



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**PROYECTO DE GRADO PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**TEMA: OPTIMIZACIÓN DE LAS SEÑALES ACÚSTICAS EN UN
ARREGLO SEMIESFÉRICO DE MICRÓFONOS UTILIZANDO
LA METODOLOGÍA BEAMFORMING DE BANDA ANCHA**

AUTOR: CAISAPANTA LARREA, AMELIA VERÓNICA

**DIRECTOR: ING. SÁENZ, FABIÁN
CODIRECTOR: ING. BERNAL, PAÚL**

SANGOLQUÍ, FEBRERO DE 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICACIÓN

Ing. Fabián Sáenz

Ing. Paúl Bernal

CERTIFICAN

Que el trabajo titulado "OPTIMIZACIÓN DE LAS SEÑALES ACÚSTICAS EN UN ARREGLO SEMIESFÉRICO DE MICRÓFONOS UTILIZANDO LA METODOLOGÍA BEAMFORMING DE BANDA ANCHA", realizado por Caisapanta Larrea Amelia Verónica, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la de la Universidad de las Fuerzas Armadas - ESPE.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Amelia Verónica Caisapanta Larrea que lo entreguen al Ingeniero Paúl Bernal, en su calidad de Director de la Carrera.

Sangolquí, febrero de 2015.

Ing. Fabián Sáenz
DIRECTOR

Ing. Paúl Bernal
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, AMELIA VERÓNICA CAISAPANTA LARREA

DECLARO QUE:

El proyecto de grado denominado “OPTIMIZACIÓN DE LAS SEÑALES ACÚSTICAS EN UN ARREGLO SEMIESFÉRICO DE MICRÓFONOS UTILIZANDO LA METODOLOGÍA BEAMFORMING DE BANDA ANCHA”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan el pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, febrero de 2015.



Srta. Amelia Verónica Caisapanta Larrea

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, AMELIA VERÓNICA CAISAPANTA LARREA

Autorizo a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo "OPTIMIZACIÓN DE LAS SEÑALES ACÚSTICAS EN UN ARREGLO SEMIESFÉRICO DE MICRÓFONOS UTILIZANDO LA METODOLOGÍA BEAMFORMING DE BANDA ANCHA", cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, febrero de 2015.



Srta. Amelia Verónica Caisapanta Larrea

DEDICATORIA

Dedico esta tesis a Dios por ser el inspirador de cada uno de mis pasos. A mis padres, mis hermanas Taty y Mayra y mi sobrina Amelí por ser el pilar de mi vida y que con todo su amor han estado conmigo en todo momento.

A mi amiga Diana Lima por su apoyo incondicional. A Luis López por siempre confiar en mí y ser esa fuerza que me ayuda cada día.

A mi director el Ing. Fabián Sáenz por sus valiosos consejos. A mi codirector el Ing. Paúl Bernal por brindarme su ayuda para continuar, cuando parecía que me iba a rendir, le agradezco desde el fondo de mi alma.

AGRADECIMIENTO

Agradezco a Dios y a todas las personas que de alguna forma, son parte de la culminación de este proyecto de investigación.

Mi más sincero agradecimiento hacia el Ing. Paúl Bernal por su apoyo constante y amistad, gracias a su paciencia y enseñanzas, sin él no hubiera podido salir adelante.

ÍNDICE DE CONTENIDOS

| | |
|---------------------------------------------------------|-------------|
| CERTIFICACIÓN | i |
| DECLARACIÓN DE RESPONSABILIDAD | ii |
| AUTORIZACIÓN | iii |
| DEDICATORIA | iv |
| AGRADECIMIENTO | v |
| ÍNDICE DE CONTENIDOS | vi |
| ÍNDICE DE FIGURAS | viii |
| ÍNDICE DE TABLAS | xi |
| ÍNDICE DE ANEXOS | xii |
| RESUMEN | xiii |
| ABSTRACT | xiv |
| GLOSARIO | xv |
| CAPÍTULO I | 1 |
| 1. INTRODUCCIÓN | 1 |
| 1.1. <i>Justificación e Importancia</i> | 1 |
| 1.2. <i>Alcance del Proyecto</i> | 2 |
| 1.3. <i>Objetivos</i> | 3 |
| 1.3.1. <i>Objetivo General</i> | 3 |
| 1.3.2. <i>Objetivos Específicos</i> | 3 |
| CAPÍTULO II | 4 |
| 2. FUNDAMENTO TEÓRICO Y CONCEPTUAL | 4 |
| 2.1. <i>El Sonido</i> | 4 |
| 2.1.1. <i>El sonido como onda sonora</i> | 4 |
| 2.1.2. <i>Perturbaciones de las ondas sonoras</i> | 5 |
| 2.2. <i>La Voz</i> | 5 |
| 2.3. <i>Beamforming</i> | 6 |
| 2.4. <i>Arreglo de Sensores</i> | 8 |
| 2.4.1. <i>Arreglo Lineal Uniforme (ULA)</i> | 9 |
| 2.4.2. <i>Arreglo Circular Uniforme (UCA)</i> | 10 |
| 2.4.3. <i>Aliasing Espacial</i> | 11 |
| 2.5. <i>Dirección de Arribo DOA</i> | 13 |
| 2.6. <i>Algoritmos Adaptativos</i> | 13 |
| CAPÍTULO III | 17 |
| 3. ANÁLISIS DE LOS DIFERENTES ALGORITMOS | 17 |
| 3.1. <i>Comportamiento de los Micrófonos</i> | 17 |

| | | |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------|-----------|
| 3.2. | <i>Estimación de la Dirección de Arribo DOA</i> | 20 |
| 3.2.1. | <i>Algoritmos para la estimación del DOA</i> | 20 |
| 3.2.1.1. | <i>Delay and Sum DAS</i> | 21 |
| 3.2.1.2. | <i>Capon</i> | 22 |
| 3.2.1.3. | <i>MUSIC</i> | 22 |
| 3.2.2. | <i>Costo computacional de los algoritmos para la estimación del DOA</i> | 23 |
| 3.2.3. | <i>Comparación de los algoritmos para la estimación del DOA</i> | 24 |
| 3.3. | <i>Algoritmos Adaptativos LCMV</i> | 24 |
| 3.3.1. | <i>Algoritmos Linearly Constrained</i> | 25 |
| 3.3.1.1. | <i>Algoritmo Constrained Least Mean Squares (CLMS)</i> | 27 |
| 3.3.1.2. | <i>Algoritmo Normalized Constrained Least Mean Squares (NCLMS)</i> | 28 |
| 3.3.1.3. | <i>Algoritmo Constrained Recursive Least-Squares (CRLS)</i> | 29 |
| 3.3.1.4. | <i>Algoritmo Constrained Conjugate Gradient (CCG)</i> | 31 |
| 3.3.2. | <i>Estructura Generalized Sidelobe Canceler (GSC)</i> | 33 |
| 3.3.3. | <i>Transformada de Householder</i> | 39 |
| 3.3.4. | <i>Comparación de las estructuras LCMV</i> | 47 |
| 3.3.4.1. | <i>Comparación de los Algoritmos Constrained</i> | 47 |
| 3.3.4.2. | <i>Comparación de la Estructura GSC</i> | 49 |
| 3.3.4.3. | <i>Comparación de la Estructura Householder</i> | 51 |
| 3.3.4.4. | <i>Velocidad de convergencia de los algoritmos LCMV</i> | 53 |
| 3.3.4.5. | <i>Costo computacional de los algoritmos LCMV</i> | 54 |
| 3.4. | <i>Análisis del Número de Taps</i> | 56 |
| CAPÍTULO IV | | 62 |
| 4. ANÁLISIS Y SIMULACIÓN DE LOS ALGORITMOS EN LA GEOMETRÍA PROPUESTA | | 62 |
| 4.1. | <i>Diseño del arreglo</i> | 62 |
| 4.2. | <i>Comparación del desempeño de los algoritmos</i> | 64 |
| 4.2.1.1. | <i>Comparación de la Estructura Constrained</i> | 64 |
| 4.2.1.2. | <i>Comparación de la Estructura GSC</i> | 66 |
| 4.2.1.3. | <i>Comparación de la Estructura Householder</i> | 68 |
| 4.3. | <i>Comparación del Diseño Propuesto</i> | 70 |
| CAPÍTULO V | | 73 |
| 5. CONCLUSIONES Y RECOMENDACIONES | | 73 |
| 5.1. | <i>Conclusiones</i> | 73 |
| 5.2. | <i>Recomendaciones</i> | 74 |
| BIBLIOGRAFÍA | | 75 |

¡Error! Marcador no definido.

ÍNDICE DE FIGURAS

| | |
|--------------------------------------------------------------------------------------------------|----|
| Figura 1. Narrowband Beamforming | 7 |
| Figura 2. Broadband Beamforming..... | 8 |
| Figura 3. Arreglo Lineal Uniforme ULA..... | 10 |
| Figura 4. Arreglo Circular Uniforme UCA..... | 11 |
| Figura 5. Patrón de radiación con espaciamento variable entre sensores para un arreglo ULA | 12 |
| Figura 6. Estructura de Beamforming Adaptativo Broadband..... | 14 |
| Figura 7. Arreglo ULA con M=4 y d=0.1 m..... | 17 |
| Figura 8. Respuesta de los Micrófonos en Frecuencia para un arreglo ULA | 18 |
| Figura 9. Respuesta Compensada de los Micrófonos en Frecuencia para un arreglo ULA | 18 |
| Figura 10. Señales Originales para un arreglo ULA | 19 |
| Figura 11. Señales Corregidas el Retardo para un arreglo ULA..... | 19 |
| Figura 12. Resultados de la ecualización del Mic 1 para un arreglo ULA | 20 |
| Figura 13. Comparación de los algoritmos para la estimación del DOA para un arreglo ULA | 24 |
| Figura 14. Algoritmos Constrained respecto al MSE para un arreglo ULA (C original)..... | 26 |
| Figura 15. Algoritmos Constrained respecto al MSE para un arreglo ULA (C modificada) | 26 |
| Figura 16. MSE del algoritmo CLMS con varios tamaños de μ para un arreglo ULA | 28 |
| Figura 17. MSE del algoritmo NCLMS con varios tamaños de μ_n para un arreglo ULA | 29 |
| Figura 18. MSE del algoritmo CRLS con varios tamaños de λ para un arreglo ULA | 31 |
| Figura 19. MSE del algoritmo CCG con varios tamaños de λ_f para un arreglo ULA | 32 |
| Figura 20. MSE del algoritmo CCG con varios tamaños de η para un arreglo ULA | 33 |
| Figura 21. Estructura Generalized Sidelobe Canceling (GSC)..... | 34 |
| Figura 22. MSE del algoritmo GSC-LMS con varios tamaños de μ para un arreglo ULA | 36 |
| Figura 23. MSE del algoritmo GSC-NLMS con varios tamaños de μ_n para un arreglo ULA | 37 |
| Figura 24. MSE del algoritmo GSC-RLS con varios tamaños de λ para un arreglo ULA | 37 |
| Figura 25. MSE del algoritmo GSC-CG con varios tamaños de λ_f para un arreglo ULA | 38 |
| Figura 26. MSE del algoritmo GSC-CG con varios tamaños de η para un arreglo ULA | 39 |
| Figura 27. MSE del algoritmo HCLMS con varios tamaños de μ para un arreglo ULA | 41 |
| Figura 28. MSE del algoritmo HNCLMS con varios tamaños de μ_n para un arreglo ULA | 42 |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 29. MSE del algoritmo HCRLS con varios tamaños de λ para un arreglo ULA | 42 |
| Figura 30. MSE del algoritmo HCCG con varios valores de λ_f para un arreglo ULA | 43 |
| Figura 31. MSE del algoritmo HCCG con varios valores de η para un arreglo ULA | 44 |
| Figura 32. MSE de las estructuras del algoritmo RLS para un arreglo ULA | 47 |
| Figura 33. Comparación de los algoritmos Constrained respecto al MSE para un arreglo ULA | 48 |
| Figura 34. Comparación de los algoritmos Constrained respecto al MSE para un arreglo ULA con todas las interacciones | 48 |
| Figura 35. Comparación de los algoritmos Constrained respecto al beampattern para un arreglo ULA | 49 |
| Figura 36. Comparación de los algoritmos GSC respecto al MSE para un arreglo ULA | 50 |
| Figura 37. Comparación de los algoritmos GSC respecto al MSE para un arreglo ULA con todas las interacciones..... | 50 |
| Figura 38. Comparación de los algoritmos GSC respecto al beampattern para un arreglo ULA | 51 |
| Figura 39. Comparación de los algoritmos Householder respecto al MSE para un arreglo ULA | 52 |
| Figura 40. Comparación de los algoritmos Householder respecto al MSE para un arreglo ULA con todas las interacciones | 52 |
| Figura 41. Comparación de los algoritmos Householder respecto al beampattern para un arreglo ULA | 53 |
| Figura 42. Algoritmos Constrained respecto al MSE con varios valores de N para un arreglo ULA | 57 |
| Figura 43. Algoritmos Constrained respecto al beampattern con varios valores de N para un arreglo ULA | 58 |
| Figura 44. Algoritmos GSC respecto al MSE con varios valores de N para un arreglo ULA | 59 |
| Figura 45. Algoritmos GSC respecto al beampattern con varios valores de N para un arreglo ULA | 59 |
| Figura 46. Algoritmos Householder respecto al MSE con varios valores de N para un arreglo ULA | 60 |
| Figura 47. Algoritmos Householder respecto al beampattern con varios valores de N para un arreglo ULA..... | 60 |
| Figura 48. Arreglo Semiesférico | 62 |
| Figura 49. Comparación de los algoritmos Constrained respecto al MSE para un arreglo semiesférico | 65 |
| Figura 50. Comparación de los algoritmos Constrained respecto al MSE para un arreglo semiesférico con todas las interacciones | 65 |
| Figura 51. Comparación de los algoritmos Constrained respecto al beampattern para un arreglo semiesférico | 66 |
| Figura 52. Comparación de los algoritmos GSC respecto al MSE para un arreglo semiesférico..... | 67 |
| Figura 53. Comparación de los algoritmos GSC respecto al MSE para un arreglo semiesférico con todas las interacciones..... | 67 |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 54. Comparación de los algoritmos GSC respecto al beampattern para un arreglo semiesférico | 68 |
| Figura 55. Comparación de los algoritmos Householder respecto al MSE para un arreglo semiesférico | 69 |
| Figura 56. Comparación de los algoritmos Householder respecto al MSE para un arreglo semiesférico con todas las interacciones | 69 |
| Figura 57. Comparación de los algoritmos Householder respecto al beampattern para un arreglo semiesférico | 70 |
| Figura 58. Comparación de los algoritmos HCCG y MUSIC respecto al beampattern para un arreglo semiesférico..... | 72 |

ÍNDICE DE TABLAS

| | |
|--------------------------------------------------------------------------------|---------|
| Tabla 1. Costo computacional de los algoritmos para la estimación del DOA..... | 23 |
| Tabla 2. MSE en dB del algoritmo LMS con μ variable | 44 |
| Tabla 3. MSE en dB del algoritmo NLMS con μ_n variable | 45 |
| Tabla 4. MSE en dB del algoritmo RLS con λ variable..... | 45 |
| Tabla 5. MSE en dB del algoritmo CG con λ_f variable | 45 |
| Tabla 6. MSE en dB del algoritmo CG con η variable | 46 |
| Tabla 7. Velocidad de convergencia de los algoritmos LCMV | 54 |
| Tabla 8. Costo computacional de los algoritmos LCMV..... | 55 |
| Tabla 9. Algoritmo CLMS | Anexo A |
| Tabla 10. Algoritmo NCLMS | Anexo A |
| Tabla 11. Algoritmo CRLS | Anexo A |
| Tabla 12. Algoritmo CCG..... | Anexo A |
| Tabla 13. Algoritmo GSC-LMS..... | Anexo B |
| Tabla 14. Algoritmo GSC-NLMS..... | Anexo B |
| Tabla 15. Algoritmo GSC-RLS..... | Anexo B |
| Tabla 16. Algoritmo GSC-CG | Anexo B |
| Tabla 17. Algoritmo HCLMS | Anexo C |
| Tabla 18. Algoritmo HNCLMS | Anexo C |
| Tabla 19. Algoritmo HCRLS | Anexo C |
| Tabla 20. Algoritmo HCCG..... | Anexo C |

ÍNDICE DE ANEXOS

ANEXO A: ALGORITMOS *CONSTRAINED*

ANEXO B: ALGORITMOS GSC

ANEXO C: ALGORITMOS *HOUSEHOLDER*

ANEXO D: CÓDIGOS PARA LA ESTIMACIÓN DEL DOA

ANEXO E: CÓDIGO DEL ALGORITMO HCCG

RESUMEN

Al ser la voz la principal forma de comunicación que posee el ser humano, la presente investigación busca ayudar a las personas con problemas de audición, mediante el realzado de la voz. Se utiliza las técnicas de *beamforming* y la Dirección de Arribo de una señal (DOA) en un arreglo de micrófonos semiesférico para su implementación futura en un casco. Este trabajo es el estudio matemático de los algoritmos adaptativos *Linearly Constrained Minimum Variance* (LCMV) con sus tres familias: *Constrained*, *Generalized Sidelobe Canceler* (GSC) y *Householder*, utilizando los algoritmos *unconstrained*: *Least Mean Squares* (LMS), *Normalized Least Mean Squares* (NLMS), *Recursive Least-Squares* (RLS) y *Conjugate Gradient* (CG). La herramienta Matlab® permite analizar las cualidades y desventajas de cada uno de los algoritmos y se optimizan sus desempeños variando sus respectivos parámetros de adaptación, a fin de obtener una rápida convergencia sin comprometer el *Mean Squared Error* (MSE). Los algoritmos adaptativos se comparan y en base a los resultados obtenidos se selecciona el algoritmo más idóneo, tomando en cuenta la velocidad de convergencia, costo computacional e implementación en hardware para su futura implementación en un dispositivo *Field Programmable Gate Array* (FPGA), todos los resultados son analizados a fin de obtener las conclusiones y recomendaciones de la investigación.

