente la pérdida de simetría de la matriz de covarianza (ver figuras 5.1 (a) y 5.2). Esto a su vez es ocasionado por un gran aumento en la norma de la matriz de covarianza y en consecuencia una pérdida debido a la representación de su valor en la aritmética usada.

En conclusión podemos decir que los factores de olvido menores a la unidad ocasionan por una parte un aumento desmedido de la norma de la matriz de covarianza lo cual a su vez se ve representado en una pérdida de simetría de la matriz y en consecuencia un aumento desmedido de los errores de redondeo los cuales hacen diverger al algoritmo.

5.1.2. Implementación CLS2

Con base en el análisis del capítulo 4, la implementación CLS2 resultó ser más robusta con respecto a errores numéricos de redondeo. En las siguientes simulaciones

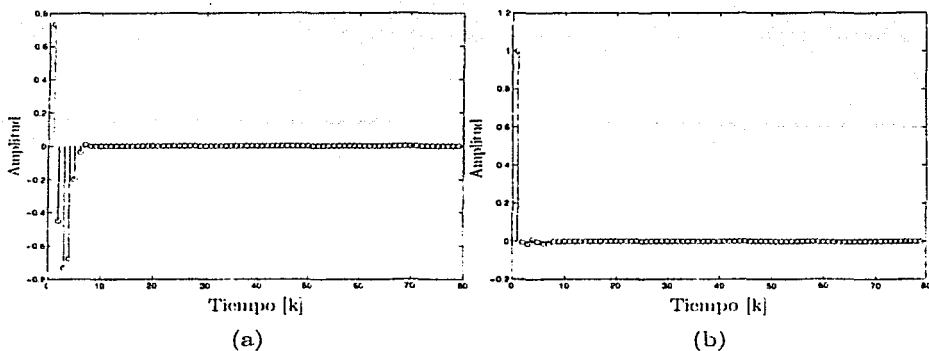


Figura 5.4: Implementación CLS1, convolución $h_c * h_f$ (a) $\lambda = 0,93$. (b) $\lambda = 0,96$.

se muestra el comportamiento de tal implementación.

La figura 5.5(a) muestra el comportamiento del error de filtrado para diferentes factores de olvido λ . A diferencia de la implementación CLS1, esta implementación resulta ser mucho más robusta como se muestra en dicha figura.

Para conocer las causas del buen comportamiento de esta algoritmo en las figuras 5.5 (b) y 5.6 se muestra el comportamiento de la norma de la matriz de covarianza a lo largo del proceso. En estas figuras se aprecia claramente como la matriz de covarianza se mantiene simétrica durante todo el proceso, $(R_p^{-1}(k) - R_p^T(k) = 0)$, para cualquier factor de olvido utilizado.

Así mismo, la norma de la matriz se mantiene estable y pequeña, por tal razón no existe una propagación de errores de redondeo y el algoritmo converge independientemente de la λ utilizada.

En la figura 5.7 (a) se muestra la respuesta en frecuencia del algoritmo y observamos claramente que ésta es invariante con respecto a el factor de olvido utilizado. Así mismo podemos apreciar que los parámetros del filtro en esta implementación convergen al valor óptimo pues la respuesta en frecuencia del filtro corresponde precisamente a la inversa de la respuesta del canal.

Por último, en las figuras 5.7 (b) y 5.8 se muestra la convolución de la respuesta al

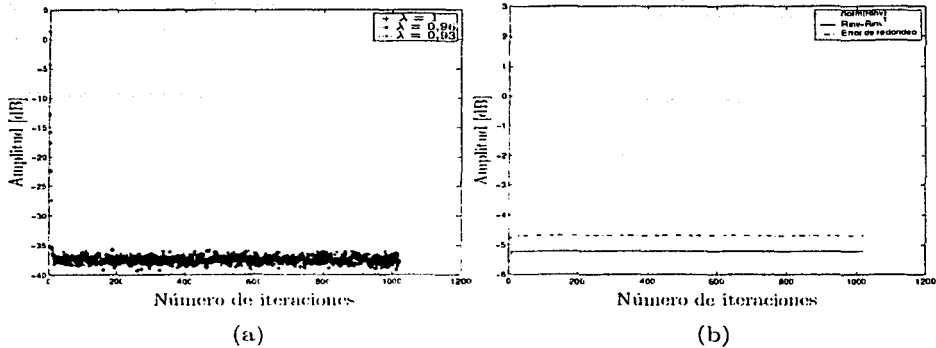


Figura 5.5: Implementación CLS2. (a) Comportamiento del error de filtrado. (b) Comportamiento de la norma de matriz de covarianza $R_p^{-1}(k)$ para $\lambda = 1$.

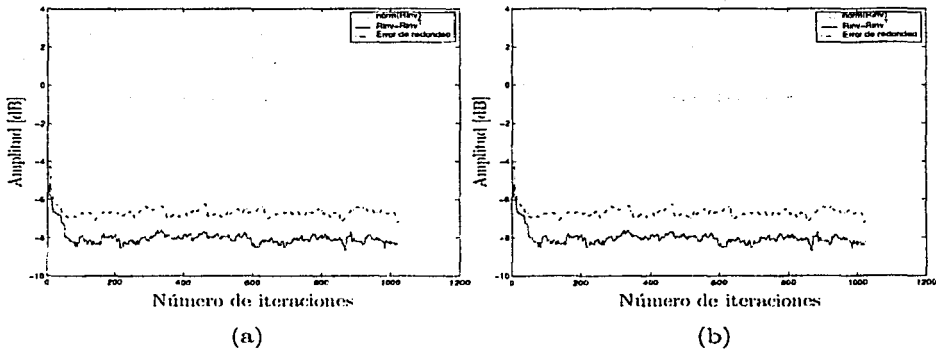


Figura 5.6: Comportamiento de la norma de la matriz de covarianza $R_p^{-1}(k)$ para la implementación CLS2. (a) $\lambda = 0.96$. (b) $\lambda = 0.93$.

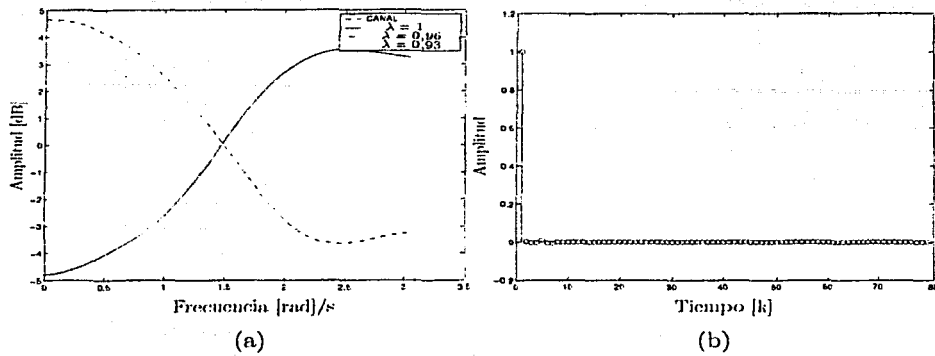


Figura 5.7: Implementación CLS2. (a) Respuesta en frecuencia. (b) Convolución $h_c * h_f$ para $\lambda = 1$.

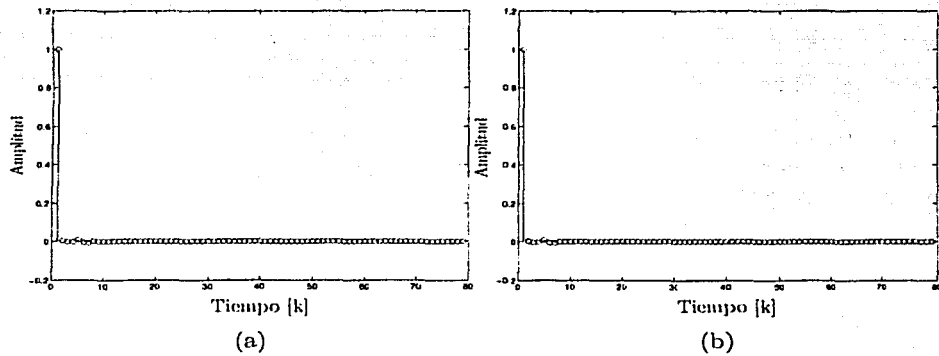


Figura 5.8: Implementación CLS2, convolución $h_c * h_f$ (a) $\lambda = 0,96$. (b) $\lambda = 0,93$.

impulso del canal con la respuesta al impulso del filtro igualador constatando que se elimina completamente la interferencia entre símbolos en esta implementación sin importar la λ que se utilice.

5.2. Algoritmos FRLS con precisión doble

Como dijimos en el capítulo 4, los algoritmos FRLS presentan problemas de divergencia debido a errores de redondeo sobre todo cuando se trabaja con factores de olvido $\lambda < 1$. En esta sección mediante las simulaciones hechas se demuestra tal afirmación y podemos incluso destacar que a pesar de que la precisión utilizada es doble (64 bits) algunos de estos algoritmos divergen lo cual demuestra la necesidad de obtener algoritmos estabilizados que nos garanticen la estabilidad con respecto a errores numéricos al utilizar aritmética finita.

5.2.1. Algoritmo Rápido de Kalman

El algoritmo rápido de Kalman deducido en el capítulo 3 y dado en el cuadro 3.3 fue el primer algoritmo rápido que se utilizó y de manera general podemos decir que sus propiedades numéricas son mejores que las de los algoritmos subsecuentes como la versión de Cioffi o Slock estudiadas en el capítulo 4. En esta sección se implementa este algoritmo en aritmética flotante de precisión doble y se compara con las versiones de Cioffi y Slock.

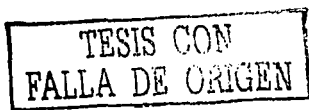
La figura 5.9 (a) muestra el comportamiento del error de filtrada en el algoritmo rápido de Kalman bajo diferentes factores de olvido. De acuerdo con la figura, esta implementación resulta ser estable en aritmética flotante de 64 bits pues sin importar el factor de olvido, el error de filtrado converge prácticamente a cero.

Como mencionamos en el capítulo 4 la variable de verosimilitud juega un papel importante en el comportamiento de los algoritmos rápidos, por esta razón, a pesar de que el algoritmo Fast Kalman no utiliza la variable de verosimilitud para obtener los parámetros, hemos graficado el comportamiento de esta variable.

El comportamiento de γ se muestra en la figura 5.9 (b), podemos observar que ésta se estabiliza alrededor de la unidad y el algoritmo se mantiene estable.

5.2.2. Algoritmo Rápido Versión Cioffi

En esta sección se muestran las simulaciones correspondientes al algoritmo rápido versión Cioffi dado en el apéndice D implementado bajo precisión doble en aritmética



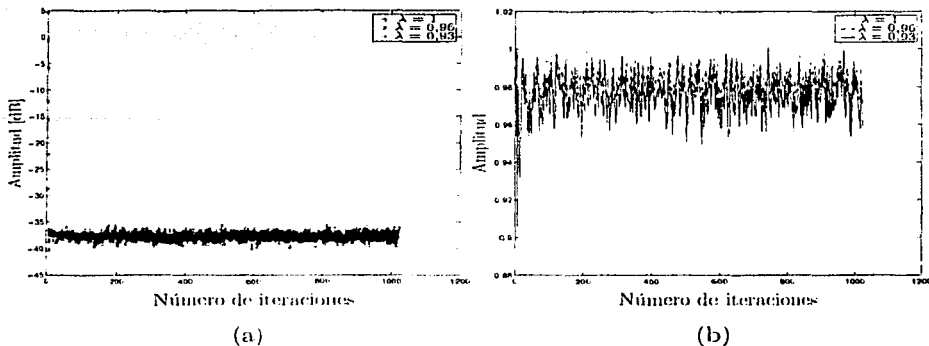


Figura 5.9: Algoritmo Rápido Fast Kalman en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

flotante. Esta versión resulta ser aparentemente mas atractiva que la versión del algoritmo "Fast Kalman" pues requiere solo $7p$ operaciones aritméticas por iteración para calcular los coeficientes del filtro, contra $8p$ del algoritmo "Fast Kalman", donde p es el orden del filtro.

Como se verá en las simulaciones, esta ganancia en tiempo de cálculo termina por hacer esta implementación mucho más sensible a errores numéricos.

La figura 5.10 (a) muestra el comportamiento del error de filtrado en esta implementación. En tal figura podemos apreciar lo inestable de esta implementación pues aún para factores de olvido iguales a la unidad, lo que asegura matrices de covarianza bien condicionadas, esta implementación diverge.

Una de las causas de esta divergencia es, como se menciona en el capítulo 4 (Ver cuadro 4.2), la gran sensibilidad a errores de redondeo de la variable de verosimilitud γ . El comportamiento de γ se muestra en la figura 5.10 (b) en donde podemos observar que sin importar el valor de λ esta variable converge a cero rápidamente, por lo que podemos pensar que la causa de la divergencia es precisamente el comportamiento numérico de γ .

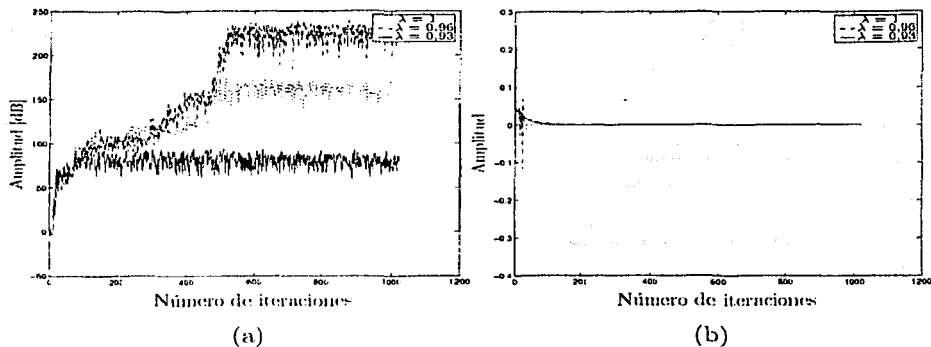


Figura 5.10: Algoritmo Rápido versión Cioffi en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

5.2.3. Algoritmo Rápido Versión Slock

En esta sección se muestran las simulaciones correspondientes al algoritmo rápido versión Slock dado en el apéndice D implementado bajo precisión doble en aritmética flotante. Al igual que en el caso de la versión de Cioffi, la aparición de esta nueva implementación tiene como objetivo disminuir aún más el tiempo de cálculo para los parámetros del filtro. Esta implementación al igual que la de Cioffi actualiza los parámetros utilizando $7p$ operaciones aritméticas por iteración.

Es importante analizar el comportamiento numérico de tal implementación y para hacerlo se presentan la siguiente simulación del algoritmo.

Comparando la figura 5.10 (a) con la figura 5.11 (a) podemos apreciar que la versión rápida de Slock es más robusta cuando menos para $\lambda = 1$, sin embargo para $\lambda \neq 1$ este algoritmo diverge. Sin embargo, para $\lambda = 1$, el algoritmo no es capaz de seguir las variaciones del canal.