PALABRAS CLAVES:

MEJORAMIENTO DE LA VOZ,

BEAMFORMING,

DIRECTION OF ARRIVAL,

ALGORITMOS ADAPTATIVOS,

MATLAB®.

ABSTRACT

For being the voice the principal form of communication that the human possess. This investigation searching help people who have hearing problems, through improvement of the voice. We use techniques of beamforming and Direction of arrival (DOA) about hemispherical microphone array for the future implementation in a helmet. This work is the mathematic study of adaptive algorithms Linearly Constrained Minimum Variance (LCMV) about their three families: Constrained, Generalized Sidelobe Canceller (GSC) and Householder using the unconstrained algorithms: Least Mean Squares (LMS), Normalized Least Mean Squares (NLMS), Recursive Least-Squares (RLS) y Conjugate Gradient (CG). The Matlab®'s tool allows us to analyze the qualities and each disadvantage of the algorithms and the improvement of their performance varying their respective parameters, in order to get a fast convergence without involves the Mean Squared Error (MSE). The adaptive algorithms are compared based on the results obtained the suitable algorithm is chosen, considering the speed of the convergence, computational cost and hardware implementation for the future implementation in a *Field Programmable Gate Array* (FPGA) device, all results are analyzed to obtain the conclusions and recommendations of the research.

KEYWORDS:

IMPROVEMENT OF THE VOICE,

BEAMFORMING, DIRECTION OF ARRIVAL,

ADAPTIVE ALGORITHMS,

MATLAB®.

GLOSARIO

| | |
|--------|-----------------------------------------------------------------------------|
| AWGN | <i>Additive White Gaussian Noise.</i> |
| CCG | <i>Constrained Conjugate Gradient.</i> |
| CG | <i>Conjugate Gradient.</i> |
| CLMS | <i>Constrained Least Mean Squares.</i> |
| CRLS | <i>Constrained Recursive Least-Squares</i> |
| DAS | <i>Delay and Sum.</i> |
| DOA | <i>Direction of Arrival.</i> |
| DSP | <i>Digital Signal Processing.</i> |
| ESPRIT | <i>Estimation of Signal Parameters via Rotational Invariance Technique.</i> |
| FPGA | <i>Field Programmable Gate Array.</i> |
| GSC | <i>Generalized Sidelobe Canceler.</i> |
| LCMV | <i>Linearly Constrained Minimum Variance.</i> |
| LMMSE | <i>Linear Minimum Mean Square Error.</i> |
| LMS | <i>Least Mean Squares.</i> |
| MOE | <i>Minimum Output Energy.</i> |
| MSE | <i>Mean Squared Error.</i> |
| MSE | <i>Mean Squared Error.</i> |
| MUSIC | <i>Multiple Signal Classification.</i> |
| NCLMS | <i>Normalized Constrained Least Mean Squares</i> |
| NLMS | <i>Normalized Least Mean Squares.</i> |
| RLS | <i>Recursive Least-Squares.</i> |
| TIC | Tecnologías de la Información y la Comunicación. |
| UCA | <i>Uniform Circular Array.</i> |
| ULA | <i>Uniform Linear Array.</i> |
| VLSI | <i>Very Large Scale Implementation.</i> |

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Justificación e Importancia

Los sistemas de Procesamiento Digital de Señales (*Digital Signal Processing* DSP) se han venido desarrollando de forma sostenible en una variedad de áreas, incluyendo el procesamiento de la voz, procesamiento de imágenes, procesamiento de video, radar, sonar, etc.

El procesamiento digital de señales de voz posee una amplia base teórica y experimental que se ha desarrollado en conjunto con el DSP durante los últimos 80 años. Además, existe una tecnología de implementación a gran escala (*Very Large Scale Implementation* VLSI) que se encuentra bien adaptada a las demandas computacionales del procesamiento digital de señales de voz (Rabiner & R. W., 2009).

El estudio de las señales de voz es un tema de importancia, dado que una gran parte de la sociedad presenta algún tipo de discapacidad auditiva, por tanto la presente investigación pretende ser la base matemática para la implementación a futuro de soluciones que mejoren la calidad de vida de dichas personas.

Existen varias técnicas para el procesamiento de señales sonoras, dependiendo de la aplicación que se realice, en este caso se utiliza la metodología de *beamforming* y la estimación de la dirección de arribo de las señales (*Direction of Arrival* DOA) que tradicionalmente eran estudiados en la teoría de antenas, substituyendo las ondas electromagnéticas por señales acústicas y el arreglo de antenas por un arreglo de micrófonos, presentando varias aplicaciones en diferentes campos como:

videoconferencias, realzado de voz, manos libre para telefonía móvil, sistemas de ayuda para la escucha, sistemas de controladores de voz, reconocimiento de voz, etc.

Los algoritmos son los encargados de realizar tales tareas, dentro de los cuales se puede nombrar: DAS (*Delay and Sum*), Capon, MUSIC (*Multiple Signal Classification*), Root-MUSIC, ESPRIT (*Estimation of Signal Parameters via Rotational Invariance Technique*) (Yang, Cho, & Choo, 2012) y LCMV (*Linearly Constrained Minimum Variance*) (Van Veen & Buckley, 2000).

El término TIC (Tecnologías de la Información y la Comunicación) surge como convergencia tecnológica de la electrónica, el software, las telecomunicaciones y ciertas ideas de administración y manejo de información, considerando como elemento clave los procesos de información y comunicación, en virtud, el hecho de poder tener dispositivos que permitan mejorar la calidad de vida a personas con discapacidad auditiva, resulta importante.

1.2. Alcance del Proyecto

A partir de la revisión del estado del arte, en la presente investigación, se estudia el desempeño de varios algoritmos para la estimación del DOA. Además se estudia los algoritmos *beamforming* adaptativo, específicamente los algoritmos LCMV. Para lo cual, se determina el número y la geometría más idónea de los micrófonos en un arreglo semiesférico que permita realzar la voz para su implementación futura en un casco.

Los algoritmos mencionados tienen cualidades y desventajas que serán verificadas, como es el caso del costo computacional, velocidad de convergencia y tipo de implementación en hardware. Dentro de esto, se puede acotar que se va emplear el LMMSE (*Linear Minimum Mean Square Error*) para reducir el error.

Se utiliza la herramienta de simulación Matlab® con el fin de comparar el rendimiento de los diferentes algoritmos. En base a los resultados, se seleccionará el

algoritmo que proporcione el mejor desempeño para su futura implementación en un dispositivo de Arreglos de Compuertas Programables en Campo (*Field Programmable Gate Array* FPGA).

1.3. Objetivos

1.3.1. Objetivo General

Determinar el algoritmo adaptativo más eficiente para un arreglo semiesférico de micrófonos que permita realzar la voz de una manera óptima.

1.3.2. Objetivos Específicos

- Estudiar los algoritmos empleados para la estimación del DOA.
- Analizar los diferentes algoritmos adaptativos LCMV.
- Implementar los algoritmos para la estimación del DOA y los algoritmos LCMV utilizando la herramienta Matlab.
- Optimizar los algoritmos, variando los parámetros de adaptación del mismo para obtener una rápida convergencia sin aumentar el error cuadrático medio (*Mean Squared Error* - MSE).
- Analizar los algoritmos y establecer el más idóneo tomando en cuenta la velocidad de convergencia, costo computacional e implementación en hardware.
- Determinar el número de sensores ideal para un arreglo de geometría semiesférico.

CAPÍTULO II

2. FUNDAMENTO TEÓRICO Y CONCEPTUAL

2.1. El Sonido

El sonido es una vibración que se propaga en forma de onda a través del aire, la cual transmite la vibración hasta llegar al oído. El oído humano es capaz de percibir sonidos de frecuencias comprendidas entre los 20 Hz y los 20 kHz, conocida como frecuencia audible. Por debajo de este rango se tiene los infrasonidos y por encima los ultrasonidos.

2.1.1. El sonido como onda sonora

El sonido es una onda mecánica longitudinal que necesita un medio para propagarse, esta no puede propagarse en el vacío. Si se supone un medio ideal, las ondas sonoras se propagan según la ecuación lineal de onda que se detalla en la Ecuación (1).

$$\nabla^2 p(\vec{r}, t) - \frac{1}{\vartheta^2} \frac{\partial^2}{\partial t^2} p(\vec{r}, t) = 0 \quad (1)$$

Donde $p(\vec{r}, t)$ representa la presión sonora del sonido en un punto \vec{r} del espacio en un instante de tiempo t , ∇^2 el operador Laplaciano y ϑ la velocidad del sonido en metros por segundo. La velocidad de propagación del sonido depende de las características del medio. En el aire, a nivel del mar y con una temperatura promedio de 20°C, la velocidad aproximada del sonido es de 340 m/s.

La Ecuación (1) se resuelve usando el método de separación de variables y se obtiene.

$$p(r, t) = -\frac{A}{4\pi r} e^{j(\omega t - \frac{2\pi}{\lambda} r)} \quad (2)$$

Donde A es la amplitud de la onda, r es la distancia radial desde la fuente en metros y ω es la frecuencia angular en radianes por segundo.

2.1.2. Perturbaciones de las ondas sonoras

Las ondas sonoras sufren un fenómeno llamado atenuación, conforme se van alejando de su fuente emisora, disminuyendo su amplitud. La atenuación es el resultado de mecanismos tales como: divergencia geométrica, absorción del aire y absorción por superficies sólidas.

En ambientes cerrados el fenómeno predominante en la propagación del sonido es la reflexión, provocando dos fenómenos conocidos como: el eco y la reverberación, los mismos que se diferencian por el tiempo que dura la reflexión.

- **El Eco.-** Es un efecto que se produce cuando las ondas reflejadas se perciben con el tiempo suficiente para poder distinguirla del sonido original. El oído humano puede captar y procesar sonidos que son producidos en un intervalo mayor a 0.1 s.
- **Reverberación.-** Consiste en todas las reflexiones del sonido que llegan al oyente. Estas reflexiones se dan en todas las direcciones, en intervalos de tiempos cortos y casi aleatorios, formando la difusión sonora. Las reflexiones están dentro de un intervalo de tiempo menor a 0.1 s, siendo este un fenómeno desapercibido para el oído humano.

2.2. La Voz

La finalidad de la voz es la comunicación entre los seres humanos, es decir, la transmisión de mensajes. El proceso de comunicación por medio de la voz comienza

con el pensamiento de un mensaje representado de forma abstracta en el cerebro del locutor. A través del complejo proceso de producción de la voz, la información se convierte en última instancia en una señal acústica (Rabiner & R. W., 2009).

La voz es la principal forma de comunicación del ser humano y se caracteriza por ser la combinación de varias frecuencias con sus correspondientes armónicos. Sin embargo, a comparación del rango completo de la audición humana, la voz cubre un rango relativamente estrecho de frecuencias entre 100 Hz hasta 6 kHz.

La voz masculina tiene un tono fundamental (*pitch*) entre 100 y 200 Hz y la voz femenina típicamente se encuentra entre 150 y 300 Hz. Las mujeres y los niños por lo general poseen tonos más agudos que los hombres.

2.3. *Beamforming*

Beamforming es una técnica de filtrado espacial cuya finalidad, es estimar una señal que arriba en una determinada dirección de interés y anular o interferir señales que arriban en direcciones no deseadas, utilizando un arreglo de sensores (Van Veen & Buckley, 2000).

Existen dos tipos de *beamforming*. El primero se presenta en la Figura 1, que muestrea el campo de ondas en el espacio, que es típicamente usado para procesar señales de ancho de banda estrecho (*narrowband signals*). La salida $y[k]$ en el tiempo discreto k del *narrowband beamforming*, es dada por la combinación lineal de los datos de los M sensores en el tiempo k (Madisetti, 2010).

$$y[k] = \sum_{m=1}^M w_m x_m[k] \quad (3)$$

El segundo tipo de *beamforming* se presenta en la Figura 2. Este tipo de *beamforming*, muestrea el campo de ondas propagadas en tiempo y espacio, se

utiliza cuando las señales de interés son de amplia extensión en frecuencia (*broadband signals*). La salida en este caso se expresa como:

$$y[k] = \sum_{m=1}^M \sum_{i=0}^{N-1} w_{m,i} x_m[k-i] \quad (4)$$

Donde $N - 1$ es el número de retardos en cada uno de los M sensores. Si se considera la señal en cada sensor como una entrada, entonces un *beamforming* representa un sistema multientrada con una única salida (Madisetti, 2010).

Para los dos tipos de *beamforming* se puede utilizar la siguiente expresión vectorial que incluye la Ecuación (3) y la Ecuación (4).

$$y[k] = \mathbf{w}^H \mathbf{x}[k] \quad (5)$$

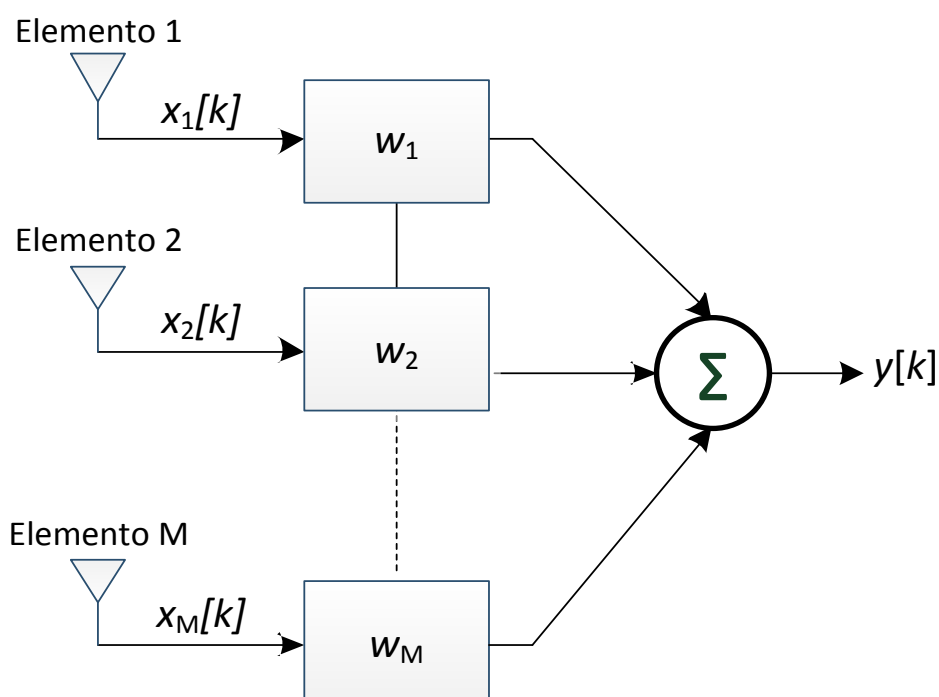


Figura 1. Narrowband Beamforming
Fuente: (Madisetti, 2010)

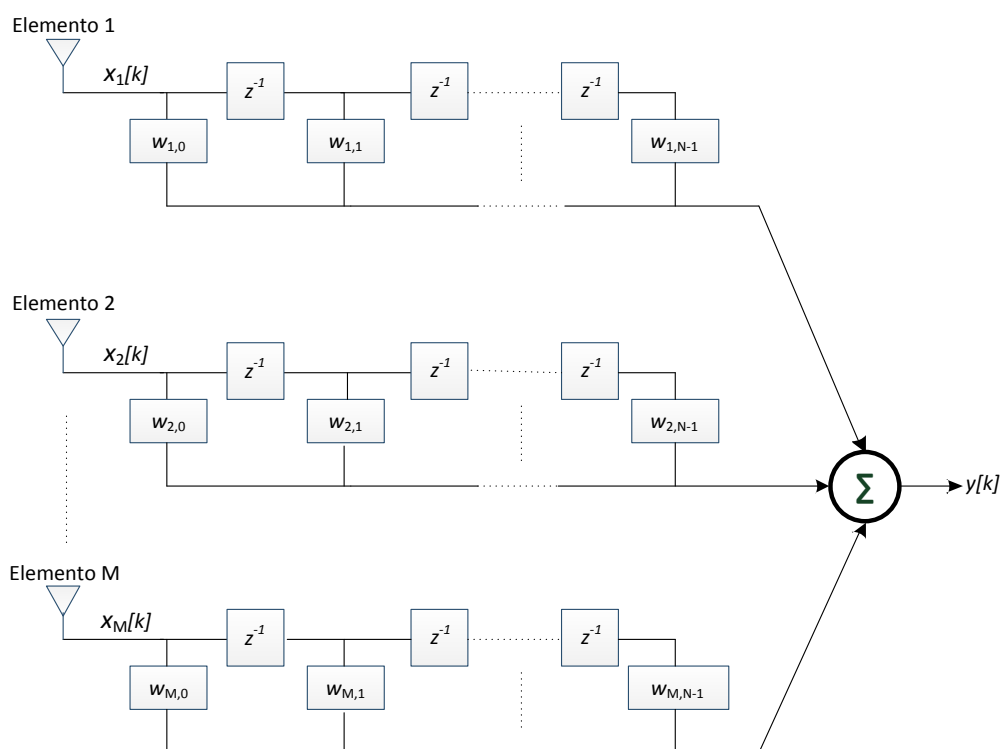


Figura 2. Broadband Beamforming
Fuente: (Madisetti, 2010)

Dependiendo de la forma en que se estiman los pesos (coeficientes del filtro) se puede tener dos categorías: *beamforming* fijo, en el cual los pesos que están asociados a cada sensor son fijos y el *beamforming* adaptativo en donde los pesos son ajustados dinámicamente dependiendo de los datos recibidos, este último surge cuando no se conoce las propiedades estadísticas de las señales y se emplea para el diseño las estadísticas recogidas en línea.

2.4. Arreglo de Sensores

Un arreglo de sensores es un conjunto de elementos dispuestos en una determinada geometría, de manera que cada sensor se combinada para obtener una mayor directividad y sensibilidad, brindando mayor ganancia y ancho de banda que la que se experimenta con un solo elemento (Monzingo, Haupt, & Miller, 2011).

Antes de analizar la expresión general de las señales que arriban a cada elemento del arreglo es necesario conocer que una onda plana llega al punto \vec{r} en un tiempo t según la siguiente expresión (Yang, Cho, & Choo, 2012).

$$s(\mathbf{r}, t) = Ae^{j(\boldsymbol{\beta} \cdot \mathbf{r} - \omega t)} \quad \text{con} \quad \left\{ \beta = \|\boldsymbol{\beta}\| = \frac{2\pi}{\lambda} = \frac{2\pi}{v/f} \right. \quad (6)$$

Donde A es la amplitud de onda, $\boldsymbol{\beta}$ el vector de onda (propagación), β factor de onda (número de ondas), λ es la longitud de onda, v velocidad de propagación de las ondas, f es la frecuencia. Según la Ecuación (6) se puede escribir que la señal llega a un elemento en la posición (r_x, r_y, r_z) en términos de la señal $s(\mathbf{0}, t)$ que arriba al origen $(0, 0, 0)$ como se detalla en la Ecuación (7).

$$s(\mathbf{r}, t) = s(\mathbf{0}, t)e^{j\beta(r_x \cos \varphi \sin \theta + r_y \cos \varphi \sin \theta + r_z \cos \varphi)} \quad (7)$$

Donde φ y θ representan el ángulos azimut y el ángulo de elevación respectivamente.

El Arreglo Lineal Uniforme (*Uniform Linear Array ULA*) y el Arreglo Circular Uniforme (*Uniform Circular Array UCA*) son los más utilizados, presentando grandes ventajas, las cuales se presentan en los siguientes capítulos. Hoy en día, estas dos geometrías dominan el mercado, pero sofisticados arreglos esféricos tridimensionales cada vez son más utilizados, ya que pueden captar mejor el campo de sonido (Benesty, Chen, & Huang, 2008).

2.4.1. Arreglo Lineal Uniforme (ULA)

Para un arreglo ULA los M sensores isotrópicos están separados a una distancia constante d a lo largo de una línea recta, como se muestra en la Figura 3. La señal $s(t)$ arriba al m -ésimo sensor según la Ecuación (8).

$$s_m(t) = s\{(md, 0, 0), t\} = s(\mathbf{0}, t)a_m(\varphi, \theta) \quad (8)$$

Donde el factor de dirección (*steering factor*) del m-ésimo elemento es:

$$a_m(\varphi, \theta) = e^{j\beta m d \cos \varphi \sin \theta} \quad (9)$$

$$a_m(\varphi, \theta) = e^{j2\pi m \frac{d}{\lambda} \cos \varphi \sin \theta} \quad (10)$$

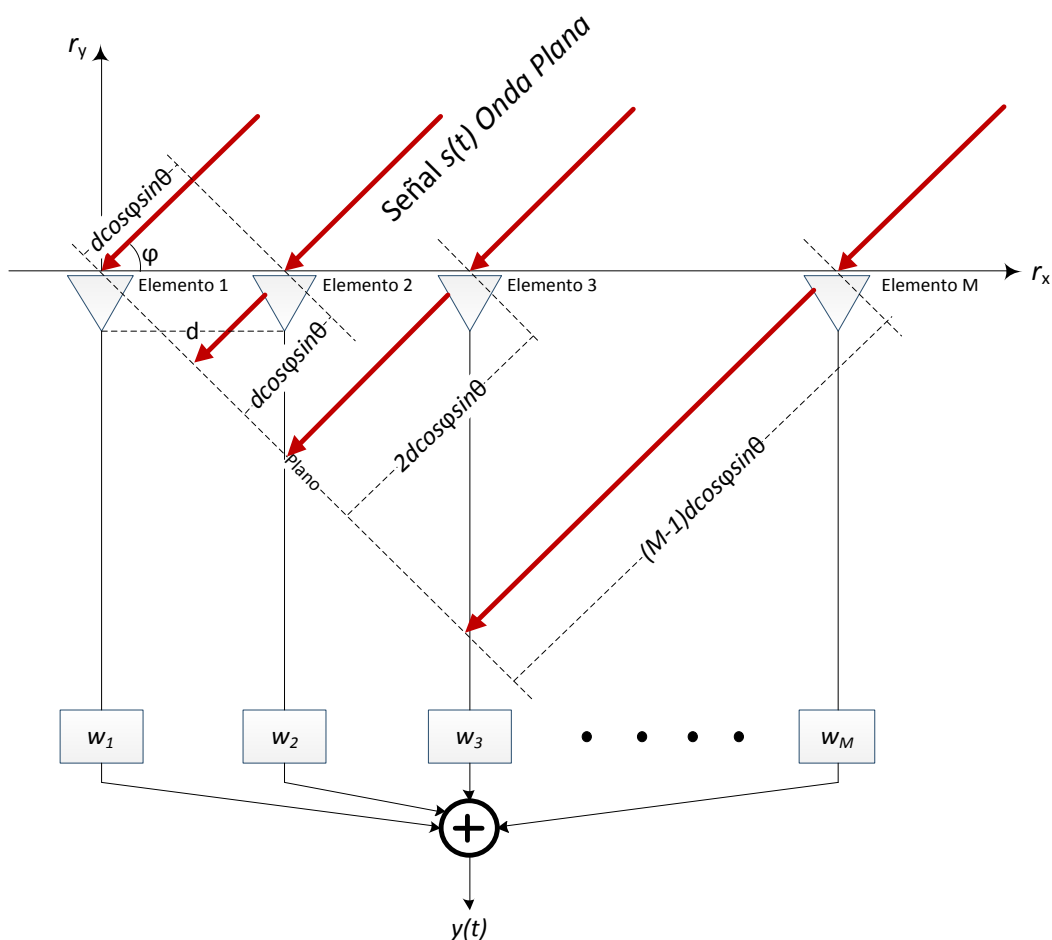


Figura 3. Arreglo Lineal Uniforme ULA

Fuente: (Yang, Cho, & Choo, 2012)

2.4.2. Arreglo Circular Uniforme (UCA)

La Figura 4 muestra el arreglo UCA de M sensores isotrópicos separados a una distancia constante de $\frac{2\pi r}{M}$ a lo largo de un círculo de radio r . Las señales arriban al m-ésimo sensor según la Ecuación (11).

$$s_m(t) = s\left\{\left(re^{j\frac{2\pi m}{M}}\right), t\right\} = s(\mathbf{0}, t)a_m(\varphi, \theta) \quad (11)$$

Donde el *steering factor* del m-enésimo elemento es:

$$a_m(\varphi, \theta) = e^{j\beta r \sin \theta \cos\left(\varphi - \frac{2\pi m}{M}\right)} \quad (12)$$

$$a_m(\varphi, \theta) = e^{j2\pi \frac{r}{\lambda} \sin \theta \cos\left(\varphi - \frac{2\pi m}{M}\right)} \quad (13)$$

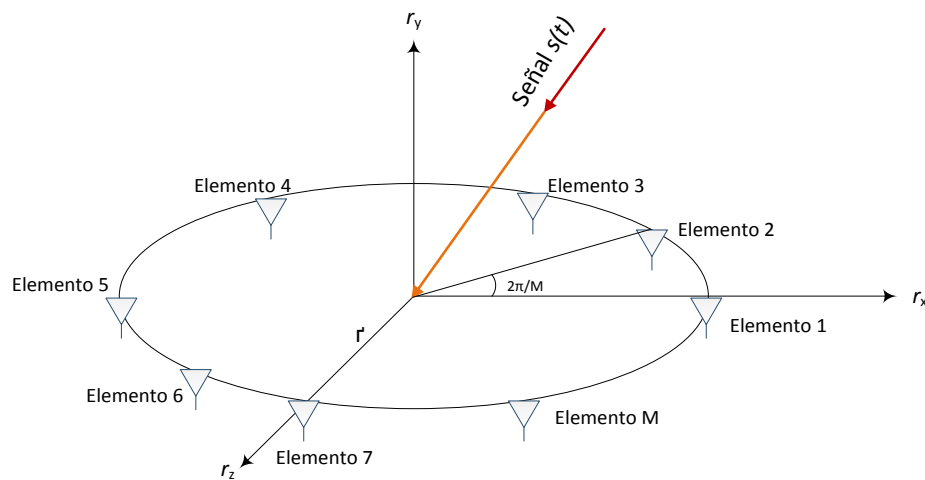


Figura 4. Arreglo Circular Uniforme UCA
Fuente: (Yang, Cho, & Choo, 2012)

2.4.3. Aliasing Espacial

De acuerdo al teorema de Nyquist, la frecuencia de muestreo debe ser mayor o igual que el doble de la máxima componente de frecuencia de la señal, para evitar el solapamiento (*aliasing*) en frecuencia. Análogamente el *aliasing* espacial sucede cuando la distancia de separación entre cada uno de los micrófonos no es la adecuada, provocando que estos sean incapaces de distinguir entre señales que arriban de diferentes direcciones. Para evitar el *aliasing* espacial es necesario que la distancia entre los micrófonos cumpla la condición que se detalla en la Ecuación (14)

$$d \leq \frac{\lambda_{min}}{2} \quad (14)$$

Donde λ_{min} es la mínima longitud de onda de la señal de interés. La Figura 5 muestra el comportamiento de selectividad de un arreglo en función de la distancia de separación de los sensores. La simulación se realizó para un arreglo ULA con 8 sensores y con un tono sinusoidal de 1 kHz en el cual se varía la distancia desde 1 cm a 34 cm. El espaciado óptimo entre sensores para una señal de 1 kHz es de aproximadamente 17 cm.

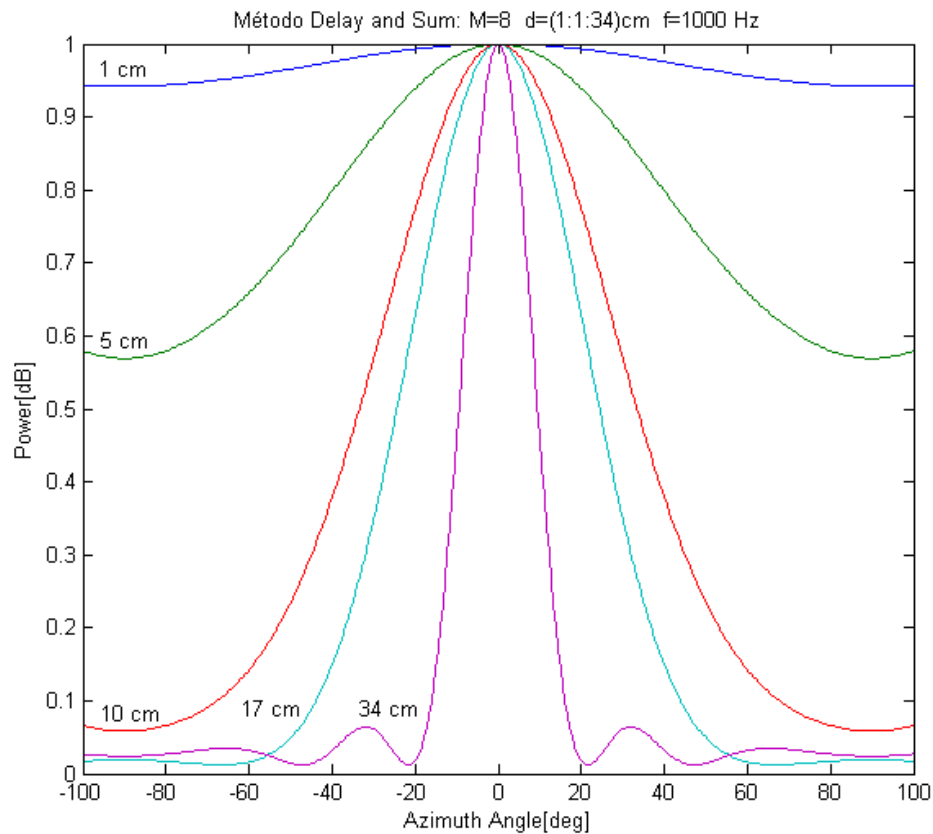


Figura 5. Patrón de radiación con espaciamento variable entre sensores para un arreglo ULA

Según la Figura 5 se puede deducir que cuando el espaciamento entre sensores disminuye el lóbulo principal se ensancha y a medida que la distancia es mayor el lóbulo principal es más directivo, pero la señal empieza a distorsionarse, por tanto la distancia de espaciamento entre sensores debe ser la adecuada.

2.5. Dirección de Arribo DOA

Los distintos métodos para la estimación del DOA, estiman el ángulo azimuth φ para 1-D, una dirección y el par de ángulos (φ, θ) para 2-D, dos dimensiones, para lo cual se basan en las señales que arriban en el arreglo de sensores (Yang, Cho, & Choo, 2012), como se mostró en la Figura 3.

La literatura dispone de varios métodos para la estimación del DOA que ofrecen diferentes desempeños y costos computacionales. Los métodos que se estudian en la presente investigación son: *Delay and Sum*, Capon y MUSIC (Yang, Cho, & Choo, 2012). Los dos primeros métodos se basan en la estimación espectral de potencia y se caracterizan por calcular el espectro espacial de las señales recibidas y estimar la dirección a partir de los máximos encontrados.

El método MUSIC se basa en la teoría de subespacios que explora las propiedades de la estructura de la matriz de autocorrelación de las señales que arriban al arreglo, la cual es separada en dos partes: subespacio del ruido y subespacio de la señal, después se realiza el cálculo de la potencia espacial, se encuentran los máximos y se hace la estimación del subespacio de la señal. En el siguiente capítulo se detallan cada uno de estos métodos.

2.6. Algoritmos Adaptativos

Los algoritmos adaptativos intentan minimizar una función de costo $\xi[k]$ y que generalmente se relaciona con el error de salida (Diniz, 2013). En la Figura 6 se muestra la estructura de un *beamforming* adaptativo *broadband* con M sensores y N *taps* (número de coeficientes del filtro). La salida de la estructura de la Figura 6 se expresa como:

$$y[k] = \mathbf{w}^H[k]\mathbf{x}[k] \quad (15)$$

Donde

$$\mathbf{w}[k] = [\mathbf{w}_1^T[k] \quad \mathbf{w}_2^T[k] \quad \cdots \quad \mathbf{w}_M^T[k]]^T \quad (16)$$

$$\mathbf{x}[k] = [\mathbf{x}_1^T[k] \quad \mathbf{x}_2^T[k] \quad \cdots \quad \mathbf{x}_M^T[k]]^T \quad (17)$$

y

$$\mathbf{x}_i^T[k] = [x_i[k] \quad x_i[k] \quad \cdots \quad x_i[k - N + 1]] \quad (18)$$

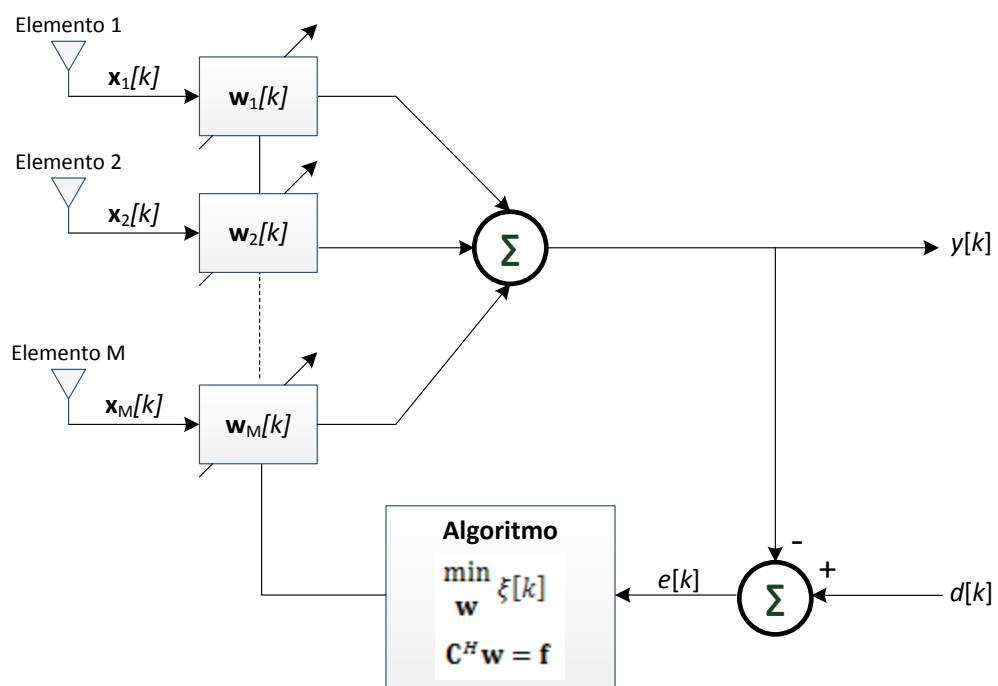


Figura 6. Estructura de *Beamforming* Adaptativo Broadband
Fuente: (Apolinario & de Campos, 2011)

El algoritmo de la Figura 6 establece que:

$$\min_{\mathbf{w}} \xi[k] \quad \text{sujeto a} \quad \mathbf{C}^H \mathbf{w} = \mathbf{f} \quad (19)$$

Donde

$$\xi[k] = E\{|e[k]|^2\} = E\{e[k]e^*[k]\} \quad (20)$$

Con la matriz de restricción \mathbf{C} de dimensión $MN \times p$, siendo p el número de restricciones y \mathbf{f} el vector de ganancia de dimensión $p \times 1$ y corresponde a la ganancia unitaria en la fuente deseada y ganancia nula para las direcciones indeseadas.