La causa de la divergencia es precisamente la misma que en algoritmo versión Cioffi y se muestra en la figura 5.11 (b) en donde apreciamos el comportamiento numérico de γ . En esta última figura podemos apreciar que cuando el algoritmo converge ($\lambda = 1$), la variable de verosimilitud γ se estabiliza en un valor cercano a la unidad. Esto nos hace pensar que, dado que γ relaciona los errores de predicción a priori

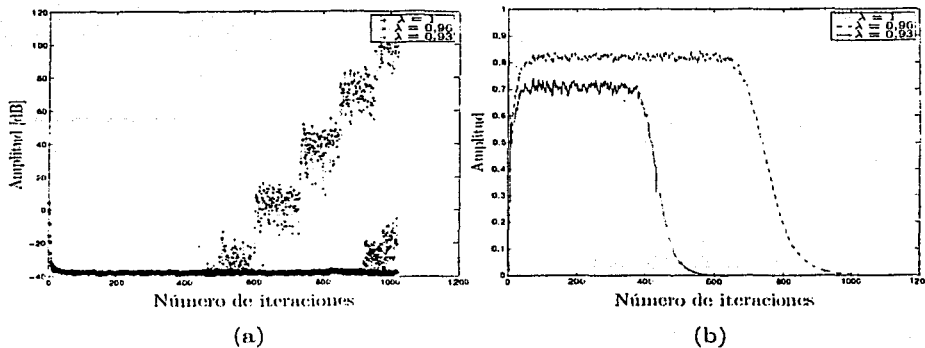


Figura 5.11: Algoritmo Rápido versión Slock en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

y a posteriori, cuando el algoritmo converge, tales errores además de ser pequeños son semejantes, pues los parámetros de un instante a otro no cambian cuando el algoritmo a convergido a su valor óptimo.

Por otra parte, dado que para asegurar convergencia el valor de γ debe ser cercano a la unidad, podemos pensar que en los algoritmos estabilizados presentados en la siguiente sección presenten un comportamiento de γ alrededor de la unidad.

5.3. Algoritmos SFRLS con precisión doble

En esta sección se presentan los algoritmos estabilizados de los algoritmos rápidos versiones Cioffi y Slock dados en el apéndice D. Las simulaciones se presentan en punto flotante de precisión doble y se puede apreciar como efectivamente estos algoritmos son más robustos que los simulados en la sección 5.2. Al igual que en las secciones anteriores, las simulaciones se realizan utilizando diferentes factores de olvido para observar la dependencia de los algoritmos con respecto a λ .

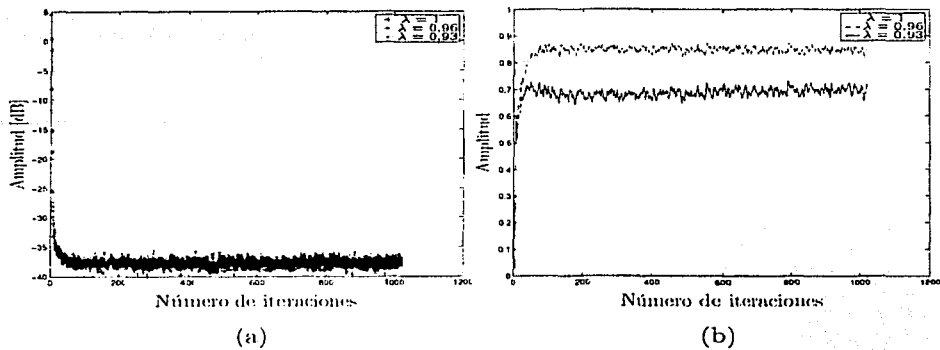


Figura 5.12: Algoritmo Rápido Estabilizado versión Cioffi (algoritmo de Botto.) en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

5.3.1. Algoritmo Estabilizado Rápido Versión Cioffi, SFRLSVC.

En las figuras 5.12 (a) y (b) se muestran respectivamente el comportamiento de la energía del error de filtrado y el comportamiento de γ en el algoritmo estabilizado propuesto en [21] para la versión del algoritmo rápido de Cioffi. Se puede apreciar como este algoritmo estabiliza de manera eficiente el algoritmo de Cioffi manteniendo el valor de la variable γ dentro de un rango cercano a la unidad.

A pesar de que las gráficas muestran un comportamiento bastante estable del algoritmo propuesto por Botto, por medio de diferentes simulaciones se observó que para valores de $\lambda < 0.96$ el algoritmo presenta severos problemas de estabilidad, lo cual reafirma lo asegurado en [15] con respecto al algoritmo de Botto en relación a problemas de inestabilidad.

Sin embargo, esta versión estabilizada puede utilizarse en aplicaciones donde las variaciones del canal no sean muy severas y se puedan usar valores de λ cercanos a la unidad que permitan asegurar estabilidad.

El problema fundamental con el algoritmo de Botto es una divergencia repentina de la variable de verosimilitud a valores negativos y por tanto una divergencia inmediata del algoritmo. Cabe resaltar que para obtener las gráficas de la figura 5.12 para valores de $\lambda < 0.96$ se hizo una selección de señales que permitieran hacer la

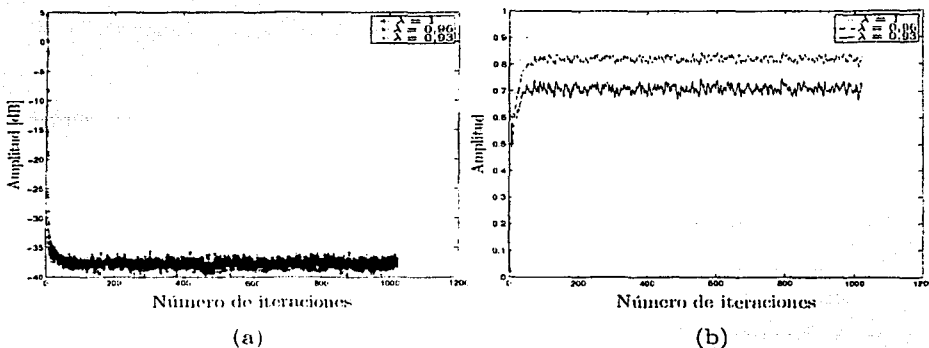


Figura 5.13: Algoritmo Rápido Estabilizado versión Slock en aritmética flotante con precisión doble. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

simulación sin que estas divergieran con el objetivo de mostrar como se comporta el algoritmo cuando γ se logra estabilizar para $\lambda < 0,96$.

5.3.2. Algoritmo Estabilizado Rápido Versión Slock, SFRLSVS

El comportamiento del algoritmo estabilizado propuesto por Slock, [15] se muestra en la figura 5.13. Claramente podemos apreciar el buen comportamiento de este algoritmo independientemente del factor de olvido utilizado.

Este algoritmo, a diferencia del presentado en la sección 5.3.1 presenta un comportamiento bastante estable. Podemos decir que probablemente el buen comportamiento de este algoritmo se debe a que este utiliza varias retroalimentaciones de los errores de redondeo utilizando tres de las variables más sensibles con el propósito de corregir dichos errores.

5.4. Algoritmos FRLS con precisión sencilla

Ahora, con el objetivo de observar más claramente los problemas de divergencia debido a errores numéricos, en esta sección se presentan los algoritmos rápidos no

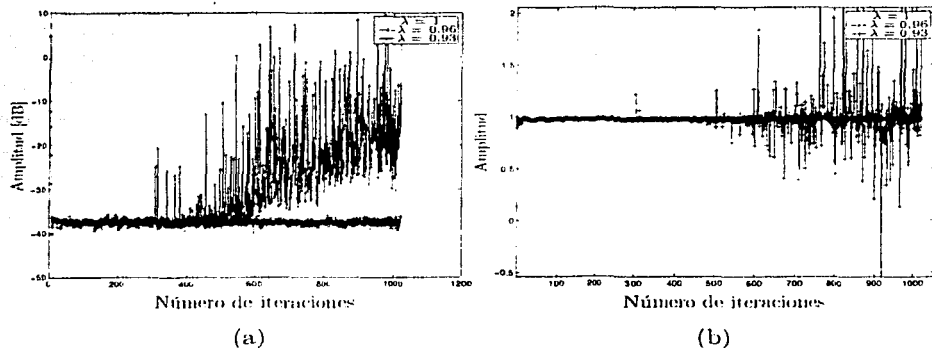


Figura 5.14: Algoritmo Rápido Fast Kalman en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

estabilizados, simulándolos en aritmética flotante de precisión sencilla (32bits). Los resultados obtenidos demuestran los problemas de divergencia que se presentan cuando se reduce a un más la precisión utilizada.

5.4.1. Algoritmo Rápido de Kalman

El algoritmo "Fast Kalman" exhibió un comportamiento bastante estable en aritmética de precisión doble, sin embargo; en la figura 5.14 (a) se muestra cómo este mismo algoritmo presenta problemas de divergencia cuando se reduce el tamaño de la palabra.

A pesar de la inestabilidad que presenta debido a errores de redondeo, este algoritmo se comporta bastante bien para λ 's cercanas a la unidad. Esta implementación a pesar de que requiere $8p$ multiplicaciones por iteración presenta un comportamiento mucho más estable que la versión de Cioffi y que la presentada por Slock.

Sin embargo, como se observa en la figura 5.14 este algoritmo presenta problemas de estabilidad para valores de $\lambda < 1$ en aritmética flotante de precisión sencilla.

El comportamiento de γ se muestra en la figura 5.14 (b), podemos apreciar que jus-

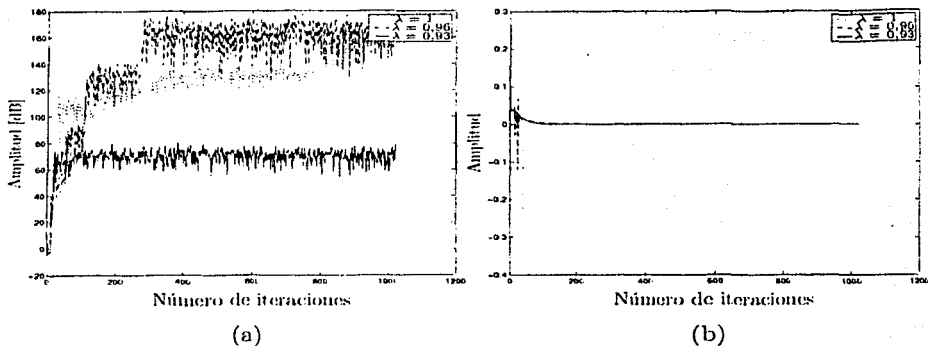


Figura 5.15: Algoritmo Rápido versión Cioffi en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

tamente cuando el algoritmo comienza a diverger, la variable de verosimilitud toma valores mayores a 1, es decir si la variable γ crece demasiado el algoritmo pierde estabilidad. esto confirma lo afirmado en el capítulo 4 con respecto a la sensibilidad de la variable de verosimilitud en los algoritmos rápidos.

Una versión estabilizada de este algoritmo mediante una corrección de los errores de predicción backward se puede observar en [26], aquí, la corrección sobre el error de predicción backward se utiliza para calcular los parámetros Backward. La estabilidad del algoritmo propuesto en [26] continúa dependiendo del factor de olvido, sin embargo, presenta un comportamiento más estable que la versión no estabilizada del algoritmo Fast Kalman y solo introduce dos multiplicaciones por iteración.

5.4.2. Algoritmo Rápido Versión Cioffi

En la figura 5.15, el algoritmo de Cioffi, de por si inestable en aritmética de precisión doble, reafirma su gran inestabilidad al ser implementado en precisión sencilla y por tanto la necesidad de obtener una versión estable para que pueda ser utilizado.

La causa de la divergencia se aprecian claramente en la figura 5.15 (b) y es precisamente que la variable de verosimilitud diverge a cero e incluso a valores negativos.

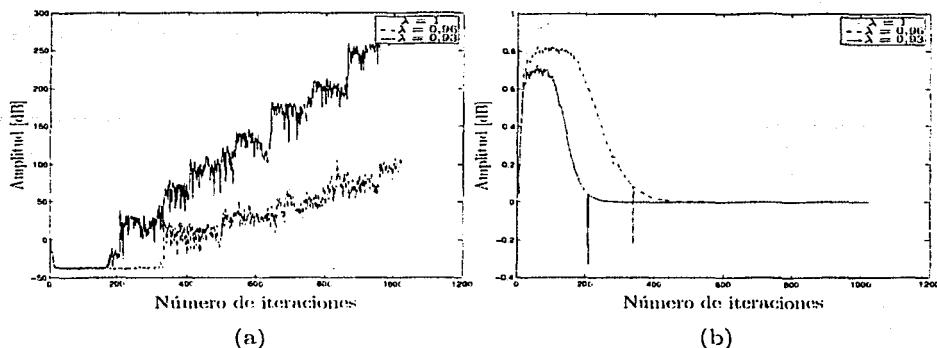


Figura 5.16: Algoritmo Rápido versión Slock en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

5.4.3. Algoritmo Rápido Versión Slock

Como se puede observar en la figura 5.16 (a), el algoritmo de Slock se comporta estable incluso en aritmética sencilla para valores de $\lambda = 1$, sin embargo, si el factor de olvido es menor a la unidad, los problemas de divergencia aparecen de inmediato y la implementación no se puede utilizar.

Comparando las figuras 5.11(a) y 5.16 (a) claramente podemos observar como los errores de redondeo se presentan más rápido cuando se utilizan menos bits de precisión pues en la figura 5.11(a) se utilizó precisión doble y la implementación presentó problemas de divergencia después de las 400 muestras mientras que en la implementación con precisión sencilla (figura 5.16 (a)) la divergencia se presenta casi de inmediato.

Además, como podemos observar, los problemas numéricos que ocasionan divergencia son mas severos conforme se utilizan factores de olvido más pequeños.

La causa de la divergencia, al igual que en la mayoría de algoritmos que involucran la variable de verosimilitud, es que ésta diverge a cero y a valores negativos. Tal comportamiento se observa en la figura 5.16 (b).

Con el propósito de poder utilizar estos algoritmos en ambientes donde las variaciones del canal requieran la utilización de $\lambda's < 1$, en la siguiente sección se evalúan

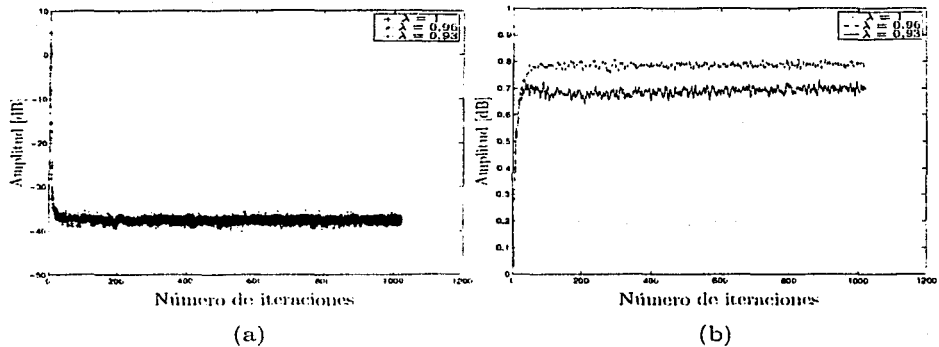


Figura 5.17: Algoritmo Rápido Estabilizado versión Cioffi (algoritmo de Botto) en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

los algoritmos estabilizados de estas versiones bajo precisión sencilla en aritmética flotante.

5.5. Algoritmos SFRLS con precisión sencilla

En esta sección se simulan los algoritmos estabilizados ahora bajo precisión sencilla y observamos su comportamiento con respecto a los algoritmos simulados en la sección 5.4. Con base en las simulaciones podemos confirmar el buen desempeño de los algoritmos estabilizados utilizados.

5.5.1. Algoritmo Estabilizado Rápido Versión Cioffi, SFRLSVC

El comportamiento del algoritmo estabilizado de Cioffi en aritmética de precisión sencilla se muestra en la figura 5.17. Tal implementación estabiliza la variable de verosimilitud a valores cercanos a la unidad, exhibiendo este algoritmo un comportamiento estable.