Utilizando la función de costo de la Ecuación (20) y relacionando la señal de error $e[k] = d[k] - y[k]$, se obtiene la solución de Wiener sujeto a $\mathbf{C}^H \mathbf{w} = \mathbf{f}$.

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{p} + \mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} (\mathbf{f} - \mathbf{C}^H \mathbf{R}^{-1} \mathbf{p}) \quad (21)$$

Donde

$$\mathbf{R} = E[\mathbf{x}[k] \mathbf{x}^H[k]] \quad (22)$$

$$\mathbf{p} = E[d^*[k] \mathbf{x}[k]] \quad (23)$$

\mathbf{R} es la matriz de autocorrelación de la señal de entrada y \mathbf{p} es el vector de correlación cruzada entre la señal deseada y la señal de entrada. Es fácil verificar que en ausencia de la señal deseada $d[k] = 0$ el vector $\mathbf{p} = \mathbf{0}$ y la solución óptima se reduce a la siguiente expresión:

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{f} \quad (24)$$

La solución de la Ecuación (24) es conocida como la ecuación óptima LCMV, donde la función de costo se denomina *Minimum Output Energy* (MOE). En ausencia de información estadística, se puede escoger que:

$$\min_{\mathbf{w}} \left\{ \xi[k] = \sum_{i=0}^k \lambda^{k-i} |d[i] - \mathbf{w}^H \mathbf{x}[i]|^2 \right\} \text{ sujeto a } \mathbf{C}^H \mathbf{w} = \mathbf{f} \quad (25)$$

Donde $\lambda \in (0, 1]$, lo que se aplica a la solución y se obtiene:

$$\mathbf{w}[k] = \mathbf{R}^{-1}[k]\mathbf{p}[k] + \mathbf{R}^{-1}[k]\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}[k]\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}[k]\mathbf{p}[k]) \quad (26)$$

Donde

$$\mathbf{R}[k] = \sum_{i=0}^k \lambda^{k-i} \mathbf{x}[i]\mathbf{x}^H[i] \quad (27)$$

$$\mathbf{p}[k] = \sum_{i=0}^k \lambda^{k-i} d^*[i] \mathbf{x}[i] \quad (28)$$

$\mathbf{R}[k]$ es la estimativa de la matriz de autocorrelación de la señal de entrada y $\mathbf{p}[k]$ la estimativa del vector de correlación cruzada entre la señal deseada y la señal de entrada. Las ecuaciones presentadas en esta sección pueden ser utilizadas. Sin embargo, su implementación requiere de un alto costo computacional. Los algoritmos adaptativos por lo general son más eficientes, para estimar la solución de forma recursiva.

Los algoritmos adaptativos LCMV como ya se indicó requiere de un conjunto de restricciones que deben ser cumplidas en cada interacción. Mediante la imposición de restricciones lineales sobre el filtro adaptativo, la necesidad de una señal deseada a menudo puede ser descartada. Este tipo de algoritmos adaptativos se dividen en tres familias: con Restricciones Lineales (*Linearly Constrained*), de estructura GSC (*Generalized Sidelobe Canceler*) y los de estructura *Householder* (Chandran, 2004). En el siguiente capítulo se estudian las diferentes familias de los algoritmos LCMV.

CAPÍTULO III

3. ANÁLISIS DE LOS DIFERENTES ALGORITMOS

3.1. Comportamiento de los Micrófonos

La motivación de la presente investigación es el estudio matemático de los algoritmos adaptativos que proporcionen el mejor desempeño para el realzado de la voz, para su futura implementación en un dispositivo FPGA. Debido al tiempo de la compra y entrega de los equipos, además por los costos, se decide emplear 4 micrófonos que es la cantidad que se disponen a la fecha.

Para verificar el desempeño de los diferentes algoritmos que se estudian en la presente investigación, se utiliza un arreglo ULA y como se indicó $M = 4$ esto quiere decir que el número de micrófonos empleados son cuatro. Los micrófonos se encuentran equidistados a una distancia de 0.1 m . La fuente emisora se localiza a una distancia de 1 m con relación al arreglo, como se muestra en la Figura 7. La fuente emisora emite señales del tipo AWGN (*Additive White Gaussian Noise*) que por sus características matemáticas su espectro abarca todas las frecuencias (Papoulis & Pillai , 2002).

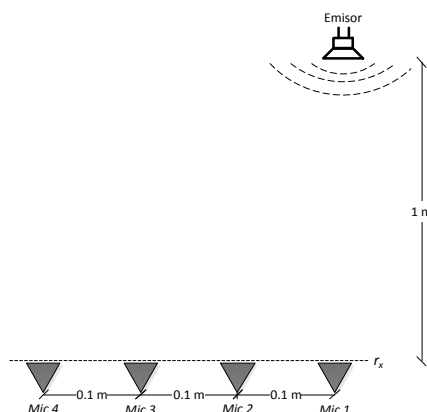


Figura 7. Arreglo ULA con $M=4$ y $d=0.1\text{ m}$

Antes de empezar el estudio de los diferentes algoritmos se analiza el comportamiento de los micrófonos para determinar si su respuesta es similar (características físicas), dado que la teoría de sensores indica que los elementos que conforman un arreglo de sensores deben ser isotrópicos.

La respuesta de los micrófonos en frecuencia se presenta en la Figura 8 y los mismos no presentan un comportamiento similar, sobre todo en altas frecuencias. Para compensar las diferencias entre los micrófonos propios de su fabricación se emplea la estructura de identificación de sistema y se utiliza como referencia el *Mic 1* (Diniz, 2013). En la Figura 9 se puede observar como la respuesta de los micrófonos en frecuencia ha cambiado y el comportamiento entre ellos es similar.

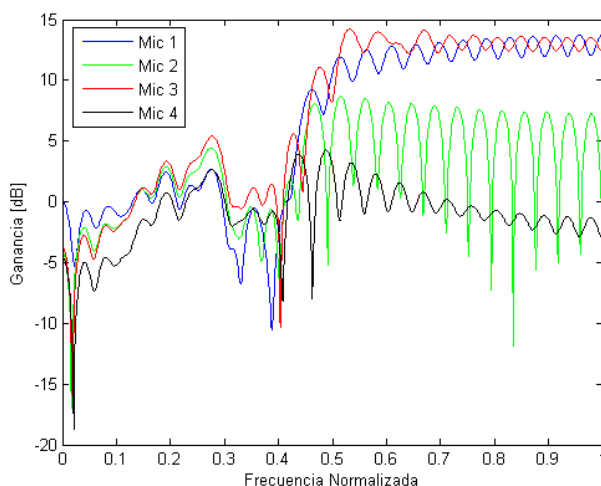


Figura 8. Respuesta de los Micrófonos en Frecuencia para un arreglo ULA

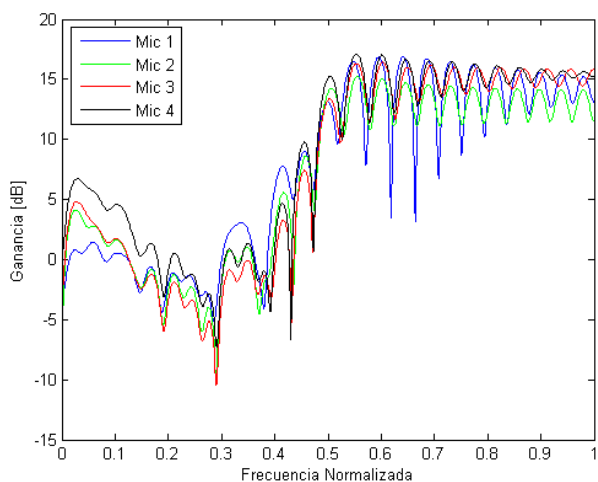


Figura 9. Respuesta Compensada de los Micrófonos en Frecuencia para un arreglo ULA

Las señales sonoras presentan retardos, como se muestra en la Figura 10, dichos retardos se producen por el hardware de procesamiento y durante la propagación de la señal, que dependiendo de la aplicación y la distancia (Tx - Rx) puede ser crítica. La estructura de predicción de señales (Diniz, 2013) permite compensar el retardo de las señales como se muestra en la Figura 11, en la cual las señales han sido corregidas el retardo.

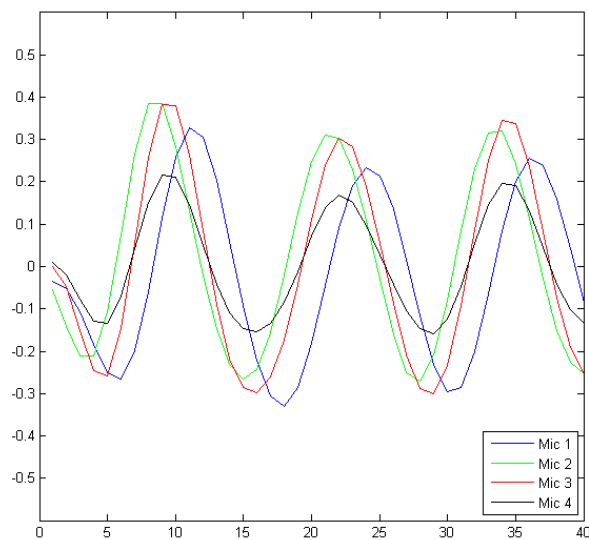


Figura 10. Señales Originales para un arreglo ULA

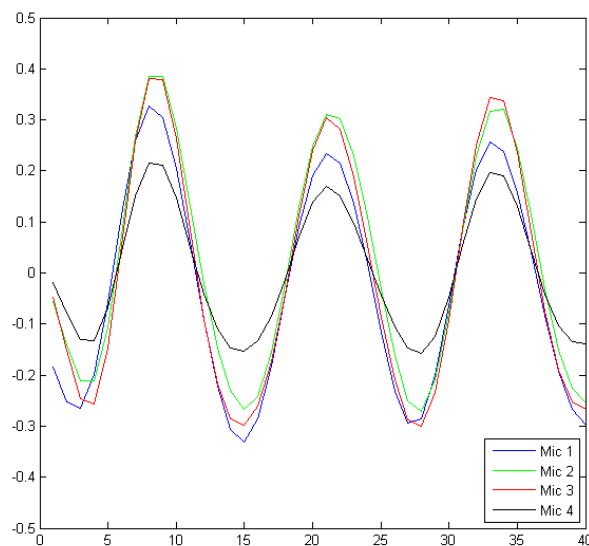


Figura 11. Señales Corregidas el Retardo para un arreglo ULA

La ecualización de canal consiste en estimar una función de transferencia para compensar la distorsión lineal causada por el canal (Diniz, 2013). Se ecualiza cada uno de los micrófonos. En la Figura 12 se muestra el resultado de la ecualización del *Mic 1*.

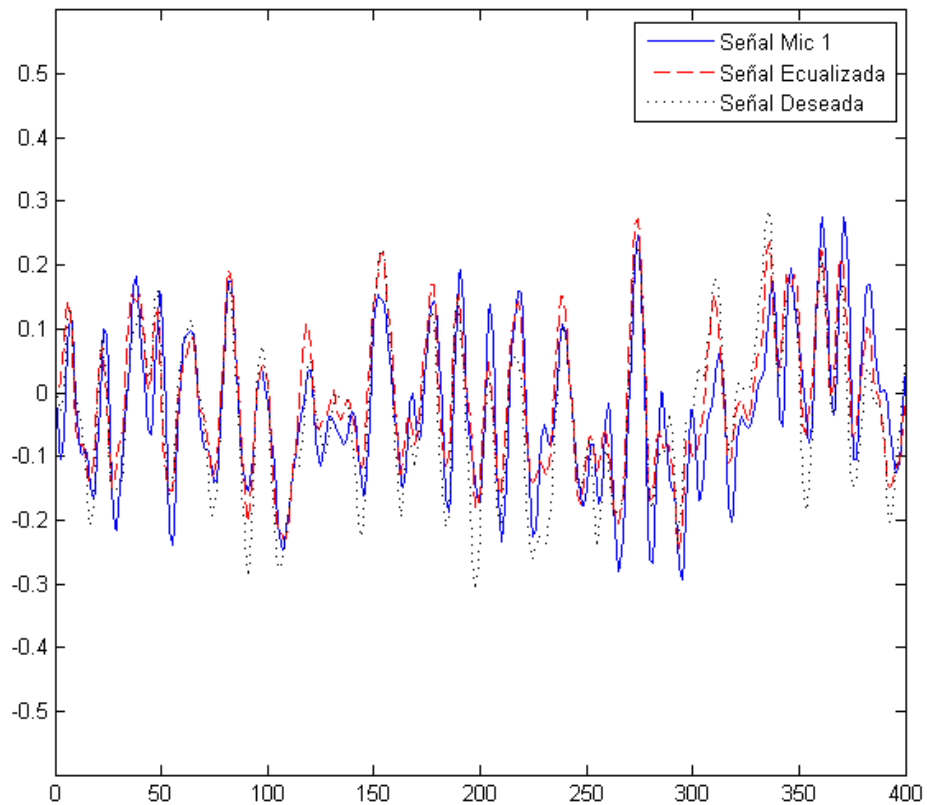


Figura 12. Resultados de la ecualización del *Mic 1* para un arreglo ULA

3.2. Estimación de la Dirección de Arribo DOA

El estudio de los algoritmos para la estimación del DOA en la presente investigación servirá como referencia para medir el desempeño de las distintas familias de los algoritmos del tipo LCMV.

3.2.1. Algoritmos para la estimación del DOA

El análisis de los algoritmos para la estimación DOA se lleva a cabo considerando que el ángulo de elevación θ es 90° , según la Figura 3, de modo que el

$\sin \theta = 1$. Esto es asumido por simplicidad, de tal manera que el problema se convierta en un problema de 1-D (Yang, Cho, & Choo, 2012). A continuación se analizan los métodos *Delay and Sum*, Capon y MUSIC.

3.2.1.1. *Delay and Sum* DAS

El método DAS se obtiene a partir de la potencia total de salida.

$$\rho_{DAS} = E\{|y[t]|^2\} = E\{|\mathbf{w}^H \mathbf{x}[k]|^2\} \quad (29)$$

$$\rho_{DAS} = E\{\mathbf{w}^H \mathbf{x}[k] \mathbf{x}^H[k] \mathbf{w}\} = \mathbf{w}^H E\{\mathbf{x}[k] \mathbf{x}^H[k]\} \mathbf{w} = \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (30)$$

Donde

$$\mathbf{R} = E\{\mathbf{x}[k] \mathbf{x}^H[k]\} = E\{(s[k] \mathbf{a}[\varphi] + \mathbf{n}[k])(s[k] \mathbf{a}^H[\varphi] + \mathbf{n}^H[k])\} \quad (31)$$

$$\mathbf{R} = \sigma_s^2 \mathbf{a}[\varphi] \mathbf{a}^H[\varphi] + \sigma_n^2 \mathbf{I}_{M \times M} \quad (32)$$

Donde $s[k]$ es la señal que arriba al m -ésimo sensor como se indicó en la Ecuación (8) pero en la Ecuación (31) $s[k]$ es su representación muestreada, $\mathbf{a}[\varphi]$ es el vector *steering* con el ángulo DOA φ , $\mathbf{n}[k]$ es el vector del ruido a la entrada del arreglo, $\sigma_s^2 = E\{s^2[k]\}$ es la potencia de la señal y $\sigma_n^2 = E\{n^2[k]\}$ es la potencia del ruido. Se supone que el arreglo recibe la señal $s[k]$ desde un ángulo azimuth φ_0 , entonces la potencia según la Ecuación (30) se maximiza para $\mathbf{w} = \mathbf{a}[\varphi_0]$, que alinea la fase de entrada $\mathbf{x}[k]$ de cada elemento del arreglo de modo que los componentes de la señal de entrada se pueden añadir de forma constructiva. Por tanto el ángulo azimuth φ_0 puede ser estimado para ser que φ sea maximizado.

$$\rho_{DAS}[\varphi] |_{\mathbf{w}=\mathbf{a}[\varphi]} = \mathbf{w}^H \mathbf{R} \mathbf{w} |_{\mathbf{w}=\mathbf{a}[\varphi]} \quad (33)$$

$$\rho_{DAS}[\varphi] = \mathbf{a}^H[\varphi] \hat{\mathbf{R}} \mathbf{a}[\varphi] \quad (34)$$

Donde la matriz de autocorrelación estimada $\hat{\mathbf{R}}$ es:

$$\hat{\mathbf{R}} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}[k] \mathbf{x}^H[k] \quad (35)$$

3.2.1.2. Capon

Con el propósito de mejorar la baja resolución del método DAS, el método de Capon modifica el *spatial spectrum* de tal manera que la contribución de señales procedentes de otras direcciones se pueda anular manteniendo la ganancia del *beamforming* en la dirección deseada. Esta modificación se puede ejecutar resolviendo el problema de minimización con restricciones:

$$\min_{\mathbf{w}} E|y[k]|^2 = \min_{\mathbf{w}} \mathbf{w}^H \mathbf{R} \mathbf{w} \quad \text{sujeto a} \quad \mathbf{w}^H \mathbf{a}[\varphi] = 1 \quad (36)$$

El método multiplicador de Lagrange se utiliza para resolver este problema y se obtiene la Ecuación (37).

$$\mathbf{w}_c = \frac{\mathbf{R}^{-1} \mathbf{a}[\varphi]}{\mathbf{a}^H[\varphi] \mathbf{R}^{-1} \mathbf{a}[\varphi]} \quad (37)$$

Sustituyendo la solución \mathbf{w}_c en la Ecuación (36) se obtiene el Capon *spatial spectrum*.

$$\rho_{\text{Capon}}[\varphi] = \frac{1}{\mathbf{a}^H[\varphi] \hat{\mathbf{R}}^{-1} \mathbf{a}[\varphi]} \quad (38)$$

3.2.1.3. MUSIC

El método de MUSIC está basado en la descomposición de la matriz de autocorrelación de la señal de entrada en autovalores. El subespacio del ruido es representado por \mathbf{E}_n y está compuesto por todos los autovalores menores y el

subespacio de la señal \mathbf{E}_s estará compuesta por los autovalores mayores. El MUSIC *spatial spectrum* se detalla en la Ecuación (39).

$$\rho_{MUSIC} = \frac{1}{\mathbf{a}^H(\theta)\mathbf{E}_n\mathbf{E}_n^H\mathbf{a}(\theta)} \quad (39)$$

3.2.2. Costo computacional de los algoritmos para la estimación del DOA

La complejidad computacional de un algoritmo no solo se determina por sí mismo, también depende de los procesadores que se utilicen. Por ejemplo, procesadores *fix-point* o procesadores *floating-point*. (Liu & Weiss, 2010). Para nuestro estudio se considera el número de multiplicaciones en cada interacción. El costo computacional de los algoritmos para la estimación del DOA se detalla en la Tabla 1, donde M es el número de micrófonos y Δ° es la escala para el barrido del ángulo. Estos costos computacionales es un aproximado y está basado en cada uno de los códigos que se utilizan para estimar el DOA en la herramienta Matlab®, dichos códigos se detallan en el Anexo D (Yang, Cho, & Choo, 2012).