Como se mencionó en la sección 5.3.1, este algoritmo puede presentar problemas de inestabilidad incluso con precisión doble cuando se utilizan λ 's $< 0,96$, por esta

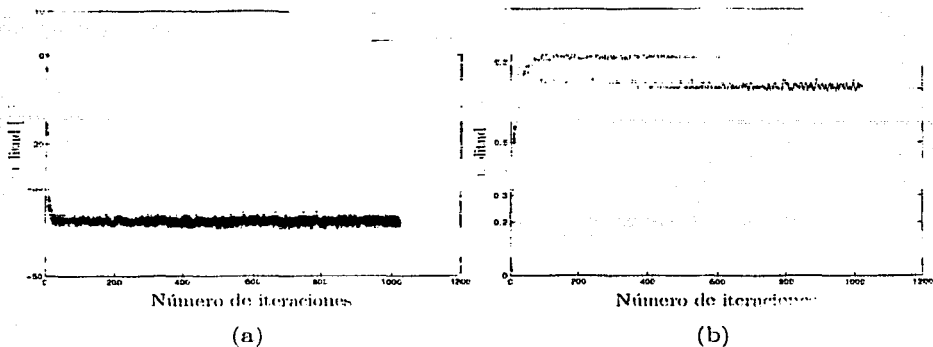


Figura 5.18: Algoritmo Rápido Estabilizado versión Slock en aritmética flotante con precisión sencilla. (a) Comportamiento del error de filtrado. (b) Comportamiento de la variable de verosimilitud γ .

razón se recomienda usar este algoritmo para valores de λ cercanos a la unidad. En [21] este algoritmo se simuló para 100,000 muestras con $\lambda = 0,96$ sin presentar problemas de divergencia.

5.5.2. Algoritmo Estabilizado Rápido Versión Slock, SFRLSVS

El comportamiento del algoritmo estabilizado de Slock implementado en aritmética flotante de precisión sencilla se muestra en la figura 5.18. Podemos apreciar que este algoritmo es robusto para cualquier valor de λ tanto en precisión doble como sencilla.

El algoritmo estabilizado de Slock resulta ser el más robusto de los algoritmos estudiados y por esta razón en la sección 5.6 se estudia el comportamiento de esta algoritmo en aritmética entera.

La figura 5.19 muestra el comportamiento frecuencial de los algoritmos estabilizados donde podemos apreciar que efectivamente la solución a la que convergen es la óptima, pues la respuesta del filtro es precisamente la inversa que la del canal.

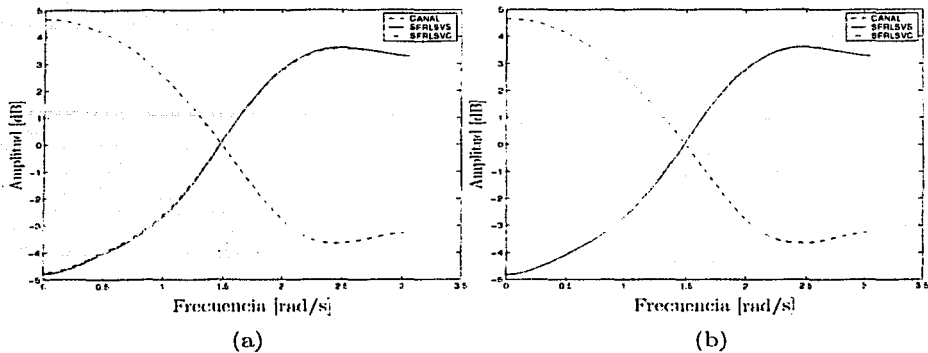


Figura 5.19: Respuesta en frecuencia de los algoritmos rápidos estabilizados, Cioffi-Botto y Slock, en precisión sencilla. (a) $\lambda = 0.96$. (b) $\lambda = 0.93$.

5.6. Algoritmos FRLS en aritmética entera

Después de haber realizado distintas simulaciones de los algoritmos estabilizados hemos observado que tanto el algoritmo estabilizado versión Cioffi como el algoritmo estabilizado versión Slock se comporta adecuadamente en aritmética flotante de 32 bits, sin embargo el algoritmo estabilizado versión Slock puede incluso implementarse en aritmética entera exhibiendo un buen comportamiento. Por esta razón en esta sección presentamos el algoritmo estabilizado de Slock en aritmética entera de 32 y 16 bits.

El algoritmo "Fast Kalman" a pesar de no presentar estabilización alguna, pudimos observar que este presenta buenas propiedades numéricas cuando se utilizan factores de olvido cercanos a la unidad. En esta sección presentamos una simulación del algoritmo "Fast Kalman" en aritmética entera de punto fijo en 16 y 32 bits.

5.6.1. Algoritmo SFRLSVS en aritmética entera

El algoritmo SFRLSVS puede ser implementado en aritmética de 16 bits, en donde, uno de los principales problemas es la dinámica de algunas de las variables. Una manera de disminuir la dinámica de las variables es trabajar con factores de olvido pequeños ($\lambda's < 0,95$), esta solución es poco efectiva pues como hemos visto; para factores de olvido pequeños los problemas de inestabilidad son mayores. Por esta

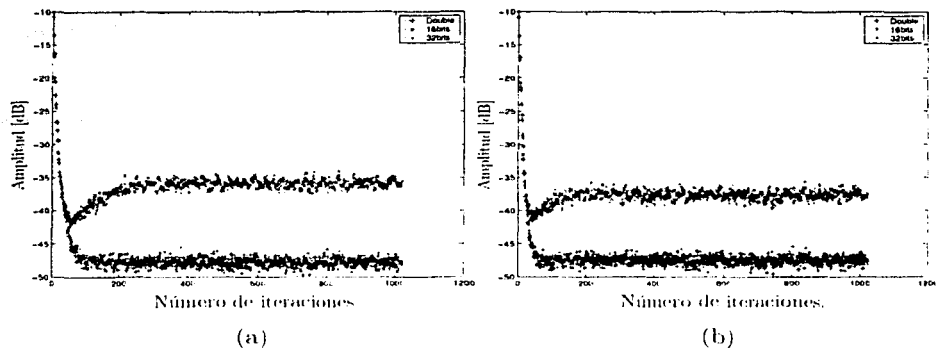


Figura 5.20: Comportamiento del error de filtrado del algoritmo SFRLSVS en aritmética entera. (a) $\lambda = 0.98$. (b) $\lambda = 0.95$.

razón es importante trabajar con λ 's cercanas a la unidad. En particular, las variables que exhiben una dinámica mayor son tanto la energía residual backward α_p^b como la inversa de γ por lo que nuestro criterio para el número de bits utilizados para parte entera se baso precisamente en los valores máximos de estas dos variables, esto con el propósito de evitar la existencia de saturaciones, las cuales pueden causar problemas numéricos más graves.

La figura 5.20 muestra el comportamiento del error de filtrado para diferentes precisiones. Como era de esperarse el algoritmo implementado en 16 bits converge a un valor mayor del error de filtrado dado que el tamaño de la palabra no permite que este llegue a valores menores, sin embargo el comportamiento que presenta es estable. Podemos también observar que tanto para $\lambda = 0,98$ como para $\lambda = 0,95$ el comportamiento del algoritmo es estable.

En cuanto a los algoritmos en 32 bits y precisión doble, prácticamente no existe diferencia en el error al que convergen y podemos decir que los errores de redondeo no afectan el comportamiento del error de filtrado.

Como hemos visto, el comportamiento de la variable de verosimilitud es de suma importancia para la estabilidad del algoritmo, por esta razón en la figura 5.21 se muestra tal comportamiento. Podemos observar que tanto para la implementación en 16 bits como para la de 32 bits el comportamiento de esta variable es estable, lo

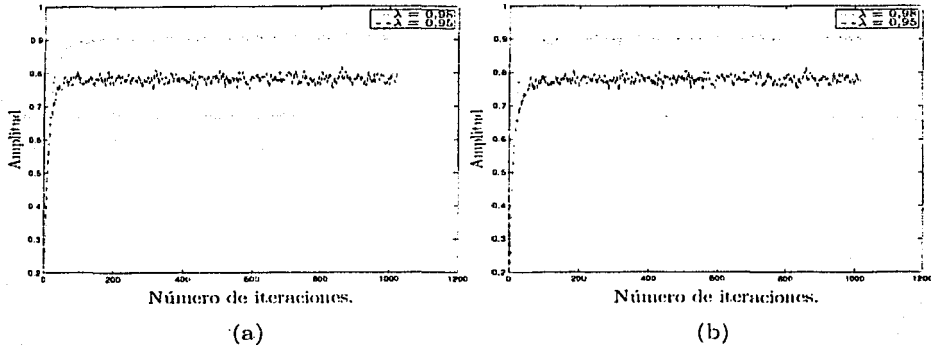


Figura 5.21: Comportamiento de la variable de verosimilitud en el algoritmo SFRLSVS en aritmética entera. (a) 32 bits. (b) 16 bits.

cual confirma la estabilidad del algoritmo.

En la figura 5.22 se muestra el error de redondeo sobre el error de filtrado. Para obtener esta gráfica se simuló el algoritmo en aritmética de precisión doble, considerando que las variables calculadas con doble precisión son libres de errores y luego se realiza la resta de las variables sin errores menos las variables calculadas en aritmética entera. Podemos observar claramente como el error de redondeo es mayor en la implementación de 16 bits esto es debido a la menor precisión con la que se trabaja. Sin embargo, el algoritmo estabilizado no permite que estos errores crezcan y podemos observar como estos se mantienen constantes sin hacer diverger el algoritmo. Así mismo, en la figura 5.23 se muestran los errores de redondeo sobre la ganancia de Kalman dual, en tal figura podemos reafirmar la estabilidad de los errores de redondeo a lo largo de todo el proceso lo que permite tener un algoritmo numéricamente estable.

Por último, para comprobar que tanto las implementaciones en 16 bits como en 32 bits convergen a los parámetros óptimos del filtro, en la figura 5.24 se muestra el comportamiento frecuencial de la respuesta del filtro. En dicha figura podemos observar como, en el caso de la implementación en 32 bits, la respuesta del filtro igualador corresponde exactamente a la inversa de la respuesta del canal tanto para $\lambda = 0.98$ como para $\lambda = 0.95$. En lo que respecta a la implementación en 16 bits, podemos decir que los errores de redondeo causan la diferencia entre los parámetros

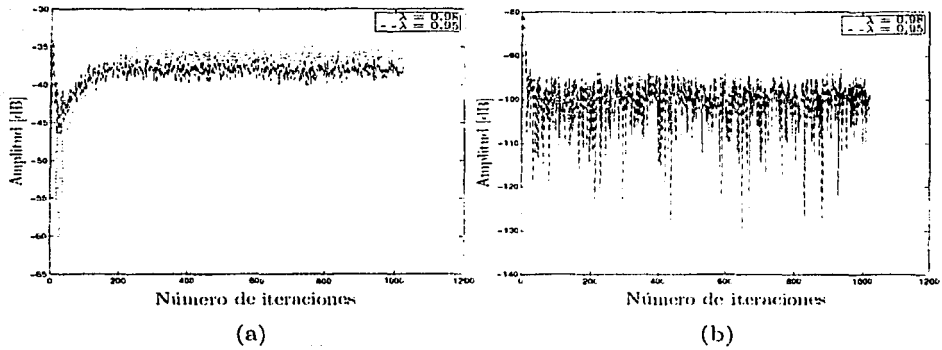


Figura 5.22: Comportamiento de los errores de redondeo en el error de filtrado $e(k)$. (a) aritmética de 16 bits. (b) aritmética de 32 bits.

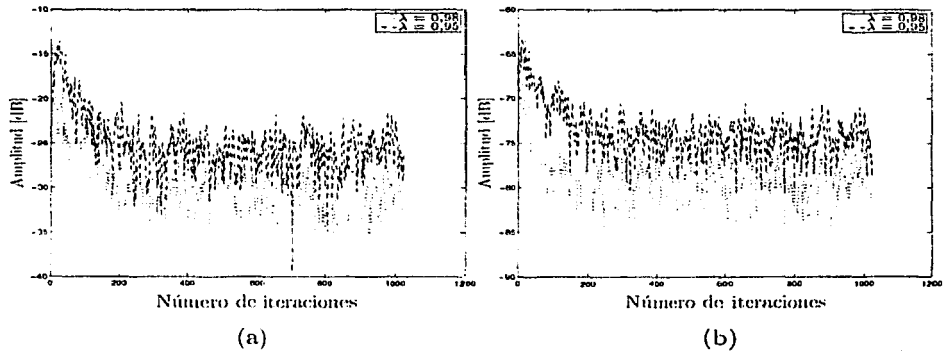


Figura 5.23: Comportamiento de los errores de redondeo en la ganancia dual de Kalman $C_p(k)$. (a) aritmética de 16 bits. (b) aritmética de 32 bits.

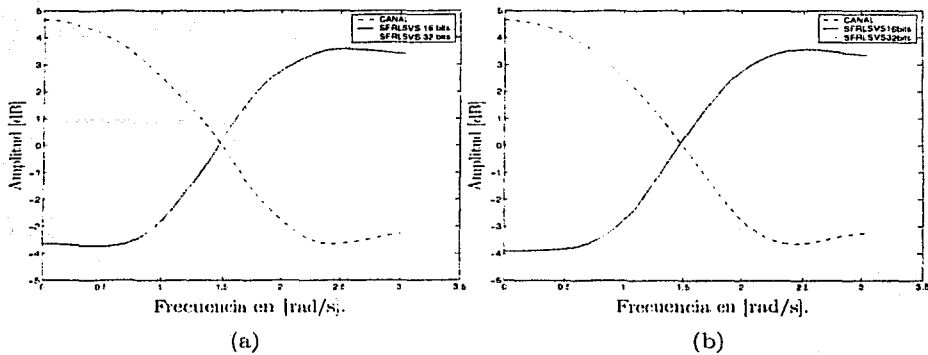


Figura 5.24: Respuesta en frecuencia de la implementación SFRLSVS en aritmética entera (a) $\lambda = 0.98$. (b) $\lambda = 0.95$.

ros óptimos y los estimados; sin embargo podemos decir que el desempeño de esta implementación es adecuado dado que las diferencias observadas en la respuesta en frecuencia son mínimas.

5.6.2. Algoritmo Fast Kalman en aritmética entera

Como hemos mencionado y demostrado a través de las simulaciones en este capítulo, el algoritmo "Fast Kalman" presenta buenas propiedades numéricas si se trabaja con factores de olvido cercanos a la unidad. Dicho comportamiento numérico puede conservarse incluso en aritmética de punto fijo bajo ciertas consideraciones sobre λ .

El factor de olvido juega un papel muy importante en la implementación de los algoritmos RLS pues, por un lado, si se trabaja con λ 's muy pequeñas estos algoritmos presentan problemas de inestabilidad incluso en precisión infinita. Sin embargo, trabajar con λ 's cercanas a la unidad ocasiona una dinámica numérica mayor en las variables involucradas, sobre todo para las energías residuales α_p^f y α_p^b . Por tal razón es importante encontrar para que valores de factores de olvido los algoritmos pueden implementarse en aritmética entera sin presentar problemas de inestabilidad debido a errores de redondeo, ver [27].