Tabla 1. Costo computacional de los algoritmos para la estimación del DOA

| Algoritmo | |
|----------------------|--------------------------------------------------------------|
| Delay and Sum | $\frac{180^\circ}{\Delta^\circ}(M^4 + 2M^3 + 4M^2 + 4M + 2)$ |
| Capon | $\frac{180^\circ}{\Delta^\circ}(M^4 + 3M^3 + 4M^2 + 4M + 2)$ |
| MUSIC | $\frac{180^\circ}{\Delta^\circ}(M^4 + 2M^3 + 7M^2 + 4M + 2)$ |

Como se puede observar en la Tabla 1 el algoritmo de menor costo computacional es el DAS, seguido del algoritmo Capon que posee mayor costo computacional, debido a la inversión de la matriz de autocorrelación estimada $\hat{\mathbf{R}}^{-1}$. El algoritmo de mayor costo computacional es el algoritmo MUSIC porque en este algoritmo se debe descomponer a la matriz de autocorrelación estimada $\hat{\mathbf{R}}$ en sus autovalores.

3.2.3. Comparación de los algoritmos para la estimación del DOA

En la Figura 13 se muestra el espectro de potencia de los métodos DAS, Capon y MUSIC empleando las señales de los micrófonos con $M = 4$. Se puede determinar según dicha figura que los picos obtenidos con los tres métodos coinciden con el DOA de la señal de entrada del sistema que es de 90° . El algoritmo DAS posee las ventajas de ser simple y de bajo costo computacional como ya de detalle en la Tabla 1, pero como se puede observar tiene baja resolución, es decir que su patrón de directividad no genera un lóbulo principal con alta directividad. El método CAPON ofrece una mejor resolución que DAS pero posee un mayor costo computacional. El método MUSIC permite una alta resolución la desventaja del algoritmo MUSIC es su alto costo computacional siendo este el más alto versus a los dos métodos estudiados.

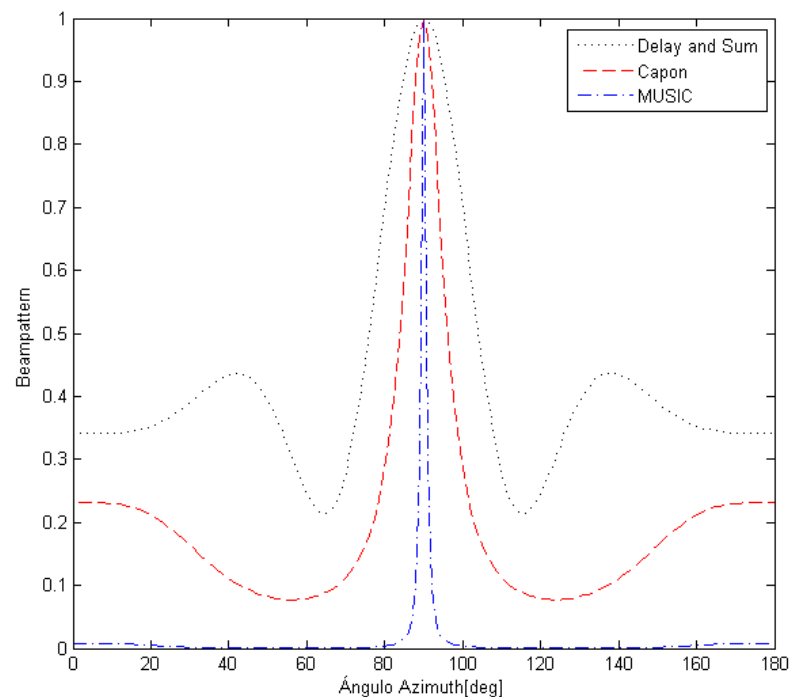


Figura 13. Comparación de los algoritmos para la estimación del DOA para un arreglo ULA

3.3. Algoritmos Adaptativos LCMV

Como se indicó en el capítulo anterior, los algoritmos LCMV se dividen en tres familias: *Linearly Constrained* tal es el caso de los CLMS (*Constrained Least Mean*

Square), entre otros. Además los algoritmos de estructura GSC que permite que cualquier algoritmo *unconstrained* pueda ser empleado como *constrained*. Por último se tiene los de estructura *Householder* que mediante la matriz \mathbf{Q} y sus reflectores cumple como el GSC pero computacionalmente son más eficiente (Apolinário J. A., 2009).

Los algoritmos adaptativos *unconstrained* que se utilizan en la presente investigación son: LMS (*Least Mean Squares*), NLMS (*Normalized Least Mean Squares*), RLS (*Recursive Least-Squares*) y CG (*Conjugate Gradient*) (Diniz, 2013).

3.3.1. Algoritmos *Linearly Constrained*

La actualización de los coeficientes en los algoritmos adaptativos *Constrained* se realiza en un subespacio que es ortogonal al subespacio generado por la matriz de restricción. La dirección de la actualización está dada por el vector de la señal de entrada premultiplicado por una matriz de proyección (de Campos, Werner, & Apolinário, September 2002). La matriz de restricción \mathbf{C} como se indicó es de dimensión $MN \times p$. Esta matriz tiene la forma siguiente:

$$\mathbf{C} = \begin{bmatrix} \overbrace{1 \quad \dots \quad 1}^M & \overbrace{0 \quad \dots \quad 0}^M & \overbrace{0 \quad \dots \quad 0}^M \\ \overbrace{0 \quad \dots \quad 0}^M & \overbrace{1 \quad \dots \quad 1}^M & \overbrace{0 \quad \dots \quad 0}^M \\ \overbrace{0 \quad \dots \quad 0}^M & \overbrace{0 \quad \dots \quad 0}^M & \overbrace{1 \quad \dots \quad 1}^M \end{bmatrix}^T \quad (40)$$

Al utilizar la matriz \mathbf{C} de la Ecuación (40) los algoritmos que se estudian en la presenta investigación poseen un desempeño bajo, puesto que el valor del MSE es alto, como se puede observar en la Figura 14, en la cual el valor promedio del MSE de cada uno de los algoritmos es de aproximadamente -7 dB , por lo que se presume que al converger los algoritmos se encuentran en un mínimo local, puesto que a pesar de que se varía el tamaño del parámetro de adaptación de los algoritmos no se obtiene una disminución en el valor del MSE. Para que los algoritmos mejoren su desempeño se preconditiona la estructura de la matriz \mathbf{C} , transformando esta matriz en una matriz *toeplitz* (Gundersen & Hakon Husoy, 2006) y como se puede observar

en la Figura 15 los algoritmos poseen un valor menor en el MSE en comparación a la Figura 14, con un valor promedio del MSE de -28 dB para CLMS, -30 dB para NLMS, -33 dB para CRLS y -35 dB para el algoritmo CCG.

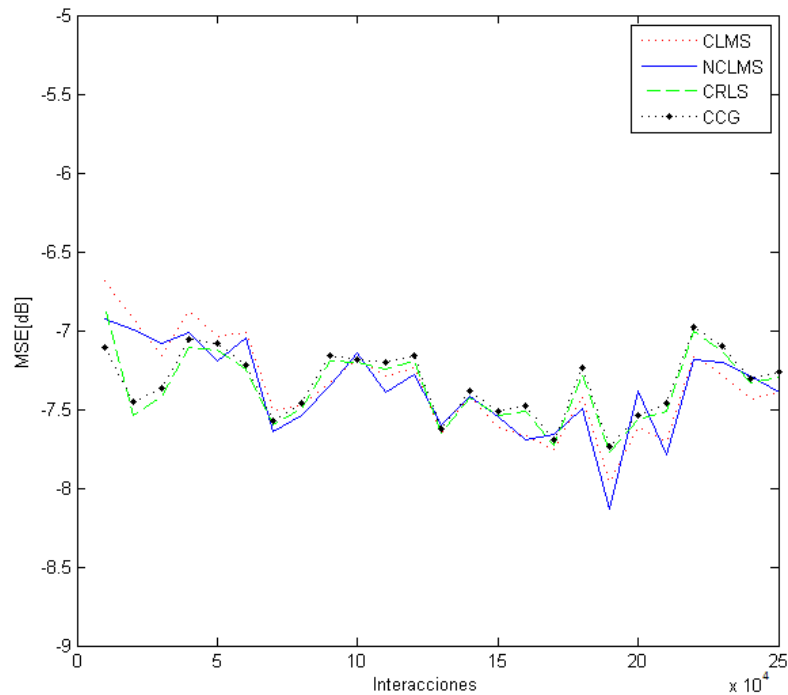


Figura 14. Algoritmos *Constrained* respecto al MSE para un arreglo ULA (C original)

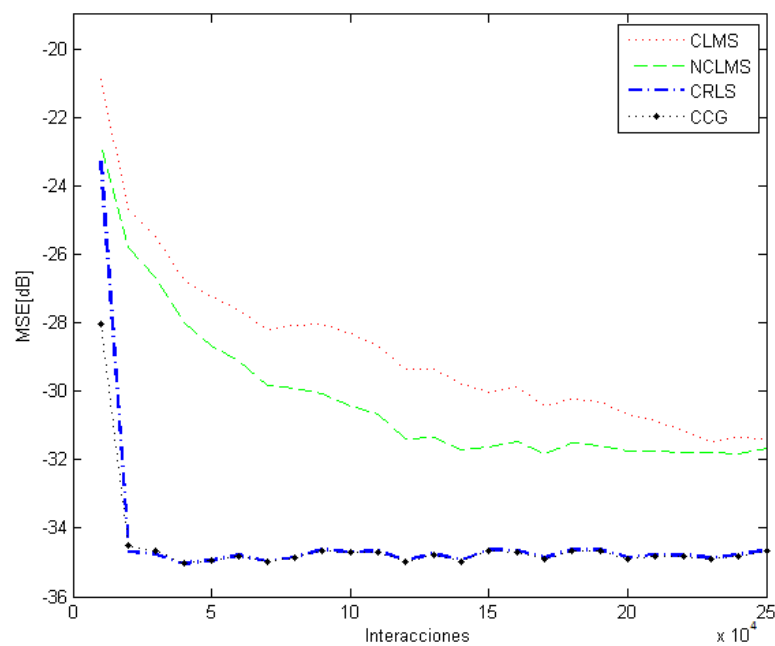


Figura 15. Algoritmos *Constrained* respecto al MSE para un arreglo ULA (C modificada)

3.3.1.1. Algoritmo *Constrained Least Mean Squares* (CLMS)

El algoritmo CLMS se ha utilizado ampliamente debido a su simplicidad y baja complejidad computacional (Chandran, 2004). La ecuación de actualización del vector de pesos para el algoritmo CLMS resuelve:

$$\min_{\mathbf{w}} |e[k]|^2 \quad \text{sujeto a} \quad \mathbf{C}^H \mathbf{w} = \mathbf{f} \quad (41)$$

Utilizando el gradiente y el método de multiplicadores de Lagrange se obtiene:

$$\mathbf{w}[k] = \mathbf{P}\{\mathbf{w}[k-1] + \mu e^*[k]\mathbf{x}[k]\} + \mathbf{F} \quad (42)$$

Donde

$$\mathbf{P} = \mathbf{I} - \mathbf{C}\{\mathbf{C}^H \mathbf{C}\}^{-1} \mathbf{C}^H \quad (43)$$

$$\mathbf{F} = \mathbf{C}\{\mathbf{C}^H \mathbf{C}\}^{-1} \mathbf{f} \quad (44)$$

La matriz \mathbf{P} es una matriz de proyección que proyecta $\mathbf{w}[k]$ sobre el hiperplano definido por $\mathbf{C}^H \mathbf{w} = 0$ y el vector \mathbf{F} trae la solución proyectada por \mathbf{P} nuevamente al hiperplano con restricciones $\mathbf{C}^H \mathbf{w} = \mathbf{f}$. Las ecuaciones del algoritmo CLMS se detallan en la Tabla 9 del Anexo A.

El algoritmo CLMS depende del parámetro μ , conocido como *step size* que controla la estabilidad y la velocidad de convergencia del algoritmo. La convergencia del algoritmo está garantizada siempre y cuando se cumpla la siguiente condición (Chandran, 2004).

$$0 < \mu < \frac{2}{3tr(\mathbf{R})} \quad (45)$$

En la Figura 16 se muestra el MSE del algoritmo CLMS cuando se varia el tamaño del *step size* en el rango de $0.006 \leq \mu \leq 0.010$. Estas variaciones se realizan con el objetivo de determinar el tamaño del *step size* que permita tener el menor valor del MSE del algoritmo. De esta variación se tiene que al aumentar el tamaño del *step size* la velocidad de convergencia aumenta pero a cambio el valor del MSE una vez que el algoritmo converge es alto en comparación con otros que emplearon pasos de adaptación de menor tamaño.

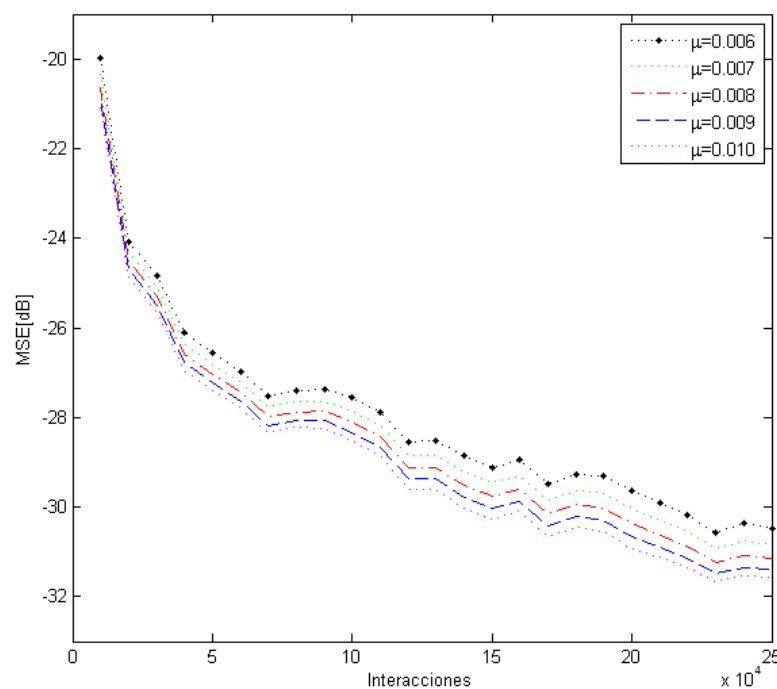


Figura 16. MSE del algoritmo CLMS con varios tamaños de μ para un arreglo ULA

3.3.1.2. Algoritmo *Normalized Constrained Least Mean Squares* (NCLMS)

La versión normalizada del algoritmo CLMS, se conoce como algoritmo NCLMS y es la normalización del algoritmo LMS. La ecuación de actualización del vector de pesos se define en la Ecuación (46).

$$\mathbf{w}[k+1] = \mathbf{P} \left\{ \mathbf{w}[k] + \mu_n \frac{e^*[k]\mathbf{x}[k]}{\mathbf{x}^H[k]\mathbf{P}\mathbf{x}[k] + \delta} \right\} + \mathbf{F} \quad (46)$$

Donde \mathbf{P} y \mathbf{F} se mostraron en la Ecuaciones (43) y (44) respectivamente. El parámetro μ_n es el *step size* que debe ser elegido entre $0 < \mu_n < 1$. El parámetro δ es un pequeño número positivo para evitar la división por cero en las primeras interacciones. Las ecuaciones del algoritmo NCLMS se detallan en la Tabla 10 del Anexo A.