Otro factor que determina el comportamiento numérico de las variables son las condi-

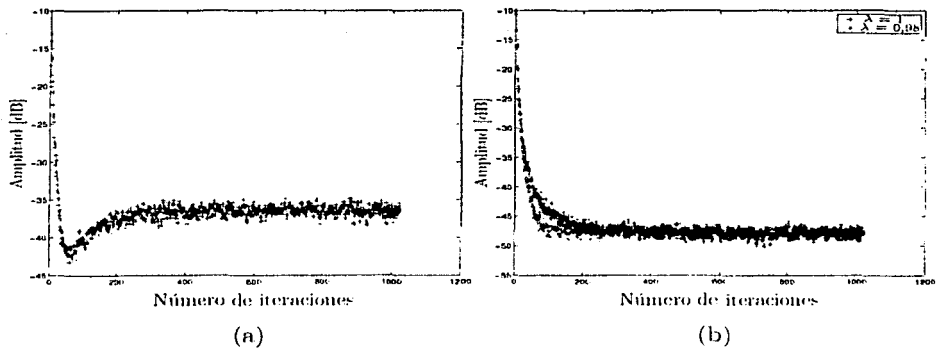


Figura 5.25: Comportamiento del error de filtrado en el algoritmo “Fast Kalman”. (a) aritmética de 16 bits ($\lambda = 0.98$). (b) aritmética de 32 bits

ciones iniciales, por esta razón siempre que se quiera implementar un algoritmo en aritmética finita es importante considerar las condiciones iniciales, puesto que la dinámica numérica de los datos nos determina el número máximo de bits a utilizar para la parte entera y por tanto, nos limitan los bits para parte fraccionaria. Conforme menos bits se tengan para parte fraccionaria, los errores de redondeo pueden crecer demasiado y ocasionar divergencia, sobre todo si se trabaja con algoritmos no estabilizados.

En la siguiente simulación se presenta el algoritmo “Fast Kalman” en aritmética entera de 16 y 32 bits para $\lambda = 0.98$.

En la figura 5.25 se muestra el comportamiento del error de filtrado del algoritmo “Fast Kalman” tanto en aritmética de 16 bits como de 32. Como era de esperarse, el algoritmo implementado en 16 bits converge a un error mayor que si se implementa en 32 bits. Si se utilizan 32 bits de precisión, es posible utilizar factores de olvido iguales a la unidad pues aunque la dinámica numérica aumenta utilizando 32 bits es posible mantener estable el algoritmo con respecto a errores de redondeo. Por otra parte podemos observar como para un factor de olvido menor a la unidad el algoritmo converge más rápido, ver [27].

Para observar el comportamiento de los errores de redondeo en la figura 5.26 se presentan los errores de redondeo sobre el error de filtrado. Esta simulación se real-

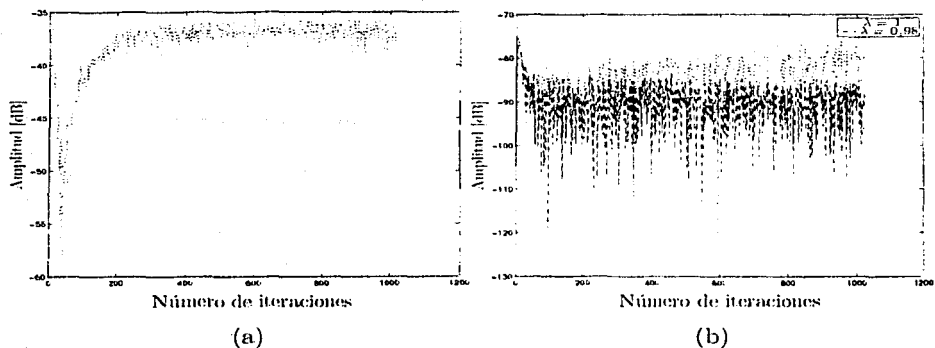


Figura 5.26: Comportamiento de los errores de redondeo en el error de filtrado $e(k)$ algoritmo Fast Kalman. (a) aritmética de 16 bits ($\lambda = 0.98$). (b) aritmética de 32 bits.

iza mediante la resta del error de filtrado considerado libre de errores de redondeo (double) menos tal variable en aritmética entera.

En la figura 5.26 (a) podemos observar como en la implementación de 16 bits los errores de redondeo aumentan drásticamente, sin embargo tienden a estabilizarse aunque definitivamente a un valor mucho mayor que para la implementación en 32 bits. Si comparamos las figuras 5.22 y 5.26 podemos observar que el comportamiento de los errores de redondeo son similares, sin embargo el algoritmo "Fast Kalman" solo presenta tales propiedades numéricas para ciertas condiciones sobre λ , además si se observa a detalle la figura 5.26 (b) el error de redondeo no se estabiliza lo que seguramente ocasionará divergencia para un número mayor de iteraciones. Sin embargo, si durante el periodo de igualación, el algoritmo es estable entonces se podrá utilizar tal implementación.

Finalmente, para observar que al algoritmo converge a su solución óptima en la figura 5.27 presentamos la respuesta en frecuencia del algoritmo "Fast Kalman" en aritmética entera de 16 y 32 bits.

En la figura 5.27 (a) podemos observar como la respuesta en frecuencia de la implementación en 16 bits no es exactamente la inversa del canal, esto es debido a que los errores de redondeo ocasionan que la respuesta del filtro se desvíe de la óptima, sin embargo los resultados observados son aceptables.

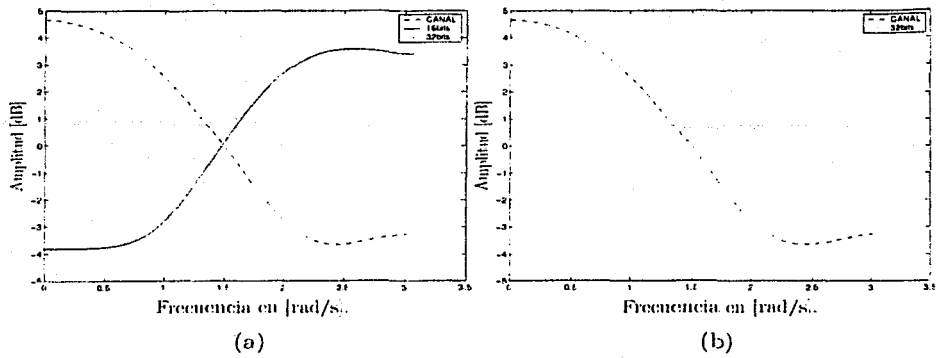


Figura 5.27: Respuesta en frecuencia de la implementación "Fast Kalman" en aritmética entera (a) $\lambda = 0.98$ (b) $\lambda = 1$

5.7. Conclusiones

En este capítulo hemos estudiado el comportamiento numérico de diferentes algoritmos de filtrado adaptable, tanto en aritmética de punto flotante como de punto fijo. En relación a los algoritmos RLS podemos decir que es importante la forma en la que se implementan las ecuaciones pues el comportamiento numérico de algunas implementaciones puede ocasionar divergencia.

Si los algoritmos RLS se implementan de manera adecuada como es el caso de la implementación CLS2, estos pueden presentar comportamientos bastante buenos conservando las buenas propiedades de los algoritmos basados en los mínimos cuadrados sin necesidad de hacer uso de implementaciones del tipo raíz cuadrada, como lo proponen [4] y [8], con el propósito de mejorar el comportamiento numérico de los algoritmos RLS.

El uso de los algoritmos rápidos se justifica dada la necesidad de disminuir el tiempo de procesamiento y aumentar la velocidad de transmisión, sin embargo, es importante asegurar un buen comportamiento numérico de los algoritmos rápidos.

Emplear algoritmos rápidos estabilizados no siempre garantiza que estos se pueden implementar en un dispositivo de procesamiento con longitud de palabra finita, por lo que es necesario simular los algoritmos en aritmética finita previo a implementarlos en un DSP, con el objetivo de evitar problemas de divergencia debido a la propagación de errores de redondeo.

El uso de factores de olvido menores a la unidad es indispensable en ambientes donde el canal varía rápidamente, sin embargo, muchos de los algoritmos presentan problemas numéricos cuando se usan λ 's < 1 . Por lo tanto, debemos considerar que λ juega un papel muy importante en el comportamiento de los algoritmos y que es necesario utilizar el valor de λ adecuado para cada aplicación, así como asegurar que el algoritmo utilizado es estable para el valor de λ escogido.

Tanto el SFRLSVC como el SFRLSVS se comportan bien si cumplimos con la condición de excitación persistente, sin embargo el algoritmo SFRLSVC presenta problemas de estabilidad si se trabaja con factores de olvido menores a 0.96. El buen desempeño de el algoritmo SFRLSVS, incluso implementado en 16 bits, se debe sin duda a la multiple retroalimentación sobre las variables sensibles, lo que permite estabilizar al algoritmo de una mejor manera.

Por último, con base en la simulación presentada en la última parte, podemos decir que si el algoritmo "Fast Kalman" se implementa de manera correcta este puede conservar sus buenas propiedades numéricas aún en aritmética de 16 bits sin necesidad de utilizar un algoritmo estabilizado que requiera del orden de 10^7 operaciones

aritméticas contra S_{μ} del "Fast Kalman".

Capítulo 6

CONCLUSIONES

Entendemos por eficiencia en un sistema de comunicaciones a la capacidad de enviar la mayor cantidad de información en el menor tiempo posible y con la menor probabilidad de error. Tal eficiencia se encuentra siempre limitada por el canal de transmisión cuyo ancho de banda es limitado, ocasionando que las señales se distorsionen. Para lograr que un sistema sea lo más eficiente posible es necesario utilizar dispositivos que compensen tales distorsiones del canal.

La distorsión introducida por el canal se debe fundamentalmente a la interferencia entre símbolos (ISI) ocasionada por la dispersión de los pulsos a lo largo del canal de transmisión. por esta razón en este trabajo hemos comenzado con el estudio de los diferentes tipos de canales, partiendo del par de cobre terminando con el canal inalámbrico. Con base en este estudio podemos decir que la distorsión se presenta en todos los tipos de canales aunque las causas no siempre son las mismas. Así, en el caso del par de cobre la distorsión se debe al ancho de banda insuficiente para transmitir los pulsos digitales, mientras que en el canal inalámbrico esta se debe a la propagación por multitraectorias.

Los dispositivos que compensan tales distorsiones del canal se conocen como igualadores y dado que a menudo los canales presentan características que varían con el tiempo es necesario utilizar igualadores adaptables.

Un igualador adaptable no es más que un filtro que puede tener diferentes tipos de estructuras dependiendo de la aplicación en particular y cuyo desempeño dependerá del algoritmo que se utilice para actualizar sus parámetros, así mismo la obtención y características de un algoritmo dependerá del criterio de optimización utilizado.

A lo largo de este trabajo hemos aprendido que la elección del criterio de optimización determina muchas de las características del algoritmo tales como estabilidad, velocidad de convergencia, tiempo de procesamiento y tipos de señales con las

que pueden trabajar, e.g. estacionarias.

Difícilmente encontraremos un algoritmo que cumpla con las mejores características, así: mientras algún algoritmo puede ser muy estable, también puede verse limitado por la velocidad de convergencia y por el tipo de señales que puede trabajar (e.g. algoritmos basados en el MSE), por otro lado, otros algoritmos pueden converger rápido pero presentar problemas de estabilidad.

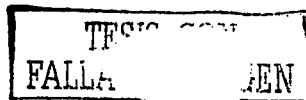
Por lo anterior podemos decir que el tipo de algoritmo usado dependerá de las necesidades que exija la aplicación y de los recursos disponibles.

Una característica no menos importante que las mencionadas anteriormente es la robustez numérica de los algoritmos. Como estudiamos y comprobamos en los capítulos 4 y 5 respectivamente, el comportamiento numérico de los algoritmos puede hacer que estos presenten problemas de divergencia. Es importante señalar que los problemas numéricos en los algoritmos RLS no solo se presentan cuando se trabaja con aritmética limitada, aunque cuando así sucede los problemas son más severos. Sin embargo, los problemas numéricos se presentan aún cuando se trabaja con precisión infinita de cálculo, por esta razón es importante trabajar con algoritmos estabilizados aún cuando el algoritmo se implemente en una longitud de palabra grande.

En particular, en la literatura se menciona que los algoritmos RLS presentan problemas numéricos que los hacen diverger, sobre todo cuando se trabaja con factores de olvido $\lambda < 1$. Sin embargo, como se mostró en el capítulo 5 estos problemas numéricos se pueden resolver si se utiliza la implementación adecuada para lo cual es necesario realizar un análisis de la propagación del error tal como se hizo en el capítulo 4.

Con respecto a los algoritmos FRLS podemos decir que éstos presentan problemas severos de divergencia debido a errores numéricos, pero sus características en cuanto a tiempo de procesamiento y velocidad de convergencia son deseables, por tal motivo, la utilización de algoritmos estabilizados se justifica siempre y cuando el algoritmo estabilizado solucione los problemas numéricos sin afectar las buenas características de los algoritmos rápidos.

Podemos decir que el problema de obtener algoritmos eficientes es similar al de obtener un sistema de comunicaciones eficiente. Esto es, en un sistema de comunicaciones, dado que la capacidad del canal es limitada, se emplea una codificación fuente para enviar la mayor cantidad de información con el mínimo número de bits, sin embargo, para realizar la corrección de errores se introducen bits de redundancia. En el caso de los algoritmos, dado que la capacidad de procesamiento de los dispositivos de procesamiento, e.g. DSP's, es limitada, se requiere de algoritmos que utilicen un menor número de operaciones aritméticas, tal es el caso de los algoritmos



FRLS, sin embargo: dado que estos presentan problemas numéricos, se introduce redundancia (algoritmos estabilizados) y en consecuencia un pequeño aumento en las operaciones aritméticas.

Además, es importante considerar que aún las implementaciones más robustas pueden presentar problemas de inestabilidad si nuestra matriz de covarianza no se mantiene bien condicionada a lo largo de todo el proceso y que la aparición de las matrices mal condicionadas puede deberse tanto a valores muy pequeños de λ como a la falta de excitación del filtro, lo que ocasiona matrices no singulares o no positivas.

Uno de los objetivos de esta tesis ha sido mostrar que en primer lugar, no existen soluciones únicas para cada problema pues siempre es importante hacer un balance de las características que más nos interesan que cumpla un algoritmo para la aplicación dada y segundo: que el algoritmo de filtrado es sólo una parte del sistema de comunicaciones y por tanto un buen desempeño del sistema en conjunto dependerá del buen funcionamiento del resto de los elementos del sistema.

En lo que respecta a la implantación de algoritmos, es importante considerar que estos serán implementados en dispositivos de procesamiento con un número finito de bits para representar las variables y que el costo de los dispositivos de procesamiento es proporcional a la capacidad de procesamiento del mismo. Por tal motivo, es importante encontrar algoritmos que puedan implementarse con un mínimo número de bits de precisión sin presentar problemas de divergencia debido a errores que provengan de la representación de los valores de las variables usadas en los algoritmos.

Apéndice A

LEMA DE INVERSIÓN MATRICIAL

A continuación se da un panorama general de una herramienta básica para la inversión de matrices. La utilidad primordial es reducir el número de operaciones para lograr dicha inversión.