La Figura 17 muestra el MSE del algoritmo NCLMS cuando se varía el tamaño del *step size* en el rango de $0.10 \leq \mu_n \leq 0.50$, de estas variaciones se puede analizar que a medida que el tamaño del *step size* aumenta la velocidad de convergencia es mayor. Para $\mu_n \geq 0.40$ el algoritmo ya convergió como se puede corroborar para $\mu_n = 0.50$, pero a cambio el valor del MSE una vez que el algoritmo converge es alto en comparación con otros que emplearon pasos de adaptación de menor tamaño.

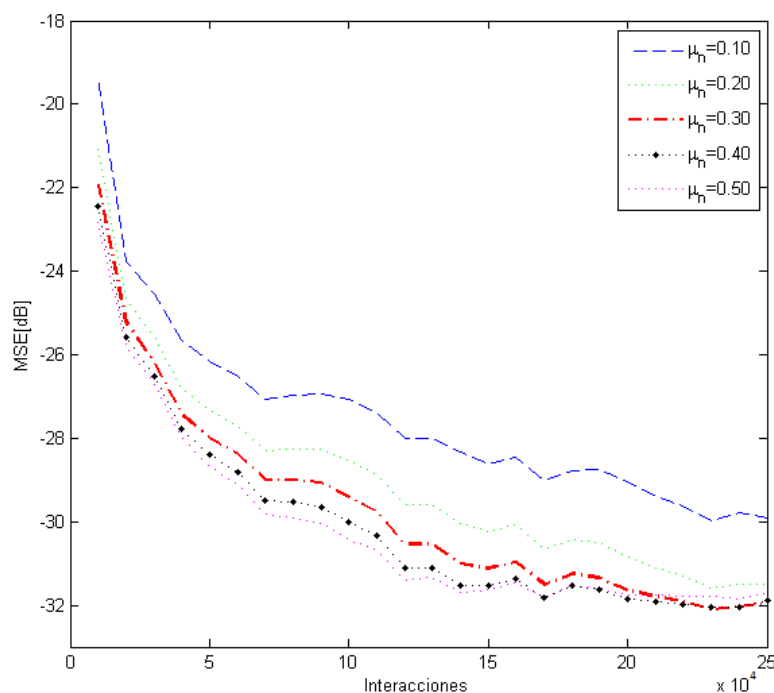


Figura 17. MSE del algoritmo NCLMS con varios tamaños de μ_n para un arreglo ULA

3.3.1.3. Algoritmo *Constrained Recursive Least-Squares* (CRLS)

El algoritmo CRLS parte de la Ecuación (26) que es la aproximación a la solución óptima de los filtros LCMV, esta ecuación se divide en dos términos.

$$\mathbf{w}[k] = \mathbf{w}_{uc}[k] + \mathbf{w}_c[k] \quad (47)$$

Donde

$$\mathbf{w}_{uc}[k] = \mathbf{R}^{-1}[k]\mathbf{p}[k] \quad (48)$$

$$\mathbf{w}_c[k] = \mathbf{R}^{-1}[k]\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}[k]\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}[k]\mathbf{p}[k]) \quad (49)$$

El vector de actualización de pesos $\mathbf{w}_{uc}[k]$ es la solución *unconstrained* y $\mathbf{w}_c[k]$ depende de la restricción impuesta por $\mathbf{C}^H\mathbf{w} = \mathbf{f}$. El vector de pesos $\mathbf{w}_{uc}[k]$ posee una expresión recursiva *unconstrained* propuesta por el algoritmo RLS que se presenta en la Ecuación (50).

$$\mathbf{w}_{uc}[k] = \mathbf{w}_{uc}[k-1] + e_{uc}^*[k]\boldsymbol{\kappa}[k] \quad (50)$$

Donde $\boldsymbol{\kappa}[k] = \mathbf{R}^{-1}[k]\mathbf{x}[k]$ es el vector de ganancia y $e_{uc}[k] = d[k] - \mathbf{w}_{uc}^H[k-1]\mathbf{x}[k]$ es el error *a priori unconstrained*. Para obtener el vector de pesos recursivo $\mathbf{w}_c[k]$ se define dos matrices auxiliares $\boldsymbol{\Gamma}[k]$ y $\boldsymbol{\Psi}[k]$ de dimensiones $(MN \times p)$ y $(p \times p)$ respectivamente.

$$\boldsymbol{\Gamma}[k] = \mathbf{R}^{-1}[k]\mathbf{C} \quad (51)$$

$$\boldsymbol{\Psi}[k] = \mathbf{C}^H\boldsymbol{\Gamma}[k] = \mathbf{C}^H\mathbf{R}^{-1}[k]\mathbf{C} \quad (52)$$

Por tanto, se obtiene:

$$\mathbf{w}_c[k] = \boldsymbol{\Gamma}[k]\boldsymbol{\Psi}^{-1}[k]\{\mathbf{f} - \mathbf{C}^H\mathbf{w}_{uc}[k]\} \quad (53)$$

La principal ventaja del algoritmo CRLS es su rápida convergencia independientemente de la matriz de autocorrelación de la señal de entrada, pero el algoritmo presenta problemas de estabilidad y alto costo computacional por la

inversión de la matriz $\mathbf{R}^{-1}[k]$. La velocidad de convergencia y estabilidad del algoritmo CRLS depende del *forgetting factor* λ que es un factor de ponderación exponencial. El tamaño de este factor debe encontrarse entre $0 < \lambda < 1$. Las ecuaciones del algoritmo CRLS se detallan en la Tabla 11 del Anexo A.

En la Figura 18 se presenta el MSE del algoritmo CRLS con variaciones del *forgetting factor* en el rango de $0.9991 \leq \lambda \leq 1$, de esas variaciones se puede analizar que el algoritmo presenta bruscos cambios a medida que se tienen pequeñas variaciones en el tamaño del *forgetting factor* y al aumentar el tamaño de este *forgetting factor* la velocidad de convergencia empieza a disminuir, es así que para $\lambda = 1$ existe un aumento del valor del MSE de aproximadamente 4 dB en comparación a los otros MSE obtenidos cuando el algoritmo ya convergió.

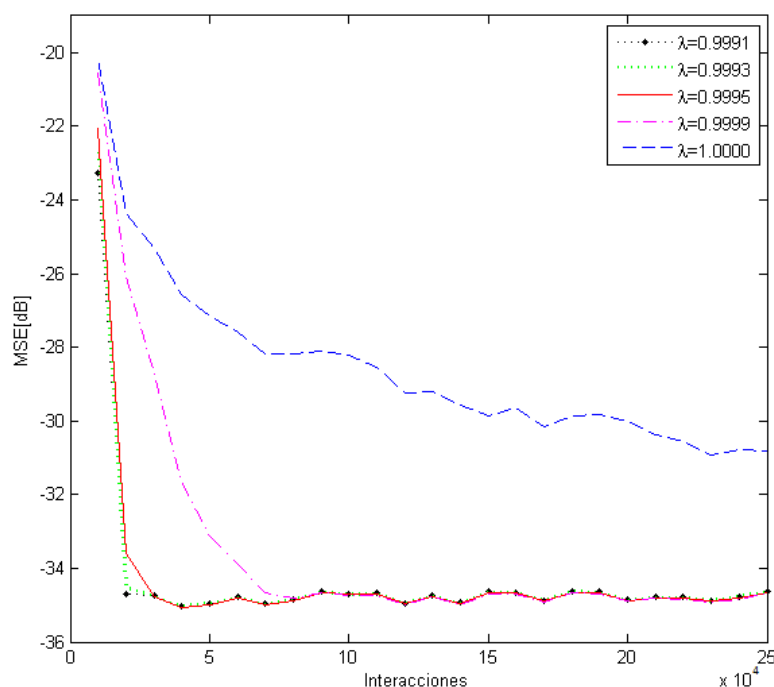


Figura 18. MSE del algoritmo CRLS con varios tamaños de λ para un arreglo ULA

3.3.1.4. Algoritmo *Constrained Conjugate Gradient* (CCG)

El algoritmo *Constrained Conjugate Gradient* se obtiene mediante una comparación con la estructura GSC (Apolinário, de Campos, & Bernal, 2000). El

comportamiento de este algoritmo depende de dos parámetros de adaptación: λ_f y η . El valor de η debe estar definido entre $\lambda_f - 0.5 \leq \eta \leq \lambda_f$. Las ecuaciones del algoritmo CCG se detallan en la Tabla 12 del Anexo A.

La Figura 19 presenta el MSE del algoritmo CCG con variaciones del λ_f en el rango de $0.9991 \leq \lambda_f \leq 1$ y con un valor fijo de $\eta = 0.70$. De las variaciones de λ_f se puede analizar que a medida que λ_f aumenta en variaciones pequeñas ($\lambda_f = 0.9991$, $\lambda_f = 0.9993$ y $\lambda_f = 0.9997$) la velocidad de convergencia es mayor pero para $\lambda_f \geq 0.9999$ el algoritmo empieza a disminuir la velocidad de convergencia y esto se puede corroborar para $\lambda_f = 1$, en el cual el algoritmo necesita más interacciones para converger.

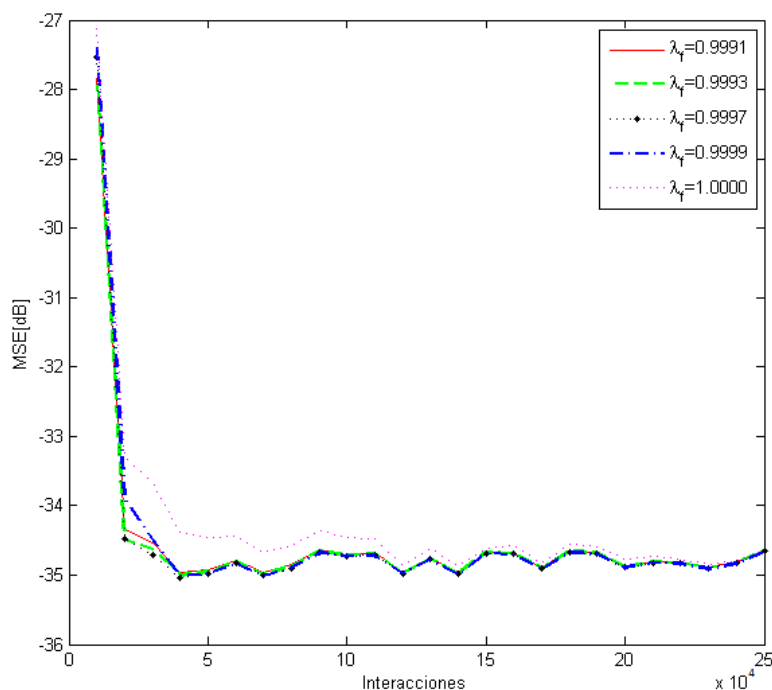


Figura 19. MSE del algoritmo CCG con varios tamaños de λ_f para un arreglo ULA

Según se analizó en la Figura 19, el valor de λ_f que presenta un mejor desempeño para el algoritmo CCG es 0.9993, si dicho tamaño se conserva fijo y se empieza a variar el tamaño de η en el rango de $0.50 \leq \eta \leq 0.90$ se obtiene la Figura 20. A medida que η varía no se obtiene cambios bruscos en la velocidad de convergencia,

por lo cual se realiza un zoom en la región más crítica de convergencia del algoritmo, entonces se tiene que al aumentar el tamaño de η aumenta la velocidad de convergencia en pequeños valores y similares valores del MSE.

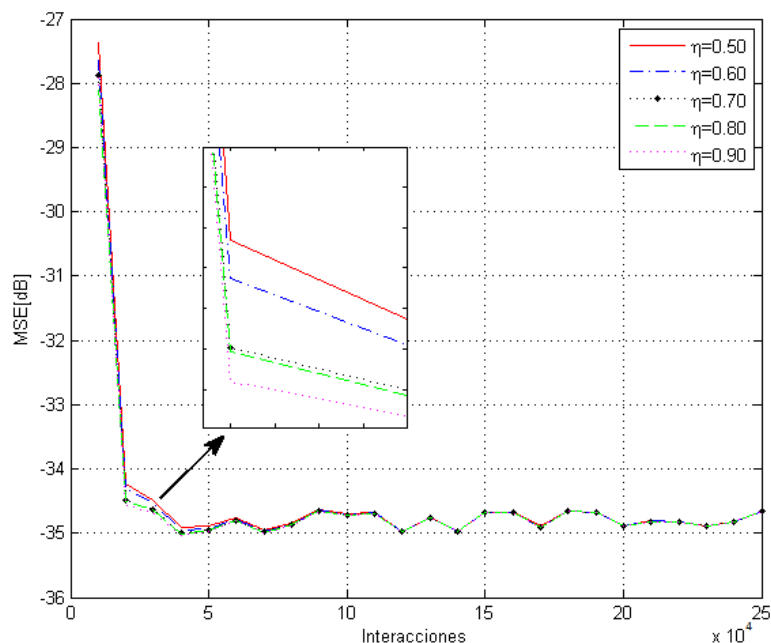


Figura 20. MSE del algoritmo CCG con varios tamaños de η para un arreglo ULA

3.3.2. Estructura *Generalized Sidelobe Canceler* (GSC)

En la Figura 21 se presenta la estructura GSC en diagrama. La estructura GSC es una alternativa diferente para implementar filtros adaptativos *linearly constrained*. En la estructura GSC, la dimensión del subespacio de adaptación se reduce adecuadamente a través de una matriz de *blocking* \mathbf{B} , permitiendo que cualquier algoritmo *unconstrained* pueda ser empleado como *constrained* (Werner, Apolinário, & Marcello L. R.).

Sea \mathbf{B} la matriz *blocking* de dimensión $MN \times (MN - p)$ que abarca el espacio nulo de la matriz de restricción \mathbf{C} . Existe una limitación para la matriz *blocking* y es que sus columnas deben ser ortogonales a la matriz de restricciones. (Chandran, 2004).

$$\mathbf{B}^H \mathbf{C} = 0 \quad (54)$$

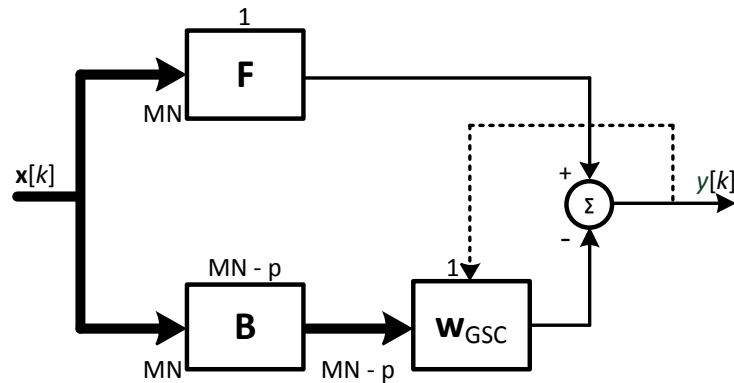


Figura 21. Estructura *Generalized Sidelobe Canceling* (GSC)
Fuente: (de Campos, Werner, & Apolinário, September 2002)

Una matriz de transformación \mathbf{T} de longitud $MN \times MN$, relaciona la estructura GSC con el filtro LCMV.

$$\mathbf{T} = [\mathbf{C} \ ; \ \mathbf{B}] \quad (55)$$

Se supone que un vector de coeficientes transformados $\bar{\mathbf{w}}[k]$ se relaciona con el vector de coeficientes $\mathbf{w}[k]$ de la estructura LCMV.

$$\mathbf{w}[k] = \mathbf{T} \bar{\mathbf{w}}[k] \quad (56)$$

La transformación del vector de pesos no modifica el error de salida, siempre que \mathbf{T} sea invertible, lo cual está garantizado a partir de la Ecuación (54). Si dividimos el vector $\bar{\mathbf{w}}[k]$ se obtiene:

$$\bar{\mathbf{w}}[k] = \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} \quad (57)$$

Donde los vectores $\bar{\mathbf{w}}_U$ y $\bar{\mathbf{w}}_L$ son de dimensión $p \times 1$ y $(MN - p) \times 1$, respectivamente. Se puede demostrar la actualización del vector del vector de pesos en la Ecuación (58).

$$\mathbf{w}[k] = \mathbf{F} - \mathbf{B}\bar{\mathbf{w}}_L \quad (58)$$

y

$$\bar{\mathbf{w}}_U = (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{f} \quad (59)$$

$$-\bar{\mathbf{w}}_L = \mathbf{w}_{GSC}[k] \quad (60)$$

Donde $\mathbf{C}\bar{\mathbf{w}}_U$ corresponde a $\mathbf{F} = \mathbf{C}\{\mathbf{C}^H \mathbf{C}\}^{-1} \mathbf{f}$. El vector \mathbf{w}_{GSC} no se ve afectado por las restricciones y se puede adaptar utilizando algoritmos adaptativos sin restricciones. Según la Figura 21, se puede deducir las siguientes ecuaciones:

$$y[k] = \mathbf{F}^H \mathbf{x}[k] - \mathbf{w}^H \mathbf{x}_{GSC} \quad (61)$$

Donde

$$d_{GSC}[k] = \mathbf{F}^H \mathbf{x}[k] \quad (62)$$

$$\mathbf{x}_{GSC} = \mathbf{B}^H \mathbf{x}[k] \quad (63)$$

La matriz de *blocking* tiene un efecto directo con la complejidad computacional, por su multiplicación con el vector de la señal de entrada en cada iteración. En el Anexo B se encuentran detallados los algoritmos GSC-LMS, GSC-NLMS, GSC-RLS y GSC-CG en las Tablas 13, 14, 15 y 16 respectivamente.

En la Figura 22 se muestra el MSE del algoritmo GSC-LMS cuando se varía el tamaño del *step size* en el rango de $0.006 \leq \mu \leq 0.010$, tal como se realizó en la estructura CLMS mostrando los dos algoritmos un similar desempeño en el valor del MSE y la velocidad de convergencia, salvo una pequeña variación para $\mu = 0.010$ atribuible a la sensibilidad de la estructura y una posible alteración del canal en el momento de la adquisición de la señal.

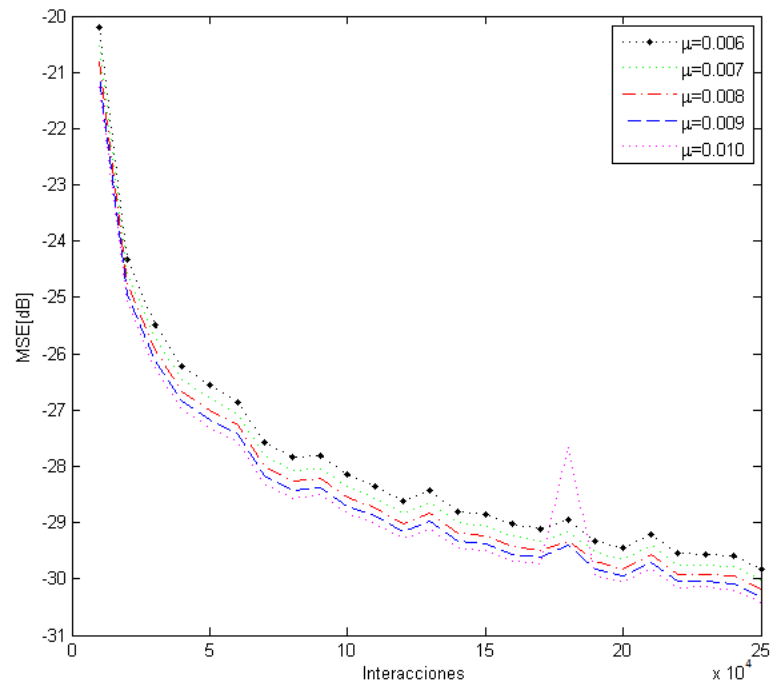


Figura 22. MSE del algoritmo GSC-LMS con varios tamaños de μ para un arreglo ULA

La Figura 23 muestra el MSE del algoritmo GSC-NLMS cuando se varía el tamaño del *step size* en el rango de $0.10 \leq \mu_n \leq 0.50$, tal como se realizó en la estructura NCLMS, mostrando los dos algoritmos un similar desempeño en el valor del MSE y la velocidad de convergencia.

En la Figura 24 se presenta el MSE del algoritmo GSC-RLS con variaciones del *forgetting factor* en el rango de $0.9991 \leq \lambda \leq 1$, de esas variaciones se puede analizar que el algoritmo presenta bruscos cambios a medida que se tienen pequeñas variaciones en el tamaño del *forgetting factor*. Al aumentar el tamaño de este *forgetting factor* la velocidad de convergencia empieza a disminuir, es así que para $\lambda = 1$ existe un aumento del MSE de aproximadamente 2 dB en comparación a los otros tamaños del *forgetting factor* utilizados en los cuales el algoritmo ya convergió.

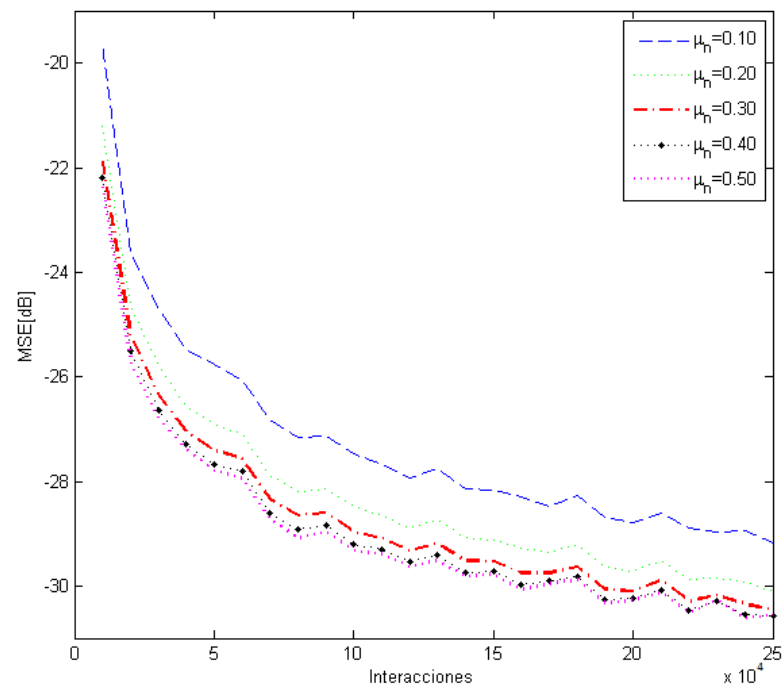


Figura 23. MSE del algoritmo GSC-NLMS con varios tamaños de μ_n para un arreglo ULA

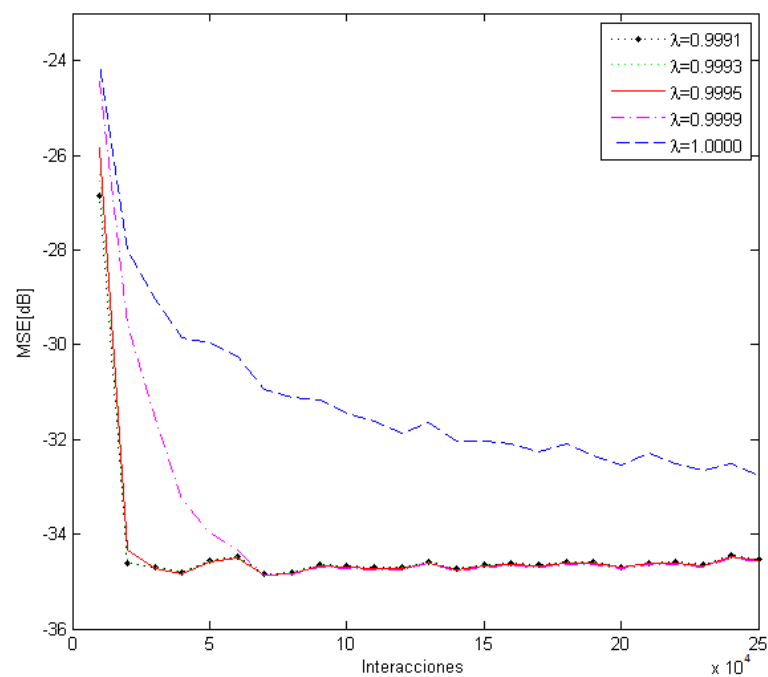


Figura 24. MSE del algoritmo GSC-RLS con varios tamaños de λ para un arreglo ULA

La Figura 25 presenta el MSE del algoritmo GSC-CG con variaciones del λ_f en el rango de $0.9991 \leq \lambda_f \leq 1$ y con un valor fijo de $\eta = 0.70$. De las variaciones de

λ_f se puede analizar que a medida que λ_f aumenta en variaciones pequeñas ($\lambda_f = 0.9991$, $\lambda_f = 0.9993$ y $\lambda_f = 0.9997$) la velocidad de convergencia es mayor pero para $\lambda_f \geq 0.9999$ el algoritmo empieza a disminuir la velocidad de convergencia y esto se puede corroborar para $\lambda_f = 1$, en el cual el algoritmo necesita más interacciones para converger.

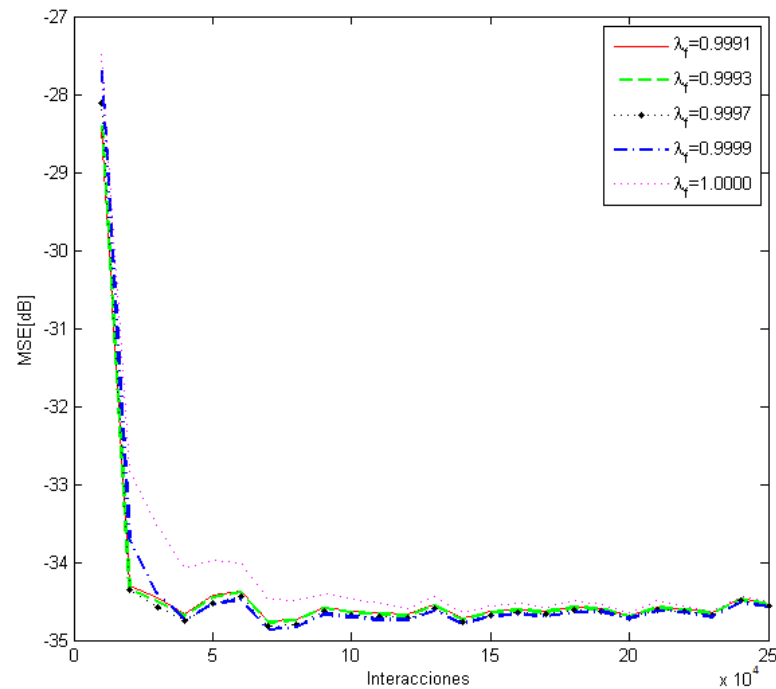


Figura 25. MSE del algoritmo GSC-CG con varios tamaños de λ_f para un arreglo ULA

Según se analizó en la Figura 25, el valor de λ_f que presenta un mejor desempeño para el algoritmo GSC-CG es 0.9991, si dicho tamaño se conserva fijo y se empieza a variar el tamaño de η en el rango de $0.50 \leq \eta \leq 0.90$ se obtiene la Figura 26. A medida que η varía no se obtiene cambios bruscos en la velocidad de convergencia, por lo cual se realiza un zoom en la región más crítica de convergencia del algoritmo, entonces se tiene que al aumentar el tamaño de η aumenta la velocidad de convergencia en pequeños valores y similares valores del MSE.

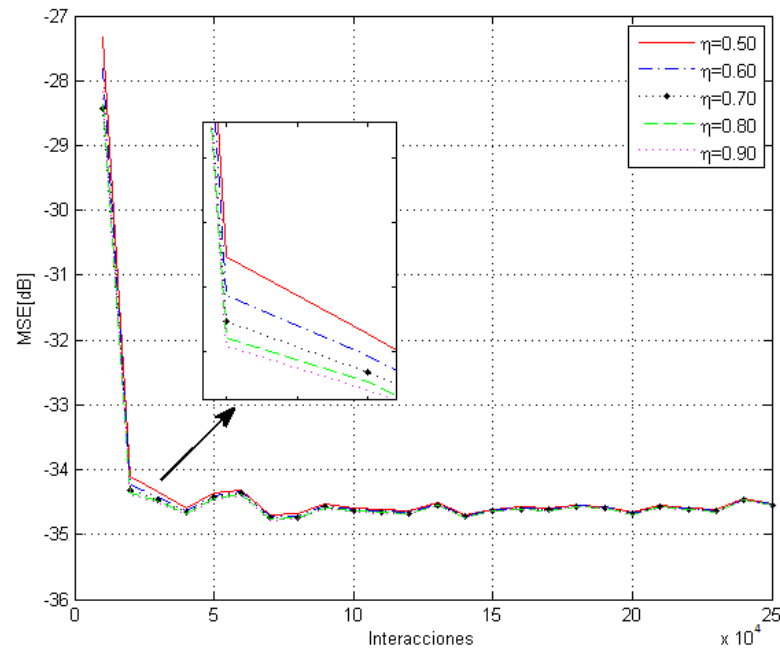


Figura 26. MSE del algoritmo GSC-CG con varios tamaños de η para un arreglo ULA

3.3.3. Transformada de *Householder*

La multiplicación de la matriz de *blocking* por el vector de señal de entrada en la estructura GSC puede ser computacionalmente alta y para muchas aplicaciones, no es práctico. La estructura *Householder* es una solución a este problema mediante la matriz \mathbf{Q} y sus reflectores de *Householder* que cumple como la estructura GSC, pero computacionalmente son más eficientes (de Campos, Werner, & Apolinário, September 2002).

La obtención del algoritmo HCLMS se inicia con la rotación en el vector $\mathbf{P}\mathbf{x}[k]$ con el fin de asegurar que el vector de coeficientes no se perturbe en otra dirección no excitado por $\mathbf{P}\mathbf{x}[k]$. Esto es posible si una matriz de rotación ortogonal \mathbf{Q} se utiliza como la transformación que va a generar un vector de coeficientes modificados $\bar{\mathbf{w}}[k]$ que se relaciona a $\mathbf{w}[k]$ (de Campos, Werner, & Apolinário, September 2002).

$$\bar{\mathbf{w}}[k] = \mathbf{Q}\mathbf{w}[k] \quad (64)$$

La matriz \mathbf{Q} se debe escoger de tal manera que $\mathbf{Q}\mathbf{Q}^H = \mathbf{I}$.

$$\bar{\mathbf{C}}\{\bar{\mathbf{C}}^H\bar{\mathbf{C}}\}^{-1}\bar{\mathbf{C}}^H = \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (65)$$

Entonces $\bar{\mathbf{C}} = \mathbf{Q}\mathbf{C}$ satisfice $\mathbf{f} = \bar{\mathbf{C}}^H\bar{\mathbf{w}}[n + 1]$ y la transformada de la matriz de proyección está dada por:

$$\bar{\mathbf{P}} = \mathbf{Q}\mathbf{P}\mathbf{Q}^H = \mathbf{I} - \bar{\mathbf{C}}\{\bar{\mathbf{C}}^H\bar{\mathbf{C}}\}^{-1}\bar{\mathbf{C}}^H = \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (66)$$

Si $\bar{\mathbf{w}}[0]$ se inicializa según.

$$\bar{\mathbf{w}}[0] = \bar{\mathbf{C}}\{\bar{\mathbf{C}}^H\bar{\mathbf{C}}\}^{-1}\mathbf{f} = \mathbf{Q}\mathbf{F} \quad (67)$$

Los primeros p elementos de $\bar{\mathbf{w}}[0]$ no necesitan ser actualizados. La actualización del algoritmo HCLMS se obtiene premultiplicado el algoritmo CLMS por \mathbf{Q} .

$$\bar{\mathbf{w}}[k + 1] = \mathbf{Q}\mathbf{w}[k + 1] \quad (68)$$

$$\bar{\mathbf{w}}[k + 1] = \mathbf{Q}\{\mathbf{P}(\mathbf{w}[k] - \mu y[k]\mathbf{x}[k]) + \mathbf{F}\} \quad (69)$$

$$\bar{\mathbf{w}}[k + 1] = \{\mathbf{Q}\mathbf{P}\mathbf{Q}^H\}\{\mathbf{Q}\mathbf{w}[k]\} - \mu y[k]\{\mathbf{Q}\mathbf{P}\mathbf{Q}^H\}\{\mathbf{Q}\mathbf{x}[k]\} + \mathbf{Q}\mathbf{F} \quad (70)$$

$$\bar{\mathbf{w}}[k + 1] = \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\mathbf{w}}[k] - \mu y[k] \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\mathbf{x}}[k] + \begin{bmatrix} \bar{\mathbf{w}}_U[0] \\ \mathbf{0} \end{bmatrix} \quad (71)$$

$$\bar{\mathbf{w}}[k + 1] = \begin{bmatrix} \bar{\mathbf{w}}_U[0] \\ \bar{\mathbf{w}}_L[k + 1] \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{w}}_U[0] \\ \bar{\mathbf{w}}_L[k] \end{bmatrix} - \mu y[k] \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{x}}_L[k] \end{bmatrix} \quad (72)$$

Donde $\bar{\mathbf{w}}_L[k]$ y $\bar{\mathbf{x}}_L[k]$ denota los últimos $MN - p$ elementos de los vectores $\bar{\mathbf{w}}[k]$ y $\bar{\mathbf{x}}[k]$ respectivamente. El vector $\bar{\mathbf{C}}\{\bar{\mathbf{C}}^T\bar{\mathbf{C}}\}^{-1}\mathbf{f}$ tiene solo los p primero

elementos no ceros. La solución $\bar{\mathbf{w}}[k]$ es basada en la transformación \mathbf{Q} a la señal de salida y como consecuencia el error de salida no se modifica por la transformación. Por lo tanto el algoritmo propuesto minimiza la misma función objetivo minimizada por el algoritmo CLMS. Los algoritmos HCLMS, HNCLMS, HCRLS y HCCG se encuentran detallados en las Tablas 17, 18, 19 y 20 respectivamente en el Anexo C.

En la Figura 27 se muestra el MSE del algoritmo HCLMS cuando se varía el tamaño del *step size* en el rango de $0.006 \leq \mu \leq 0.010$, tal como se realizó en la estructura GSC-LMS mostrando los dos algoritmos un similar desempeño en el valor del MSE y la velocidad de convergencia.