Sea M una matriz particionada de la siguiente forma:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (\text{A.1})$$

Si queremos encontrar la inversa de la matriz definida en la ecuación A.1, podemos escribir:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (\text{A.2})$$

o de manera equivalente:

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (\text{A.3})$$

De la ecuación A.2 es posible obtener el sistema de ecuaciones siguiente:

$$A\alpha + B\gamma = I \quad (\text{A.4})$$

$$A\beta + B\delta = 0 \quad (\text{A.5})$$

$$C\alpha + D\gamma = 0 \quad (\text{A.6})$$

$$C\beta + D\delta = I \quad (\text{A.7})$$

De manera equivalente, de la ecuación A.3 podemos obtener el siguiente sistema:

$$\alpha A + \beta C = I \quad (\text{A.8})$$

$$\alpha B + \beta D = 0 \quad (\text{A.9})$$

$$\gamma A + \delta C = 0 \quad (\text{A.10})$$

$$\gamma B + \delta D = I \quad (\text{A.11})$$

A partir de las primeras cuatro ecuaciones, es posible obtener los elementos de la matriz inversa particionada:

De la ecuación A.6 tenemos:

$$\gamma = -D^{-1}C\alpha \quad (\text{A.12})$$

Sustituyendo la ecuación A.12 en la ecuación A.4 tenemos:

$$\alpha = (A - BD^{-1}C)^{-1} \quad (\text{A.13})$$

De la ecuación A.5:

$$\beta = -A^{-1}B\delta \quad (\text{A.14})$$

Sustituyendo la ecuación A.14 en la ecuación A.7 tenemos:

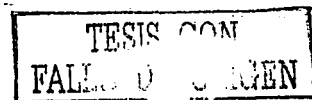
$$\delta = (D - CA^{-1}B)^{-1} \quad (\text{A.15})$$

Sustituyendo la ecuación A.13 en la ecuación A.12 tenemos:

$$\gamma = -D^{-1}C(A - BD^{-1}C)^{-1} \quad (\text{A.16})$$

Por último, sustituyendo la ecuación A.15 en la ecuación A.14 tenemos:

$$\beta = -A^{-1}B(D - CA^{-1}B)^{-1} \quad (\text{A.17})$$



Y podemos escribir:

$$M^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \quad (\text{A.18})$$

Utilizando el segundo sistema de ecuaciones dado por las ecuaciones A.8-A.11 podemos obtener otra forma para la inversa de la matriz:

De la ecuación A.8 tenemos:

$$\alpha = (I - \beta C)A^{-1} \quad (\text{A.19})$$

Sustituyendo la ecuación A.19 en la ecuación A.9 tenemos:

$$\beta = A^{-1}B(D - CA^{-1}B)^{-1} \quad (\text{A.20})$$

De la ecuación A.11 tenemos:

$$\delta = (I - \gamma B)D^{-1} \quad (\text{A.21})$$

Sustituyendo la ecuación A.21 en la ecuación A.10:

$$\gamma = -D^{-1}C(A - BD^{-1}C)^{-1} \quad (\text{A.22})$$

Sustituyendo la ecuación A.20 en la ecuación A.8 tenemos:

$$\alpha = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \quad (\text{A.23})$$

De igual forma, sustituyendo la ecuación A.22 en A.11, tenemos:

$$\delta = D^{-1} + D^{-1}C(A - BD^{-1}C)BD^{-1} \quad (\text{A.24})$$

Y finalmente podemos escribir:

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & A^{-1}B(D - CA^{-1}B)^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)BD^{-1} \end{bmatrix} \quad (\text{A.25})$$

También se puede demostrar que:

$$A^{-1}B(D - CA^{-1}B)^{-1} = (A - BD^{-1}C)^{-1}BD^{-1} \quad (\text{A.26})$$

$$D^{-1}C(A - BD^{-1}C)^{-1} = (D - CA^{-1}B)^{-1}CA^{-1} \quad (\text{A.27})$$

Si sustituimos la ecuación A.27 en la ecuación A.25 podemos escribir la ecuación A.25 como:

$$M^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A^{-1}B \\ I \end{bmatrix} [(D - CA^{-1}B)^{-1}] [-CA^{-1} - I] \quad (\text{A.28})$$

Si sustituimos la ecuación A.26 en la ecuación A.25 podemos escribir la ecuación A.25 como:

$$M^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & D^{-1} \end{bmatrix} + \begin{bmatrix} I \\ D^{-1}C \end{bmatrix} [(A - BD^{-1}C)^{-1}] [I - BD^{-1}] \quad (\text{A.29})$$

Las ecuaciones A.28 y A.29 se conocen como el lema de inversión para una matriz particionada y frecuentemente son utilizadas para la deducción de los algoritmos FRLS.

OBTENCIÓN DEL LEMA DE INVERSION MATRICIAL

Igualando la ecuación A.18 con la ecuación A.25 podemos obtener las dos versiones del lema de inversión matricial utilizado en la deducción de los algoritmos RLS:

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \quad (\text{A.30})$$

$$(D - CA^{-1}B)^{-1} = D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \quad (\text{A.31})$$



Apéndice B

SOLUCIÓN PARA LA PREDICCIÓN FORWARD

En este apéndice se presentará la solución para el filtro predictor de la figura 2.5. Primero se utilizará el criterio del error cuadrático promedio (MSE) y se llegará a la solución a través del algoritmo de Levinson-Durbin como un método que facilita la inversión de la matriz de correlación basado en su característica de matriz de Toeplitz para el caso de señales estacionarias. En segundo término se obtendrá la solución empleando el criterio de los mínimos cuadrados (LS) para llegar a la obtención del algoritmo RLS (LS recursivo) para la predicción.

Cabe señalar que el empleo de uno u otro criterio dependerá de la aplicación pues cada uno presenta sus ventajas. Por ejemplo, el criterio de los mínimos cuadrados no restringe las características de las señales a utilizar pero los algoritmos obtenidos son más complejos, por otra parte el criterio del error cuadrático promedio restringe las señales empleadas a ser estacionarias en sentido amplio pero los algoritmos a los que se llega son mucho más fáciles de implementar.

B.1. Minimización del error de predicción basado en el error cuadrático promedio

En el capítulo 2 introducimos la idea de procesos autoregresivos para generar señales estacionarias en sentido amplio. La idea consiste en encontrar un modelo para la generación de una señal estacionaria en sentido amplio que cumpla con la siguiente ecuación en diferencias:

$$\sum_{j=0}^p a_j^* u(k-j) = v(k) \quad (\text{B.1})$$

La ecuación B.1 se puede escribir como:

$$u(k) = - \sum_{j=1}^p a_j^* u(k-j) + v(k) \quad (\text{B.2})$$

La ecuación B.2 representa el modelo generador de una señal WSS y corresponde precisamente al problema de la predicción forward puesto que se desea obtener la señal $u(k)$ en término de sus valores pasados, si definimos un estimador de dicha señal estacionaria, salida de nuestro filtro predictor, como:

$$\hat{u}(k) = - \sum_{j=1}^p a_j^* u(k-j) \quad (\text{B.3})$$

podemos definir el error de predicción como la diferencia entre la señal deseada y la señal estimada:

$$e_p^f(k) = u(k) + \sum_{j=1}^p a_j^* u(k-j) \quad (\text{B.4})$$

El criterio del error cuadrático promedio consiste en minimizar la función costo definida por:

$$J_{MSE}(k) = E[(e_p^f(k))^2] \quad (\text{B.5})$$

Al minimizar la función costo con respecto a los parámetros del filtro predictor, a_0, a_1, \dots, a_p , podemos obtener la solución para el filtro predictor en forma matricial como:

$$R_p[k-1] A_p[k] = -r_p[k] \quad (\text{B.6})$$

donde

$A_p^T[k] = [a_1, a_2, \dots, a_p]$, es el vector de parámetros del filtro.

$R_p[k]$, es la matriz de correlación de la señal de entrada, definida como: $E[\mathcal{U}_p^*(k-1)\mathcal{U}_p(k-1)]$.

$r_p[k]$, es el vector de correlación entre la señal de entrada y la de salida, definido como: $E[\mathcal{U}_p^*(k-1)u(k)]$.

y.

$U_p(k-1)$ representa las muestras pasadas de la señal de entrada, es decir: $U_p(k-1) = [u(k-1), u(k-2), \dots, u(k-p)]$.

La ecuación B.6 define un sistema de ecuaciones conocidas como ecuaciones de Yule-Walker, [11].

Para resolver las ecuaciones de Yule-Walker, es necesario invertir la matriz de autocorrelación del proceso de entrada, $R_p(k-1)$, lo que implica del orden de p^3 multiplicaciones, lo cual en aplicaciones reales hace poco eficiente la solución por el tiempo y la capacidad de cálculo requeridos. Una manera más eficiente de resolver las *ecuaciones de Yule-Walker* es mediante el algoritmo de Levinson-Durbin el cual se fundamenta en el hecho de que la matriz de autocorrelación R_p tiene una estructura de Toeplitz, lo que significa que es simétrica y sus elementos diagonales y subdiagonales son los mismos. Tomando en cuenta estas consideraciones, el algoritmo de Levinson-Durbin, toma la ecuación de Yule-Walker al orden $p-1$, esto es:

$$R_{p-1}[k-1]A_{p-1}[k] = -r_{p-1}[k] \quad (\text{B.7})$$

Esto implica que si queremos encontrar los parámetros al orden p debemos encontrar los parámetros al orden $p-1$. Al igual que en un polinomio de grado n , donde sus raíces son diferentes a las de un polinomio de grado $n-1$, en este caso el vector de parámetros al orden p tiene diferentes elementos que el vector de parámetros al orden $p-1$. Tomando en consideración lo anterior se tiene:

$$\begin{bmatrix} R_p[k-1] & \overline{r_p[k]} \\ \overline{r_p[k]} & r_u(0) \end{bmatrix} \begin{bmatrix} A_p^{p-1}[k] \\ a_p^{p-1}[k] \end{bmatrix} = \begin{bmatrix} r_p[k] \\ r_u[p-1] \end{bmatrix} \quad (\text{B.8})$$

donde:

$$\overline{r_p[k]} = [r_u[p], \dots, r_u[1]]$$

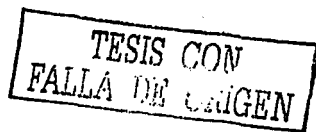
$$A_p^{p-1}[k] = [a_0, a_1, \dots, a_p] \text{ al orden } p+1,$$

$$a_p^{p-1} = a_{p-1} \text{ al orden } p-1.$$

A partir de la ecuación B.8 se puede obtener una recursividad para los parámetros del filtro de la siguiente forma, [27]:

$$A_p^{p-1}[k] = A_p^p[k] - k_p \overline{A_p^p[k]} \quad (\text{B.9})$$

La ecuación recursiva B.9 nos dice que podemos obtener de manera recursiva los parámetros de nuestro filtro predictor al orden $p-1$ a partir de los parámetros al orden p . El algoritmo de Levinson-Durbin se resume en el cuadro B.1.



| ALGORITMO DE LEVINSON |
|---|
| Condiciones iniciales |
| $a_m^m = 1$ |
| Predicción |
| $\hat{x}_{m+1} = \hat{x}_m - k_m \alpha_m$ |
| $\alpha_m = \sum_{j=0}^p a_j^m \rho_x (p-1-j)$ |
| $k_m = a_{m+1}^m = -\frac{\alpha_m}{\hat{x}_m}$ |
| $a_{m+1}^{m-1} = a_m^m + k_m a_{m-1}^m$ |

Cuadro B.1: Algoritmo de Levinson-Durbin

Una manera alterna para la solución del filtro predictor es el algoritmo de Leroux, el cual calcula los parámetros k_p , conocidos como "coeficientes de reflexión", del filtro sin necesidad de pasar por los parámetros a_p . El algoritmo de Leroux proporciona los parámetros k_p correspondientes a un filtro predictor con estructura lattice, el cual tiene mejores propiedades de estabilidad que un filtro transversal¹.

B.2. Minimización del error de predicción basado en el criterio de los mínimos cuadrados

Otra solución para el filtro predictor se puede obtener si consideramos otro criterio de optimización. Para el caso del criterio de los mínimos cuadrados la función a minimizar esta dada por la ecuación B.10.

A continuación se aplicará el criterio de los mínimos cuadrados para obtener la matriz de parámetros $A_p(k)$ del sistema FIR óptimo.

Consideraremos señales reales monodimensionales y el factor de olvido λ :

$$E_p(k) = \sum_{i=N}^M \lambda^{i-j} \varepsilon_p^i(j) \varepsilon_p^{i-1}(j) \quad (\text{B.10})$$

¹Para un estudio más completo de los filtros predictores lattice véase [28], [11].

donde el error a posteriori se define:

$$\varepsilon_p^T(j) = a(j) - A_p^T(k)U_p(j-1) \quad (\text{B.11})$$

sustituyendo para $N=1$ y suponiendo que el orden del filtro es p :

$$\begin{aligned} E_p(k) &= \sum_{j=1}^p \lambda^{k-j} \{ a(j) - A_p^T(k)U_p(j-1) \}^T \{ a^T(j) - U_p^T(j-1)A_p(k) \} \\ E_p(k) &= \sum_{j=1}^p \lambda^{k-j} \{ a(j)u^T(j) - A_p^T(k)U_p(j-1)u^T(j) \\ &\quad - a(j)U_p^T(j-1)A_p(k) - A_p^T(k)U_p(j-1)U_p^T(j-1)A_p(k) \} \end{aligned}$$

Definimos entonces:

$$\sum_{j=1}^p \lambda^{k-j} U_p(j-1)u^T(j) = r_p^f(k) \quad (\text{B.12})$$

$$\sum_{j=1}^p \lambda^{k-j} a(j)u^T(j) = r_{p0}^f(k) \quad (\text{B.13})$$

$$\sum_{j=1}^p \lambda^{k-j} U_p(j-1)U_p^T(j-1) = R_p^f(k) \quad (\text{B.14})$$

$$E_p(k) = r_{p0}^f(k) - A_p^T(k)r_p^f(k) - r_{p0}^{f^T}(k)A_p(k) - R_p^f(k)R_p^f(k)A_p(k)$$

Para obtener los parámetros óptimos, derivamos con respecto al vector de parámetros, A_p^T :

$$\frac{\partial E_p(k)}{\partial A_p} = r_p^f(k) - r_p^f(k) - \{ R_p^f(k) - R_p^{f^T}(k) \} A_p(k)$$

Como $R_p^f(k)$ es simétrica:

$$R_p^f(k) = R_p^{f^T}(k)$$

e igualando a cero,

$$\frac{\partial E_p(k)}{\partial A_p} = 0$$

Obtenemos entonces la solución óptima para la obtención de los parámetros $A_p(k)$:

$$R_p^f(k)A_p(k) = -r_p^f(k) \quad (\text{B.15})$$

¹Se sugiere ver apéndice de algebra vectorial en [27].



La ecuación B.15 parece no diferir con la ecuación B.6 obtenida con el criterio del error cuadrático promedio, sin embargo la diferencia radica en la manera en que se definen la matriz y el vector de correlación; además, el problema de la inversión de la matriz de autocorrelación también se presenta, pero en este caso gracias a la forma en que esta definida se puede obtener un cálculo recursivo de la inversa de esta matriz ayudándonos del lema de inversión matricial¹.

Primeramente, se puede obtener una forma recursiva para evaluar $R_p^f(k)$ y $r_p^f(k)^2$:

$$R_p^f(k) = \lambda R_p^f(k-1) - U_p(k-1)U_p^T(k-1) \quad (\text{B.16})$$

Similar para $r_p^f(k)$:

$$r_p^f(k) = \lambda r_p^f(k-1) + U_p(k-1)u^T(k) \quad (\text{B.17})$$

Es importante evaluar también de manera recursiva $A_p(k)$ por lo que a continuación se obtendrá a detalle esta recursividad. Sustituyendo la ecuación B.17 en la ecuación B.15:

$$R_p^f(k)A_p(k) = -\lambda r_p^f(k-1) - U_p(k-1)u^T(k) \quad (\text{B.18})$$

Y sustituyendo la ecuación B.15 en la ecuación B.18:

$$R_p^f(k)A_p(k) = \lambda R_p^f(k-1)A_p(k-1) - U_p(k-1)u^T(k) \quad (\text{B.19})$$

Si sumamos y restamos $U_p(k-1)U_p^T(k-1)A_p(k-1)$ en la ecuación B.19:

$$R_p^f(k)A_p(k) = \{\lambda R_p^f(k-1) - U_p(k-1)U_p^T(k-1)\}A_p(k-1) - U_p(k-1)\{u^T(k) + U_p^T(k-1)A_p(k-1)\} \quad (\text{B.20})$$

Si definimos el error a priori como³:

$$e_p^f(k) = u(k) - A_p^T(k-1)U_p(k-1) \quad (\text{B.21})$$

Entonces tenemos que:

$$R_p^f(k)A_p(k) = R_p^f(k)A_p(k-1) - U_p(k-1)u_p^{fT}(k)$$

$$R_p^f(k)[A_p(k) - A_p(k-1)] = -U_p(k-1)e_p^{fT}(k)$$

¹Vea apéndice A.

²En el capítulo 3 se muestra el proceso para obtener tales recursiones.

³Este error se define cuando utilizamos los parámetros al instante $k-1$.

y finalmente:

$$A_p(k) = A_p(k-1) - [R_p^f(k)]^{-1} U_p(k-1) e_p^{fT}(k) \quad (\text{B.22})$$

Donde:

$$K_p(k-1) = -[R_p^f(k)]^{-1} U_p(k) \quad (\text{B.23})$$

A la ecuación B.23 se le conoce como ganancia de Kalman. Podemos entonces escribir la ecuación B.22 en términos de la ganancia de Kalman como:

$$A_p(k) = A_p(k-1) + K_p(k-1) e_p^{fT}(k) \quad (\text{B.24})$$

Ahora, ocupamos el lema de inversión matricial para evaluar $[R_p^f(k)]^{-1}$. De la ecuación B.16 podemos escribir:

$$[R_p^f(k)]^{-1} = [\lambda R_p^f(k-1) + U_p(k-1) U_p^T(k-1)]^{-1}$$

Aplicando el lema de inversión matricial a la ecuación A.30, ver apéndice A:

$$[R_p^f(k)]^{-1} = \lambda^{-1} R_p^{-1}(k-1) + \lambda^{-1} R_p^{-1} U_p(k) [1 - U_p^T(k) \lambda^{-1} R_p^{-1}(k-1) U_p(k)]^{-1} U_p^T(k) \lambda^{-1} R_p^{-1}(k-1)$$

Por último:

$$[R_p^f(k)]^{-1} = \lambda^{-1} [R_p^f(k-1)]^{-1} + \frac{\lambda^{-1} [R_p^f(k-1)]^{-1} U_p(k-1) U_p^T(k-1) [R_p^f(k-1)]^{-1}}{\lambda - U_p^T(k-1) [R_p^f(k-1)]^{-1} U_p(k-1)} \quad (\text{B.25})$$

Con esta última expresión solo necesitamos calcular la inversa de $R_p^f(k)$ una sola vez y después evaluar $[R_p^f(k)]^{-1}$ en forma recursiva. El algoritmo para predicción basado en el criterio de mínimos cuadrados se muestra en el cuadro B.2.

ALGORITMO PARA PREDICCIÓN

Condiciones iniciales

$$R^{-1}[0] = \frac{1}{\lambda} I_p$$

$$A_p[0] = 0$$

$$U_p[0] = 0$$

Predicción

$$\hat{c}_p[k] = a[k] - A_p[k-1]U_p[k-1]$$

$$R^{-1}[k] = \frac{1}{\lambda} R^{-1}[k-1] - \frac{1}{\lambda} \frac{R^{-1}[k-1]U_p[k-1]U_p[k-1]^T R^{-1}[k-1]}{\lambda - U_p[k-1]^T R^{-1}[k-1]U_p[k-1]}$$

$$A[k] = A[k-1] + R^{-1}[k]U_p[k]e(k)$$

Cuadro B.2: Algoritmo predictor RLS

TESIS CON
FALLA DE ORIGEN

Apéndice C

ANÁLISIS DE VALORES Y VECTORES CARACTERÍSTICOS.

El comportamiento de las matrices de correlación en los algoritmos de filtrado determina en muchos casos el buen comportamiento del algoritmo. Es posible analizar el comportamiento de tal matriz por medio de una descomposición en valores y vectores característicos que como veremos nos proporcionan mucha información sobre la matriz de correlación.

Las propiedades presentadas en este apéndice se basan en [13].

Dejemos que la matriz hermitiana R denote la matriz de correlación de dimensión $M \times M$ de un proceso estocástico discreto estacionario en sentido amplio. Dicho proceso se representa por un vector de observaciones $u(k)$ de dimensión $M \times 1$. En general esta matriz puede contener valores complejos en sus elementos. El problema consiste en encontrar el vector q de dimensión $M \times 1$ que satisfaga la condición dada por la ecuación C.1, para algún valor de λ constante.

$$Rq = \lambda q \quad (C.1)$$

Esta condición establece que el vector q es transformado linealmente al vector λq por la matriz hermitiana R . Dado que λ es una constante, el vector q por lo tanto es invariante en dirección izquierda (en el espacio M -dimensional) por una transformación lineal.

Para una matriz típica R de $M \times M$, existirán M de tales vectores. Para mostrarlo, reescribimos la ecuación C.1:

$$(R - \lambda I)q = \bar{0}$$

donde I es una matriz identidad de $M \times M$ y $\bar{0}$ es el vector nulo de $M \times 1$. La matriz $(R - \lambda I)$ debe ser singular¹, así la última ecuación tiene una solución diferente de cero en el vector q si y solo si el determinante de la matriz $(R - \lambda I)$ es igual a cero:

$$\det(R - \lambda I) = 0$$

Al realizar este determinante se obtiene un polinomio en λ de grado M . De esta manera, en general la última ecuación tiene M distintas raíces y de manera correspondiente tiene M soluciones en el vector q . Esta ecuación es llamada ecuación característica de la matriz R . $\lambda_1, \lambda_2, \dots, \lambda_M$ denotan las M raíces de la ecuación y se denominan valores característicos de la matriz R .

Aunque la matriz R tiene M valores característicos, no necesariamente son diferentes, cuando esto sucede se dice que la matriz R tiene valores característicos degenerados.

Si λ_i denota el i -ésimo valor característico de la matriz R y q_i es un vector diferente de cero tal que:

$$Rq_i = \lambda_i q_i$$

Entonces, el vector q_i se denomina vector característico asociado con λ_i ; así, un vector característico corresponde solamente a un valor característico. Sin embargo, un valor característico puede tener más de un vector característico, e.g. Si q_i es un vector característico asociado con un valor característico λ_i , entonces también lo es aq_i para cualquier $a \neq 0$.

C.0.1. Propiedades de los valores característicos y vectores característicos

Esta propiedades se presentan considerando una matriz de correlación para procesos estacionarios.

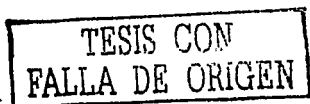
Propiedad 1. Si $\lambda_1, \lambda_2, \dots, \lambda_M$ denotan los valor característicos de la matriz de correlación R , entonces los valor característicos de la matriz R^k son iguales a $\lambda_1^k, \lambda_2^k, \dots, \lambda_M^k$ para cualquier $k > 0 \in \mathbf{Z}$.

$$R^k q = \lambda^k q \tag{C.2}$$

Entonces:

- 1) Si λ es un valor característico de R entonces λ^k es un valor característico de R^k .
- 2) Cualquier vector característico de R también es un vector característico de R^k .

¹Sus renglones son linealmente dependientes y no tiene inversa.



Propiedad 2. Si q_1, q_2, \dots, q_M son los vectores característicos correspondientes a los valores característicos $\lambda_1, \lambda_2, \dots, \lambda_M$ de la matriz R de $M \times M$, entonces los vectores característicos q_1, q_2, \dots, q_M son linealmente independientes.

$$\sum_{i=1}^M v_i q_i = 0 \quad (C.3)$$

Propiedad 3. Si $\lambda_1, \lambda_2, \dots, \lambda_M$ son los valores característicos de la matriz de correlación R de $M \times M$, entonces todos estos valores característicos son reales y no negativos.

Se puede expresar a un valor característico determinado como:

$$\lambda_i = \frac{q_i^H R q_i}{q_i^H q_i} \quad \forall i = 1, 2, \dots, M \quad (C.4)$$

Dado que la matriz de autocorrelación R siempre es definida no negativa, la forma hermitiana $q_i^H R q_i$ en el numerador siempre es real y no negativa, por lo tanto:

$$\lambda_i \geq 0; \quad \forall i \quad (C.5)$$

A la relación dada en la ecuación C.4 se le conoce como cociente de Rayleigh del vector q_i . Por tanto, podemos decir que un valor característico de la matriz de correlación es igual al cociente de Rayleigh de su correspondiente vector característico.

Propiedad 4. Si q_1, q_2, \dots, q_M son los vectores característicos correspondientes a los valores característicos $\lambda_1, \lambda_2, \dots, \lambda_M$ de la matriz de correlación R de $M \times M$, entonces los vectores q_1, q_2, \dots, q_M son ortogonales entre sí.

$$q_i^H q_j = 0; \quad \forall i \neq j \quad (C.6)$$

Propiedad 5. Transformación de similaridad unitaria.

Si q_1, q_2, \dots, q_M son vectores característicos correspondientes a los valores característicos $\lambda_1, \lambda_2, \dots, \lambda_M$ de la matriz de correlación R de $M \times M$, definida por:

$$Q = [q_1, q_2, \dots, q_M]$$

con q_1, q_2, \dots, q_M ortonormales. Definimos la matriz diagonal de $M \times M$:

$$\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_M\}$$

Entonces la matriz R puede ser diagonalizada de la siguiente forma:

$$Q^H R Q = \Lambda$$

Podemos escribir la ecuación C.1 en forma matricial:

$$R Q = Q \Lambda \quad (C.7)$$

Y dado que los vectores característicos son ortonormales:

$$Q^H Q = I \Rightarrow Q^{-1} = Q^H$$

Una matriz con esta propiedad se llama matriz unitaria. Entonces de la ecuación C.7:

$$Q^H R Q = \Lambda \quad (C.8)$$

A la última transformación se le conoce como transformación de similaridad unitaria y quiere decir que la matriz de autocorrelación puede ser diagonalizada por una transformación de similaridad unitaria, donde la matriz Q que se usa para diagonalizar R tiene como columnas a un grupo ortonormal de vectores característicos de R y la matriz diagonal resultante Λ tiene como elementos en su diagonal los valores característicos de R .

Propiedad 6. Si $\lambda_1, \lambda_2, \dots, \lambda_M$ son los valores característicos de la matriz R de $M \times M$. Entonces la suma de estos valores característicos es igual a la traza de la matriz ¹ R .

$$\text{tr}\{R\} = \sum_{i=1}^M \lambda_i \quad (C.9)$$

Propiedad 7. La matriz de correlación R es mal condicionada si la razón del mayor valor característico de R con respecto al menor es grande.

La importancia de esta propiedad se basa en el hecho de que el desarrollo de un algoritmo para la solución efectiva de algún problema de procesamiento de alguna señal y la comprensión de la teoría de perturbación asociada van de la mano.

Considerando el siguiente sistema de ecuaciones, que puede representar a las ecuaciones normales presentadas en los algoritmos:

$$Aw = d$$

donde,

A es la matriz de coeficientes,

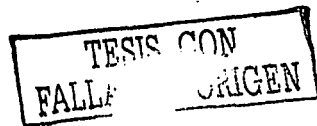
w son las incógnitas, y

d son los términos independientes.

La matriz A y el vector d son cantidades que dependen de los datos de nuestra señal y w es un vector de coeficientes caracterizando un filtro lineal de interés.

Una formulación elemental de la teoría de perturbación nos dice que si la matriz A y el vector d son perturbados por pequeñas cantidades δA y δd , respectivamente y

¹La traza de una matriz cuadrada se define como la suma de los elementos de la diagonal.



si $\frac{\partial A}{\partial a_i}$ y $\frac{\partial \delta a}{\partial a_i}$ son del orden de ϵ con $\epsilon \ll 1$, tenemos:

$$\frac{\|\delta a\|}{\|a\|} \leq \epsilon \chi(A)$$

Donde δa es el cambio producido en a y $\chi(A)$ es el número condición¹ de la matriz A con respecto a la inversión. El número condición es llamado así porque describe la mala condición ó mal comportamiento de la matriz A cuantitativamente y se define:

$$\chi(A) = \|A\| \|A^{-1}\|$$

Donde $\|\cdot\|$ simboliza una norma². La normas satisfacen las siguientes condiciones:

- 1) $\|A\| \geq 0$, $\|A\| = 0$ si y solo si $A = 0$.
- 2) $\|CA\| = |C| \|A\|$, donde C es cualquier número real y $|C|$ su magnitud.
- 3) $\|A + B\| \leq \|A\| + \|B\|$ Desigualdad del triángulo.
- 4) $\|AB\| \leq \|A\| \|B\|$ Consistencia mutua.

Utilizando la definición de norma espectral definida como la raíz cuadrada del mayor eigenvalor de la matriz producto $A^H A$, donde A^H es la hermitiana transpuesta de A :

$$\|A\|_s = \sqrt{\text{mayor eigenvalor de } A^H A}$$

Expresando el máximo valor característico como el cociente de Rayleigh del correspondiente vector característico y tomando el cuadrado de la última expresión:

$$\|A\|_s^2 = \max \frac{x^H A^H A x}{x^H x} = \max \frac{\|Ax\|^2}{\|x\|^2}$$

Donde $\|x\|^2$ y $\|Ax\|^2$ son productos internos del vector x y de la matriz Ax respectivamente. Por tanto $\|x\|$ es la norma euclídiana de x . Y la norma espectral de A esta dada por:

$$\|A\|_s = \max \frac{\|Ax\|}{\|x\|}$$

Entonces, utilizando esta definición de norma espectral para la norma de la matriz R , puesto que R es hermitiana se tiene:

$$R^H = R$$

Utilizando la propiedad 1, el máximo valor característico de $R^H R$ es igual a λ^2 y tenemos la norma espectral de R :

$$\|R\|_s = \lambda_{\max}$$

¹Ver apéndice E.

²La norma de una matriz es un número que se le asigna a ésta y que es en cierto sentido una medida de la magnitud de la matriz. Ver apéndice E.

De manera similar se puede demostrar que la norma espectral de R^{-1} es [13]:

$$\|R^{-1}\|_2 = \frac{1}{\lambda_{\min}}$$

Y el número condición de la matriz de correlación:

$$\chi(R) = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (\text{C.10})$$

A la ecuación C.10 se le conoce como valor característico disperso ó razón de valor característico de la matriz de correlación y siempre es ≥ 1 .

El efecto de una matriz de correlación mal condicionada lo podemos apreciar suponiendo una matriz \mathbf{R} normalizada, $r(0) = 1$. Entonces, si su valor característico disperso es muy grande se tiene que R^{-1} tiene valores muy grandes y esto puede ocasionar problemas al resolver un sistema de ecuaciones involucrando R^{-1} .

Propiedad 8. Los valores característicos de la matriz de correlación de un procesos estocástico o señal aleatoria discreta estan limitados por el valor mínimo y máximo de la densidad espectral de potencia del proceso. Se puede demostrar que los valores característicos estan limitados por los valores máximos y mínimos de $S(\omega)$ [13]:

$$S_{\min}(\omega) \leq \lambda_i \leq S_{\max}(\omega) \quad \forall i = 1, \dots, M$$

En consecuencia:

$$\chi(R) = \frac{\lambda_{\max}}{\lambda_{\min}} \leq \frac{S_{\max}}{S_{\min}} \quad (\text{C.11})$$

Es interesante notar que conforme la dimensión de la matriz de correlación se aproxima al infinito, los máximo λ_{\max} y mínimo λ_{\min} valores característicos se aproximan a S_{\max} y S_{\min} respectivamente.

Propiedad 9. Teorema mínimax.

Sea \mathbf{R} la matriz de correlación y $\lambda_1, \lambda_2, \dots, \lambda_M$ los correspondientes valores característicos ordenados en forma decreciente:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$$

Este teorema establece:

$$\lambda_k = \min_{\dim(\Phi)=k} \max_{\Phi, x \neq 0} \frac{x^H R x}{x^H x}; \quad k = 1, 2, \dots, M \quad (\text{C.12})$$

Donde:

Ψ es un subespacio del espacio de vectores complejos de $M \times 1$.

$\dim(\Psi)$ es la dimensión del subespacio Ψ .

$x \in \Psi$ es el vector $x \neq 0$ que varía sobre el espacio Ψ .

Si tenemos un espacio vectorial de dimensión M con vectores base q_1, q_2, \dots, q_M y definimos $x = \sum_{i=1}^k a_i q_i$ a un vector de dimensión M en el subespacio de dimensión k con vectores base q_1, q_2, \dots, q_k . Entonces obviamente $k \leq M$ pero la dimensión del vector x es M .

Postmultiplicando ambos lados de la ecuación C.7 por Q^{-1} y sabiendo que Q es una matriz unitaria, podemos expresar la matriz de correlación como:

$$R = Q \Lambda Q^H = \sum_{i=1}^M \lambda_i q_i q_i^H \quad (C.13)$$

La ecuación C.14 se conoce como teorema espectral. Expresando el cociente de Rayleigh del vector x :

$$\frac{x^H R x}{x^H x} = \frac{\sum_{i=1}^k a_i^2 \lambda_i}{\sum_{i=1}^k a_i^2} \quad (C.14)$$

La ecuación C.14 establece que el cociente de Rayleigh de un vector x ubicado en el subespacio Ψ de dimensión k representado por los vectores característicos base q_1, q_2, \dots, q_k es una media pesada de los valores característicos $\lambda_1, \lambda_2, \dots, \lambda_k$ y, puesto que hemos asumido $\lambda_1 \leq \lambda_2 \leq \dots, \lambda_k$. Para cualquier subespacio de dimensión k :

$$\max_{x \in \Psi, x \neq 0} \frac{x^H R x}{x^H x} \leq \lambda_k$$

Y por lo tanto:

$$\min_{\dim(\Psi)=k} \max_{x \in \Psi, x \neq 0} \frac{x^H R x}{x^H x} \leq \lambda_k \quad (C.15)$$

También se puede demostrar que para cualquier subespacio Ψ de dimensión k , representado por los vectores característicos $q_{i_1}, q_{i_2}, \dots, q_{i_k}$, donde i_1, i_2, \dots, i_k es un subconjunto de $1, 2, \dots, M$, existe al menos un vector $x \neq 0$ común a Ψ y al subespacio Ψ' representado por los vectores característicos $q_k, q_{k+1}, q_{k+2}, \dots, q_M$ y cuyo cociente de Rayleigh se puede expresar como la media pesada de los valores característicos $\lambda_k, \lambda_{k+1}, \dots, \lambda_M$. [3]. Esto es:

$$\frac{x^H R x}{x^H x} = \frac{\sum_{i=k}^M b_i^2 \lambda_i}{\sum_{i=k}^M b_i^2} \quad (C.16)$$

Puesto que hemos asumido $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_M$. Para cualquier subespacio de dimensión k :

$$\max_{x, \Psi, x \neq 0} \frac{x^H R x}{x^H x} \geq \lambda_k$$

Y por lo tanto:

$$\min_{x, \Psi, \dim \Psi = k} \max_{x, \Psi, x \neq 0} \frac{x^H R x}{x^H x} \geq \lambda_k \quad (\text{C.17})$$

Tomando las ecuaciones C.15 y C.17 se demuestra el teorema de minimax considerando que:

- 1) No se requiere algún conocimiento especial de la eigenestructura de \mathbf{R} que sólo debe ser adoptada para definir los valores característicos.
- 2) Señala una doble característica de la eigenestructura de \mathbf{R} . *a)* Los vectores característicos representan la base particular para un espacio M-dimensional más eficiente en el sentido energético. *b)* Los valores característicos son ciertas energías de un vector de entrada de $M \times 1$, $U(n)$, como veremos en la siguiente propiedad.

Propiedad 10. Expansión de Karhunen-Loève.

Si $U(n)$ de dimensión $M \times 1$ denota una secuencia de datos derivados de un proceso WSS de media cero y matriz de correlación \mathbf{R} y q_1, q_2, \dots, q_M son los vectores característicos asociados a los M valores característicos de \mathbf{R} , entonces el vector $U(n)$ puede ser expandido como una combinación lineal de estos eigenvectores:

$$U(n) = \sum_{i=1}^M c_i(n) q_i \quad (\text{C.18})$$

Donde los coeficientes $c_i(n)$ son variables aleatorias no correlacionadas de media cero definidos por el producto interno:

$$c_i(n) = \sum_{l=1}^M q_l^H U(n) \quad (\text{C.19})$$

Además:

$$E[c_i(n) c_j^*(n)] = \begin{cases} \lambda_i; & i = j \\ 0; & i \neq j \end{cases}$$

La ecuación anterior confirma que los coeficientes son no correlacionados y que cada uno de ellos tiene un valor medio cuadrático igual a su respectivo valor característico.

Obteniendo la norma al cuadrado de la ecuación C.18:

$$\|U(u)\|^2 = \sum_{i=1}^M c_i(u)^2$$

Lo cual nos indica que el coeficiente $c_i(u)$ tiene una energía igual a la de la observación medida del vector uu a lo largo de la i -ésima coordenada. Naturalmente, esta energía es una variable aleatoria cuyo valor medio es igual al i -ésimo valor característico, $E[c_i(u)^2] = \lambda_i, i = 1, 2, \dots, M$.

Apéndice D

ALGORITMOS DE FILTRADO FRLS

D.1. Algoritmo FRLS Versión Cioffi



ALGORITMO FRLS VERSIÓN CIOFFI [21]

Condiciones Iniciales

$$\alpha_p^i(k-1) = \mu \lambda^i; \quad \alpha_p^h(k-1) = \mu$$

$$\tau_p(k-1) = 1; \quad A_p(k-1) = B_p(k-1) = H_p(k-1) = C_p(k-1) = U_p(k-1) = 0$$

Actualización de la ganancia dual de Kalman $C_p(k)$

$$e_p^i(k) = u(k) - A_p^T(k-1)U_p(k-1)$$

$$\varepsilon_p^i(k) = e_p^i(k)\tau_p(k-1)$$

$$\alpha_p^i(k) = \lambda \alpha_p^i(k-1) + \varepsilon_p^i(k) e_p^{iT}(k)$$

$$\tau_{p-1}(k) = \lambda \tau_p(k-1) \alpha_p^T(k-1) \alpha_p^T(k)$$

$$C_{p-1}(k) = \begin{bmatrix} 0 \\ C_p(k-1) \end{bmatrix} - \begin{bmatrix} 1 \\ -A_p(k-1) \end{bmatrix} \lambda^{-1} \alpha_p^T(k-1) e_p^i(k) = \begin{bmatrix} D_p(k) \\ \mu(k) \end{bmatrix}$$

$$A_p(k) = A_p(k-1) - C_p(k-1) \varepsilon_p^{iT}(k)$$

$$e_p^h(k) = -\lambda \alpha_p^h(k-1) \mu(k)$$

$$\tau_p(k) = \frac{\tau_{p-1}(k)}{1 + \tau_{p-1}(k) C_p^T(k) C_p(k)}$$

$$\varepsilon_p^h(k) = e_p^h(k) \tau_p(k)$$

$$\alpha_p^h(k) = \lambda \alpha_p^h(k-1) + \varepsilon_p^h(k) e_p^{hT}(k)$$

$$C_p(k) = D_p(k) - B_p(k-1) \mu(k)$$

$$B_p(k) = B_p(k-1) + C_p(k) \varepsilon_p^{hT}(k)$$

Actualización del filtro

$$e_p(k) = d(k) - W_p^T(k-1)U_p(k)$$

$$\varepsilon_p(k) = e_p(k) \tau_p(k)$$

$$W_p(k) = W_p(k-1) + C_p(k) \varepsilon_p^T(k)$$

147

Cuadro D.1: Algoritmo FRLS Version Cioffi

D.2. Algoritmo FRLS Versión Slock

ALGORITMO FRLS VERSIÓN SLOCK [15]

Condiciones Iniciales

$$\alpha_p^{-1}(k-1) = [\mu\lambda^{\mu}]^{-1}; \quad \alpha_p^0(k-1) = \mu$$

$$\gamma_p(k-1) = 1; \quad A_p(k-1) = B_p(k-1) = W_p(k-1) = C_p(k-1) = U_p(k-1) = 0$$

Actualización en orden "Update"

$$e_p^f(k) = w(k) - A_p^T(k-1)U_p(k-1)$$

$$C_{p+1}^1(k) = -\lambda^{-1}\alpha_p^{-1}(k-1)e_p^f(k)$$

$$C_{p+1}(k) = \begin{bmatrix} 0 \\ C_p(k-1) \end{bmatrix} - \begin{bmatrix} 1 \\ A_p(k-1) \end{bmatrix} C_{p+1}^1(k) = \begin{bmatrix} M_p(k) \\ \mu(k) \end{bmatrix}$$

$$\gamma_{p+1}^1(k) = \gamma_p^{-1}(k-1) - C_{p+1}^1(k)e_p^f(k)$$

Predicción Forward

$$\varepsilon_p^f(k) = e_p^f(k) - \gamma_p(k-1)$$

$$A_p(k) = A_p(k-1) - C_p(k-1)\varepsilon_p^f(k)$$

$$\alpha_p^{-1} = \lambda^{-1}\alpha_p^{-1}(k-1) - [C_{p+1}^1(k)]^T \gamma_{p+1}^1(k) C_{p+1}^1(k)$$

Actualización en orden "Downdate"

$$e_p(k) = -\lambda\alpha_p^0(k-1)C_{p+1}^{-1}(k)$$

$$[C_{p+1}^{-1}(k)] = C_p^0(k-1) - C_{p+1}^1(k)A_p^T(k-1)$$

$$C_p(k) = M_p(k) - C_{p+1}^{-1}(k)B_p(k-1)$$

$$\gamma_p^{-1}(k) = \gamma_{p+1}^{-1}(k) - [C_{p+1}^{-1}(k)]e_p(k)e_p^T(k)$$

Cuadro D.2: Algoritmo FRLS Versión Slock

| ALGORITMO FRLS VERSIÓN SLOCK (CONTINUACIÓN) |
|---|
| Predicción Backward |
| $\varepsilon_p^b(k) = \hat{e}_p^b(k) \gamma_p(k)$ $B_p(k) = B_p(k-1) + \varepsilon_p^b(k) C_p(k)$ $\alpha_p^b(k) = \lambda^{-1} \alpha_p^b(k-1) + \varepsilon_p^b(k) \varepsilon_p^{bT}(k)$ |
| Filtrado |
| $e_p(k) = d(k) + W_p^T(k-1) U_p(k)$ $\varepsilon_p(k) = e_p(k) \gamma_p(k)$ $W_p(k) = W_p(k-1) + C_p(k) \varepsilon_p^T(k)$ |

Cuadro D.3: Continuación algoritmo FRLS Versión Slock

Apéndice E

ALGUNOS ASPECTOS DE ALGEBRA VECTORIAL

E.1. Rango de una matriz

Dejemos que A sea una matriz de $m \times n$. El tamaño r ($1 \leq r \leq \min(m, n)$) de la matriz identidad en la forma reducida (canónica) de A se refiere al rango, escrito como $r = \text{rango}(A)$. [18].

Una submatriz de una matriz A dada, se define como A misma ó cualquier arreglo resultante después de que ciertas líneas se suprimen de A . Por ej. la matriz:

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix}$$

Tiene como submatrices, primero a si misma y después a las matrices:

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \begin{bmatrix} x_{11} & x_{13} \\ x_{21} & x_{23} \end{bmatrix} \begin{bmatrix} x_{12} & x_{13} \\ x_{22} & x_{23} \end{bmatrix} \\ \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \end{bmatrix} [x_{11} \ x_{12} \ x_{13}] [x_{21} \ x_{22} \ x_{23}] \dots$$

y finalmente sus elementos individuales. Las submatrices cuadradas de una matriz dada son de utilidad particular. El determinante de una submatriz de orden r de una matriz dada es a menudo llamado determinante de orden r de la matriz.

Se dice que una matriz es de rango r si y solo si tiene por lo menos una submatriz no singular de orden r , pero no tiene submatrices no singulares de orden mayor a r . Se dice que una matriz es de rango cero sí y solo sí todos sus elementos son cero.

E.2. Normas de matrices y números condición

La exactitud de las soluciones calculadas en los sistemas lineales puede ser probada y depende de que tan sensible sea la matriz de coeficientes del sistema a pequeñas variaciones. La sensibilidad de una matriz puede ser medida en términos de su número condición. El número condición de una matriz no singular se define en términos de su norma y de la norma de su inversa. Antes de discutir sobre los números condición es necesario establecer algunos aspectos de las normas estándares de matrices.

E.2.1. Normas de matrices

Tal como las normas de vectores son usadas para medir el tamaño de los vectores, las normas de las matrices pueden ser usadas para medir el "tamaño" de las matrices.

La norma de Frobenius sobre $\mathbb{R}^{m \times n}$ se introduce por el producto interno sobre $\mathbb{R}^{m \times n}$ y se denota $\| \cdot \|_F$. Esta norma puede calcularse tomando la raíz cuadrada de la suma del cuadrado de sus elementos:

$$\| A \|_F = \left[\sum_{j=1}^n \sum_{i=1}^m a_{ij}^2 \right]^{1/2}$$

La ecuación anterior define una familia de normas de matrices dado que define una norma sobre $\mathbb{R}^{m \times n}$ para cualesquiera m y n . Esta norma tiene importantes propiedades listadas a continuación:

1. Si a_j representa el vector columna j de A , entonces:

$$\| A \|_F = \left(\sum_{j=1}^n \sum_{i=1}^m a_{ij}^2 \right)^{1/2} = \left[\sum_{j=1}^n \| a_j \|_2^2 \right]^{1/2}$$

2. Si $a(i, :)$ representa el vector renglón i de A , entonces:

$$\| A \|_F = \left[\sum_{j=1}^m \sum_{i=1}^n a_{ij}^2 \right]^{1/2} = \left[\sum_{i=1}^m \| a(i, :)^T \|_2^2 \right]^{1/2}$$

donde:

$$\| x \|_2 = \sqrt{x^T x}$$

3. Si $x \in \mathbb{R}^n$, entonces:

$$\begin{aligned} \|Ax\|_2 &= \left[\sum_{i=1}^m \left(\sum_{j=1}^n a_{ij}x_j \right)^2 \right]^{\frac{1}{2}} = \left[\sum_{i=1}^m (a(i,:)x)^2 \right]^{\frac{1}{2}} \\ &\leq \left[\sum_{i=1}^m \|x\|_2^2 \|a(i,:)^\top\|_2^2 \right]^{\frac{1}{2}} \\ &= \|A\|_F \|x\|_2 \end{aligned}$$

4. Si $B = (b_1, b_2, \dots, b_r)$ es una matriz de $n \times r$, de la propiedad 1 y 3:

$$\begin{aligned} \|AB\|_F &= \|(Ab_1, Ab_2, Ab_3, \dots, Ab_r)\|_F \\ &= \left[\sum_{i=1}^r \|Ab_i\|_2^2 \right]^{\frac{1}{2}} \\ &\leq \|A\|_F \left[\sum_{i=1}^r \|b_i\|_2^2 \right]^{\frac{1}{2}} \\ &= \|A\|_F \|B\|_F \end{aligned}$$

Existen muchas otras normas que podemos usar para $\mathfrak{R}^{m \times n}$ además de la norma de Frobenius. Cualquier norma usada debe satisfacer las siguientes 3 condiciones que definen las normas en general:

1. $\|A\| \geq 0$ y $\|A\| = 0$ si y solo si $A = 0$.
2. $\|\alpha A\| = |\alpha| \|A\|$
3. $\|A + B\| \leq \|A\| + \|B\|$

Las familias de normas más utilizadas además satisfacen la propiedad adicional:

$$4. \|AB\| \leq \|A\| \|B\|$$

En consecuencia, solo consideraremos familias de normas que tienen esta propiedad adicional. Una importante consecuencia de esta propiedad:

$$\|A^n\| \leq \|A\|^n$$

En particular, si $\|A\| \leq 1$, entonces $\|A^n\| \rightarrow 0$ conforme $n \rightarrow \infty$.

NORMAS COMPATIBLES Y SUBORDINADAS

En general, la norma de una matriz $\| \cdot \|_M$ sobre $\mathbb{R}^{m \times n}$ y la norma de un vector $\| \cdot \|_r$ sobre \mathbb{R}^n se dice que son compatibles si:

$$\| Ax \| \leq \| A \|_M \| x \|_r, \forall x \in \mathbb{R}^n$$

En particular, de la propiedad 3 podemos decir que $\| \cdot \|_F$ y la norma vector $\| \cdot \|_2$ son compatibles. Para cada norma estándar de vector podemos definir una norma de matriz compatible usando la norma vector para calcular un operador norma para la matriz. Una norma de matriz definida de esta manera se dice que es subordinada de la norma vector.

Si la familia de las normas de matrices $\| \cdot \|_M$ es subordinada de la familia de normas de vectores $\| \cdot \|_r$, entonces $\| \cdot \|_M$ y $\| \cdot \|_r$ son compatibles y las normas de matrices $\| \cdot \|_M$ satisfacen la propiedad 4.

Podemos definir 3 normas subordinadas de matrices como:

$$\| A \|_2 = \max_{x \neq 0} \frac{\| Ax \|_2}{\| x \|_2}$$

$$\| A \|_1 = \max_{x \neq 0} \frac{\| Ax \|_1}{\| x \|_1}$$

$$\| A \|_\infty = \max_{x \neq 0} \frac{\| Ax \|_\infty}{\| x \|_\infty}$$

La primera de estas normas $\| A \|_2$ es difícil de calcular, sin embargo las otras dos las podemos calcular como:

$$\| A \|_1 = \max_{1 \leq j \leq n} \left(\sum_{i=1}^m |a_{ij}| \right)$$

y

$$\| A \|_\infty = \max_{1 \leq i \leq m} \left(\sum_{j=1}^n |a_{ij}| \right)$$

E.2.2. Números condición

Las normas de las matrices pueden usarse para estimar la sensibilidad de un sistema lineal a pequeños cambios en los elementos de la matriz de coeficientes.

Definición: Se dice que una matriz A es "mal condicionada" si cambios relativamente pequeños en sus elementos pueden causar cambios relativamente grandes en la solución a $Ax = b$. Se dice que A es "bien condicionada" si pequeños cambios en sus elementos resultan relativamente pequeños cambios en la solución a $Ax = b$.

Si la matriz A es "mal condicionada", la solución calculada para $Ax = b$ generalmente no será muy exacta. Incluso si los elementos de A pueden representarse exactamente como números de punto flotante, pequeños errores de redondeo generados en el proceso de reducción pueden tener un efecto drástico en la solución calculada. Por otro lado, si la matriz es "bien condicionada" y se usa una técnica de pivoteo apropiada se puede calcular la solución con mayor exactitud. En general, la exactitud de la solución depende del condicionamiento de la matriz. [18].

Si pudiéramos medir el condicionamiento de A , esta medida podría ser usada para determinar un límite para el error relativo de la solución calculada.

Supongamos que A es una matriz no singular de $n \times n$ y considérese el sistema $Ax = b$. Si x es la solución exacta del sistema y x' es la solución calculada, entonces el error puede ser calculado por el vector $e = x - x'$. Si $\| \cdot \|$ es una norma sobre \mathbb{R}^n entonces $\| e \|$ es una medida del error absoluto y $\frac{\|e\|}{\|x\|}$ es una medida del error relativo. En general, no tenemos manera de determinar los valores exactos sobre $\| e \|$ y $\frac{\|e\|}{\|x\|}$. Un posible camino para probar la exactitud de x' es usarla en el sistema original y ver que tan cerca es $b' = Ax'$ de b . Al vector:

$$r = b - b' = b - Ax'$$

se le denomina **vector residual** y puede ser calculado fácilmente, a la cantidad:

$$\frac{\| b - Ax' \|}{\| b \|} = \frac{\| r \|}{\| b \|}$$

se le conoce como **residual relativo**. El criterio para determinar si el residual relativo es un buen estimador del error, lo determina el condicionamiento de A .

En general, demostraremos que si A es mal condicionada, el residual relativo puede ser mucho más pequeño que el error relativo. Por otro lado, para matrices bien condicionadas, el residual relativo y el error relativo son muy parecidos. Recordando que si $\| \cdot \|$ es una norma de matriz compatible sobre $\mathbb{R}^{n \times n}$ entonces para cualquier matriz C de $n \times n$ y cualquier vector $y \in \mathbb{R}^n$, tenemos, ver propiedad 4:

$$\| Cy \| \leq \| C \| \| y \| \quad (\text{E.1})$$

Resultando ahora:

$$r = b - Ax' = Ax - Ax' = A(x - x') = Ae.$$

donde:

$$\epsilon = A^{-1}r$$

De la ecuación E.1:

$$\| \epsilon \| \leq \| A^{-1} \| \| r \|$$

y

$$\| r \| = \| Ae \| \leq \| A^{-1} \| \| r \|$$

Por lo tanto:

$$\frac{\| r \|}{\| A \|} \leq \| \epsilon \| \leq \| A^{-1} \| \| r \| \quad (\text{E.2})$$

Ahora x es la solución exacta para $Ax = b$, y entonces $x = A^{-1}b$. De la misma forma que se obtuvo la ecuación anterior tenemos:

$$\frac{\| b \|}{\| A \|} \leq \| x \| \leq \| A^{-1} \| \| b \| \quad (\text{E.3})$$

De E.2 y E.3:

$$\frac{\| 1 \|}{\| A \| \| A^{-1} \|} \frac{\| r \|}{\| b \|} \leq \frac{\| \epsilon \|}{\| x \|} \leq \| A \| \| A^{-1} \| \frac{\| r \|}{\| b \|}$$

El número $\| A \| \| A^{-1} \|$ se llama número condición de A y se denota $cond(A)$ o $\chi(A)$. De esta forma, [18]:

$$\frac{\| 1 \|}{cond(A)} \frac{\| r \|}{\| b \|} \leq \frac{\| \epsilon \|}{\| x \|} \leq cond(A) \frac{\| r \|}{\| b \|} \quad (\text{E.4})$$

La ecuación E.4 relaciona el tamaño del error relativo $\frac{\| \epsilon \|}{\| x \|}$ con el residual relativo $\frac{\| r \|}{\| b \|}$. Si el número condición es cercano a 1, el error relativo y el residual relativo serán similares. Si el número condición es grande, el número relativo puede ser mucho más grande que el residual relativo.

E.2.3. Matrices positivas

Las matrices de covarianza no singulares son positivas, esta es la razón por la cual las matrices positivas son importantes para el análisis de estimación.

Matrices definidas positivas:

$P(n, n)$ simétrica, es positiva si $x^T P x > 0 \quad \forall n$ vectores x con $\|x\| > 0$.

La expresión $x^T P x$ es una forma cuadrática en la componentes de x . E.g.:

$$x^T P x = \sum_{j=1, j=1}^n P(i, j) x_i x_j$$

Propiedades: Si P es simétrica, $P > 0$ denota que P es positiva:

1. $P > 0$ si todos sus valores característicos son positivos
2. Si $P > 0$, los elementos de su diagonal son positivos
3. Si M es no singular y $P > 0$, entonces $M^T P M$ también es positiva
4. Si $P > 0$, P^{-1} existe y es positiva
5. Si $P > 0$, la matriz obtenida al eliminar un renglón y la correspondiente columna de P también es positiva
6. Si $P > 0$, y $\rho(i, j) = \left[\frac{P(i, j)}{P(i, i)P(j, j)} \right]^{\frac{1}{2}}$, para $i, j = 1, 2, \dots, n$, $i \neq j$, entonces:

$$\rho(i, j) < 1 \text{ y}$$

$\rho(i, j)$ se denomina coeficiente de correlación.

E.2.4. Valores singulares de una matriz

Una matriz H se define positiva o semipositiva sí y solo sí tiene una raíz cuadrada positiva.

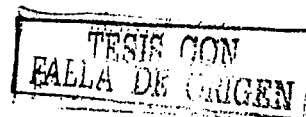
Si $H \geq 0$ o $H > 0$, entonces $(Hx, x) \geq 0$ o $(Hx, x) > 0 \quad \forall x \in \mathbb{R}^n$.

Los valores característicos $\lambda_1, \lambda_2, \dots, \lambda_M$ de la matriz $H = (A^* A)^{\frac{1}{2}}$ se denominan valores singulares S_1, S_2, \dots, S_n de la matriz A .

Así, para $i = 1, \dots, n$

$$S_i(A) = \lambda_i([A^* A]^{\frac{1}{2}})$$

¹Esta es una forma de Cholesky de la matriz H . Ver [4].



Obviamente, los valores singulares de la matriz son no negativos pues hablamos de matrices definidas positivas o semipositivas, $x^T P x > 0$ o $x^T P x \geq 0$.

Bibliografía

- [1] S. H. Ardalan. *Floating-point error analysis of recursive least squares and least mean-squares adaptive filters*. IEEE Transactions Circuits Systems **33** (1986). no. 12. 1192-1208.
- [2] ———. *Fixed-point roundoff error analysis of the exponentially windowed rls algorithm for time varying systems*. IEEE Transactions on Acoustics Speech and Signal Processing **35** (1987). no. 6. 770-783.
- [3] Philip A. Bello. *Characterization of randomly time-variant linear channels*. Performed in part, under subcontract with ITT Communication Systems (1963). no. 4. 360-393.
- [4] Gerard. J. Bierman. *Factorization methods for discrete sequential estimation*. primera ed.. Academic Press. 1977.
- [5] León W. Couch. *Digital and analog communication systems*. quinta ed.. Prentice Hall. 1993.
- [6] Samuel D. Conte & Carl de Boor. *Elementary numerical analysis*. tercera ed.. Mc. Graw Hill. 1975.
- [7] Paulo S. R. Diniz. *Adaptive filtering*. segunda ed.. Kluwer academic Publishers. 1999.
- [8] Michael Verahaegen & Paul Van Dooren. *Numerical aspects of different kalman filter implementation*. IEEE Transactions on Automatic Control **31** (1986). no. 10. 907-917.
- [9] B. Farhang-Boroujeny. *Adaptive filters*. primera ed.. Wiley. 1998.
- [10] C.K. Chui & G.Chen. *Kalman filtering*. tercera ed.. Springer. 1998.
- [11] Simon Haykin. *Introduction to adaptive filters*. primera ed.. Macmillan Publishing Company. 1984.
- [12] ———. *Modern filters*. primera ed.. Macmillan Publishing Company. 1989.
- [13] ———. *Adaptive filter theory*. tercera ed.. Prentice Hall. 1996.

- [14] ———. *Communication systems*, segunda ed., Wiley, 1999.
- [15] Dirk T. M. Slock & Thomas Kailath. *Numerically stable fast transversal filters for recursive least squares adaptive filtering*. IEEE Transactions On Signal Processing **39** (1991), no. 1, 92–114.
- [16] J. M. Cioffi & T. Kailath. *Fast recursive least-squares transversal filters for adaptive filtering*. IEEE Transactions Acoustic Speech Signal Processing **32** (1984), no. 2, 304–337.
- [17] Stephen G. Kellison. *Fundamentals of numerical analysis*, primera ed., Irwing, 1975.
- [18] Steven J. León. *Linear algebra with applications*, quinta ed., Prentice Hall, 1999.
- [19] S. Y. Liao. *Engineering applications of electromagnetic theory*, segunda ed., West Publishing Company, 1988.
- [20] Stefan Ljung & Lennart Ljung. *Error propagation properties of recursive least-squares adaptation algorithms*. Automatica **21** (1985), no. 2, 157–167.
- [21] Jean-Luc Botto & George V. Moustakides. *Stabilizing the fast kalman algorithms*. IEEE Transactions on Acoustics Speech and Signal Processing **37** (1989), no. 9, 1342–1348.
- [22] Jose Alonso Ybañez & Nadir Castañeda Palomino. *Evaluación e implantación de algoritmos de filtrado adaptable para la igualación de canal en sistemas de comunicaciones digitales*. Tesis de licenciatura. Enero 2003.
- [23] David Parsons. *The mobile radio propagation channel*, segunda ed., Wiley, 1992.
- [24] John G. Proakis. *Digital communications*, cuarta ed., Prentice Hall, 1994.
- [25] Shahid U. H. Qureshi. *Adaptive equalization*. Proceedings of the IEEE (1985).
- [26] Larry Hipólito Escobar Salguero. *Algoritmos de filtrado adaptable: Implementación, evaluación, comparación y aplicaciones en telecomunicaciones*. Master's thesis. UNAM, Noviembre 1997.
- [27] Rogelio Alcántara Silva. *Implantation d'algorithmes rapides sur des processeurs de traitement du signal*. Ph.D. thesis. École Nationale Supérieure des Télécommunications, 1986.
- [28] ———. *Apuntes de procesamiento digital de señales*, 2002.
- [29] Lajos Hanzo & Raymond Steele. *Mobile radio communications*, segunda ed., Wiley, 1999.

- [30] G. W. Stewart. *Perturbation bounds for the qr factorization of a matrix*. SIAM J. Numerical Analysis 14 (1977), no. 3. 509-518.
- [31] Sviriv. *Comunicaciones ópticas*. Notas del curso. Enero 1998.
- [32] M. H. Verhaegen. *Round-off error propagation in four generally-applicable, recursive, least-squares estimation schemas*. Automatica 25 (1989), no. 3. 437-444.