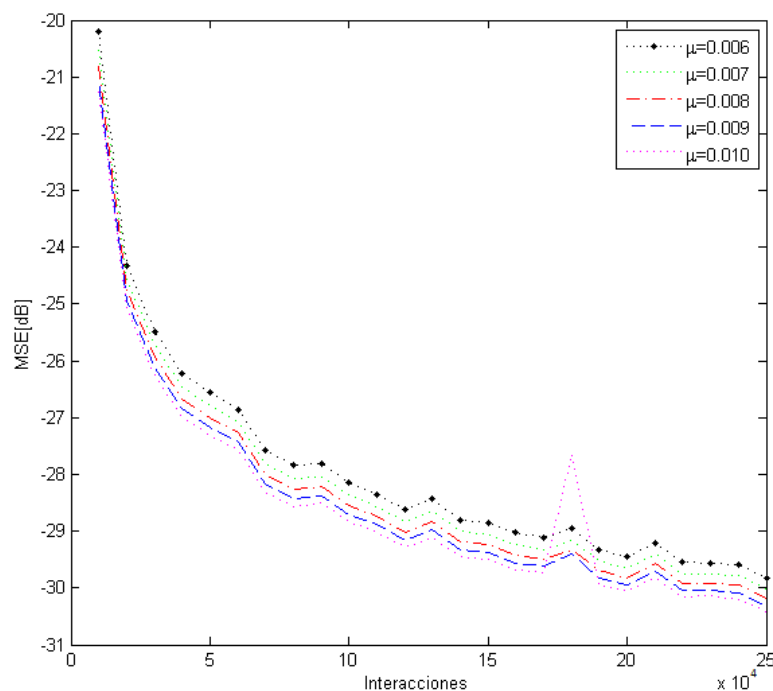


Figura 27. MSE del algoritmo HCLMS con varios tamaños de μ para un arreglo ULA

La Figura 28 muestra el MSE del algoritmo HNCLMS cuando se varía el tamaño del *step size* en el rango de $0.10 \leq \mu_n \leq 0.50$, tal como se realizó en la estructura NCLMS y GSC-NLMS mostrando los tres algoritmos un similar desempeño en el valor del MSE y la velocidad de convergencia.

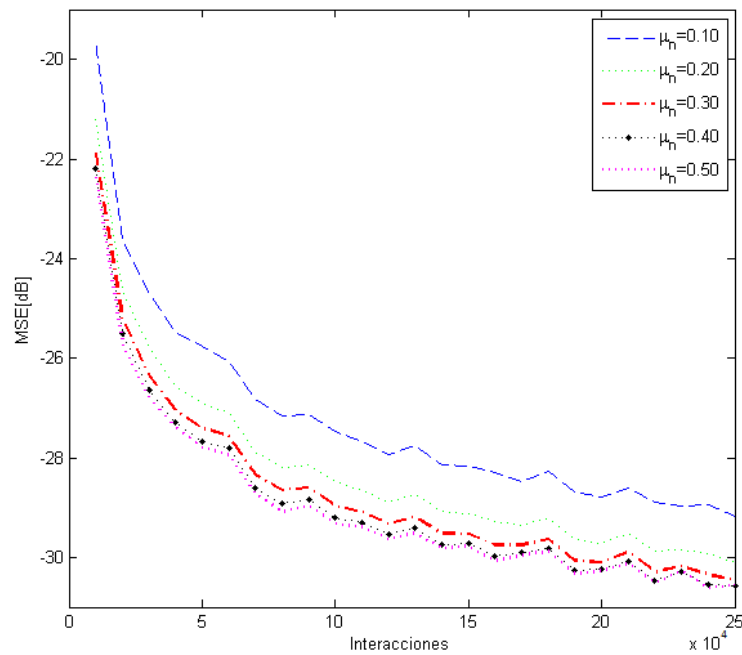


Figura 28. MSE del algoritmo HNCLMS con varios tamaños de μ_n para un arreglo ULA

En la Figura 29 se presenta el MSE del algoritmo HCRLS con variaciones del *forgetting factor* en el rango de $0.9991 \leq \lambda \leq 1$, tal como se realizó en la estructura GSC-RLS mostrando los tres algoritmos un similar desempeño en el valor del MSE y la velocidad de convergencia.

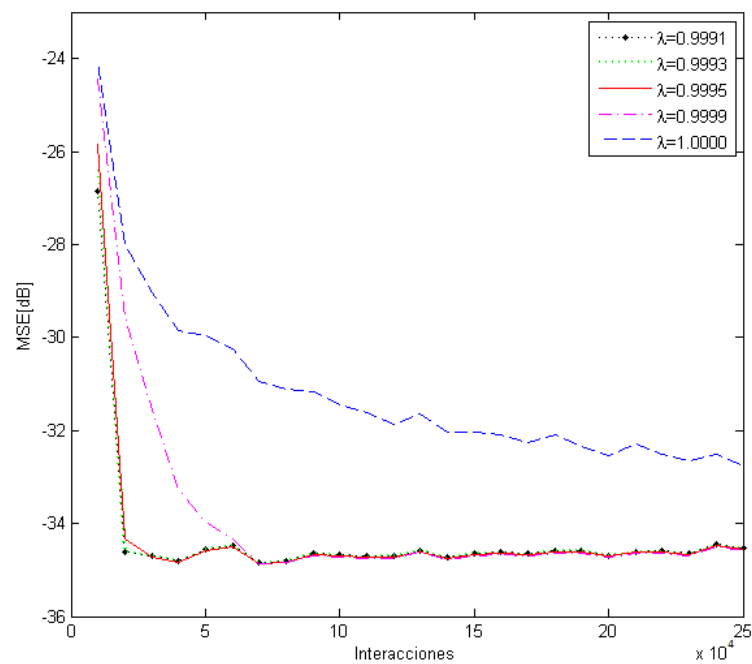


Figura 29. MSE del algoritmo HCRLS con varios tamaños de λ para un arreglo ULA

La Figura 30 presenta el MSE del algoritmo HCCG con variaciones del λ_f en el rango de $0.9991 \leq \lambda_f \leq 1$ y con un valor fijo de $\eta = 0.70$. Se puede analizar de estas variaciones de λ_f que a medida que el parámetro de adaptación aumenta en variaciones pequeñas la velocidad de convergencia aumenta en pequeñas variaciones.

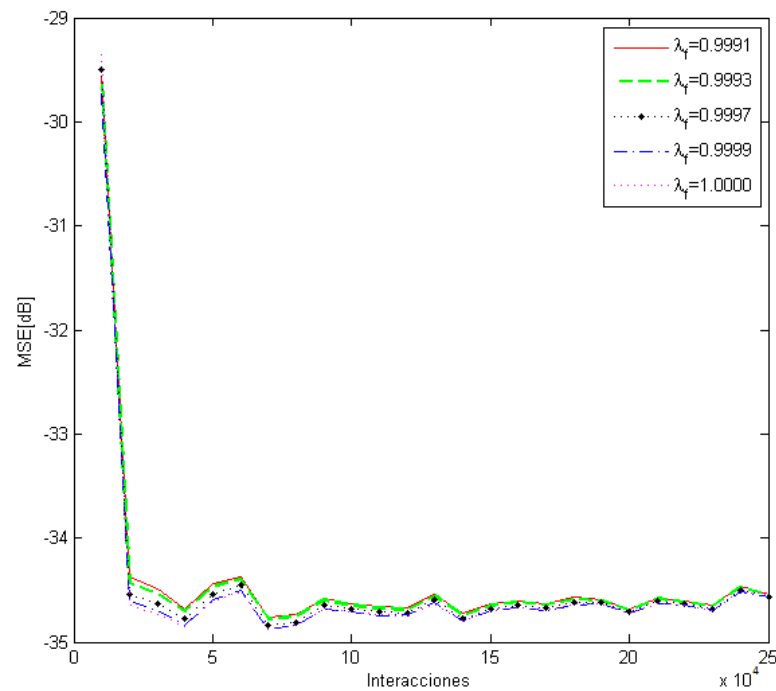


Figura 30. MSE del algoritmo HCCG con varios valores de λ_f para un arreglo ULA

Según se analizó en la Figura 30, el valor de λ_f que presenta un mejor desempeño para el algoritmo HCCG es 0.9999, si se conserva fijo este tamaño y se empieza a variar el tamaño de η en el rango de $0.50 \leq \eta \leq 0.90$ se obtiene la Figura 31. A medida que η aumenta se obtiene pequeños cambios la velocidad de convergencia, por lo cual se realiza un zoom en la región más crítica de convergencia del algoritmo, entonces se tiene que al aumentar el tamaño de η aumenta la velocidad de convergencia en pequeños valores y similares valores del MSE.

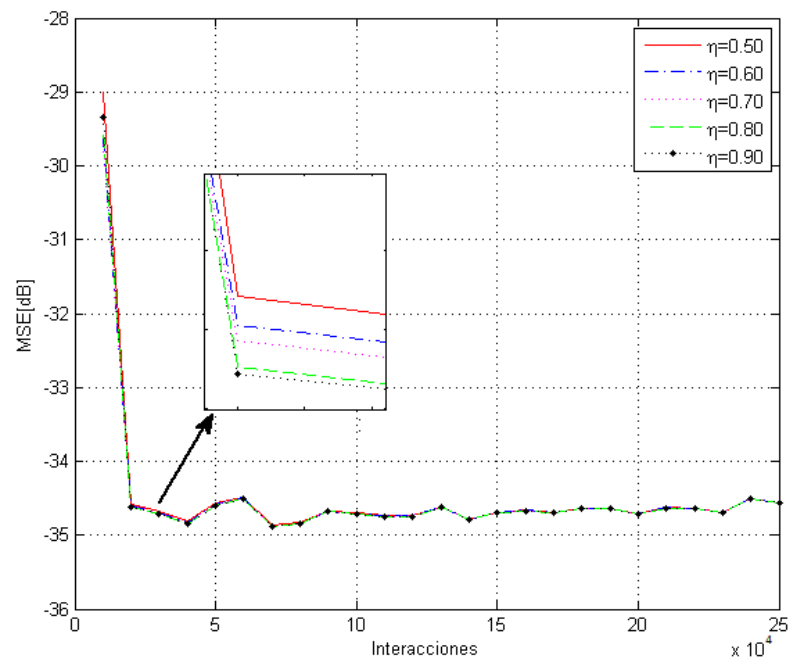


Figura 31. MSE del algoritmo HCCG con varios valores de η para un arreglo ULA

En la Tabla 2 se presenta un resumen del MSE del algoritmo LMS para las estructuras *Constrained*, *GSC* y *Householder* versus los diferentes *step size* utilizados, las respuestas marcadas con negrilla representan el mejor desempeño del algoritmo para cada una de las tres estructuras.

Tabla 2. MSE en dB del algoritmo LMS con μ variable

| Algoritmo LMS | | | | | |
|---------------|----------|----------|-----------------|-----------------|----------|
| μ | 0.006 | 0.007 | 0.008 | 0.009 | 0.010 |
| CLMS | -27.1185 | -27.4273 | -27.6910 | -27.9188 | -28.1162 |
| GSC-LMS | -27.1859 | -27.4300 | -27.6321 | -27.8002 | -27.8807 |
| HCLMS | -27.1859 | -27.4300 | -27.6321 | -27.8002 | -27.8807 |

Un resumen del MSE del algoritmo NLMS para las estructuras *Constrained*, *GSC* y *Householder* versus los diferentes *step size* utilizados se muestra en la Tabla 3. Las respuestas marcadas con negrilla representan el mejor desempeño del algoritmo para cada una de las estructuras.

Tabla 3. MSE en dB del algoritmo NLMS con μ_n variable

| Algoritmo NLMS | | | | | |
|----------------|----------|----------|----------|----------|-----------------|
| μ_n | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 |
| NCLMS | -26.6242 | -28.0734 | -28.8170 | -29.2204 | -29.4207 |
| GSC-NLMS | -26.5092 | -27.6192 | -28.1109 | -28.3568 | -28.4688 |
| HNCLMS | -26.5092 | -27.6192 | -28.1109 | -28.3568 | -28.4688 |

La Tabla 4 muestra un resumen del MSE del algoritmo RLS para las estructuras *Constrained*, *GSC* y *Householder* versus los diferentes *forgetting factor* utilizados, las respuestas marcadas con negrilla representan el mejor desempeño del algoritmo para cada una de las estructuras.

Tabla 4. MSE en dB del algoritmo RLS con λ variable

| Algoritmo RLS | | | | | |
|---------------|-----------------|----------|----------|----------|----------|
| λ | 0.9991 | 0.9993 | 0.9995 | 0.9999 | 1.0000 |
| CRLS | -32.9463 | -32.7339 | -32.4385 | -30.8572 | -27.5875 |
| GSC-RLS | -33.8434 | -33.7595 | -33.6325 | -32.8125 | -30.5932 |
| HCRLS | -33.8434 | -33.7595 | -33.6325 | -32.8125 | -30.5932 |

En la Tabla 5 se muestra un resumen del MSE del algoritmo CG para las estructuras *Constrained*, *GSC* y *Householder* versus los diferentes parámetros adaptativos de λ_f utilizados y con $\eta = 0.70$. Las respuestas marcadas con negrilla representan el mejor desempeño del algoritmo para cada una de las estructuras.

Tabla 5. MSE en dB del algoritmo CG con λ_f variable

| Algoritmo CG | | | | | |
|--------------|-----------------|-----------------|----------|-----------------|----------|
| λ_f | 0.9991 | 0.9993 | 0.9997 | 0.9999 | 1.0000 |
| CCG | -34.1359 | -34.1565 | -34.1125 | -34.0572 | -33.8006 |
| GSC-CG | -34.0706 | -34.0696 | -34.0690 | -33.9821 | -33.7131 |
| HCCG | -34.2268 | -34.2460 | -34.2730 | -34.3252 | -34.2856 |

El MSE del algoritmo CG para las estructuras *Constrained*, GSC y *Householder* versus los diferentes parámetros adaptativos de η utilizados y con λ_f fijo según el mejor desempeño obtenido para el algoritmo (Tabla 5) se presenta en la Tabla 6. Las respuestas marcadas con negrilla representan el mejor desempeño del algoritmo para cada una de las estructuras.

Tabla 6. MSE en dB del algoritmo CG con η variable

| Algoritmo CG | | | | | |
|--------------|----------|----------|-----------------|-----------------|----------|
| η | 0.50 | 0.60 | 0.70 | 0.80 | 0.90 |
| CCG | -34.0390 | -34.1037 | -34.1565 | -34.1853 | -34.1713 |
| GSC-CG | -33.8531 | -33.9593 | -34.0706 | -34.0597 | -34.0506 |
| HCCG | -34.2288 | -34.3102 | -34.3253 | -34.3041 | -34.2829 |

Si se compara el rendimiento algoritmo por algoritmo para las tres estructuras: *Constrained*, GSC y *Householder* resultan ser casi similares. Para una mejor idea en la Figura 32 se implementa el algoritmo RLS comparando las tres estructuras y se tiene que para las estructuras GSC y *Householder* existe una sola curva siendo su comportamiento exactamente el mismo, en tanto que la estructura *Constrained* presenta un rendimiento parecido a las otras dos.

Si bien, la teoría indica que en condiciones iguales de inicialización la respuesta de las tres estructuras debe ser las mismas, esto no sucede en la presente investigación, puesto que los micrófonos no presentan la misma respuesta (isotrópica) cuando se comparan entre cada uno de ellos, como se pudo observar en la Figura 9. Otra razón es porque no se puede garantizar a un cien por ciento la estacionalidad del fenómeno debido entre otras cosas a parámetros externos, como por ejemplo la manipulación del usuario. Además porque se preconditiono la matriz \mathbf{C} en la estructura *Constrained* por los problemas ya mencionados.

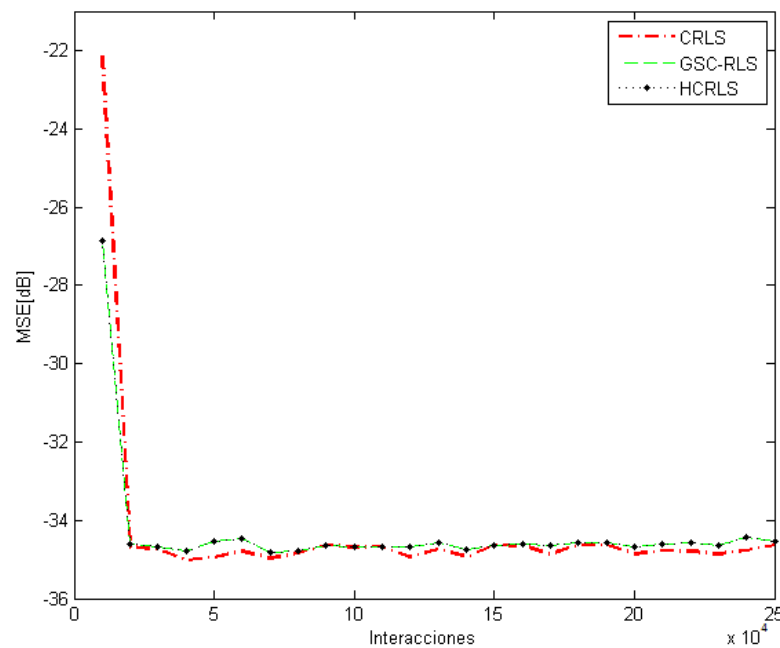


Figura 32. MSE de las estructuras del algoritmo RLS para un arreglo ULA

3.3.4. Comparación de las estructuras LCMV

3.3.4.1. Comparación de los Algoritmos *Constrained*

Las respuestas de la estructura *Constrained* se presentan en las Figuras 33, 34 y 35. Los parámetros de adaptación que se utilizan para cada uno de los algoritmos se detallan en las Tablas 2, 3, 4 y las Tablas 5, 6 para los algoritmos CLMS, NCLMS, CRLS y CCG respectivamente.

La Figura 33 muestra el MSE de los cuatro algoritmos, como se puede observar la curva de los algoritmos CLMS y NCLMS son de lenta convergencia. Los algoritmos CRLS y CCG presentan una curva similar y posee una velocidad de convergencia más alta. El MSE se ilustra adicionalmente en la Figura 34 donde se representa los algoritmos con todas las interacciones, no se aplica el método de Montecarlo a esta figura porque se está trabajando en tiempo real, pero se puede observar como los cuatro algoritmos convergen, confirmando que los algoritmos CRLS y CCG poseen una mayor velocidad de convergencia alcanzando

aproximadamente un MSE de -35 dB, mientras que los algoritmos CLMS y NLMS son de lenta convergencia alcanzando un MSE aproximado de -30 dB.

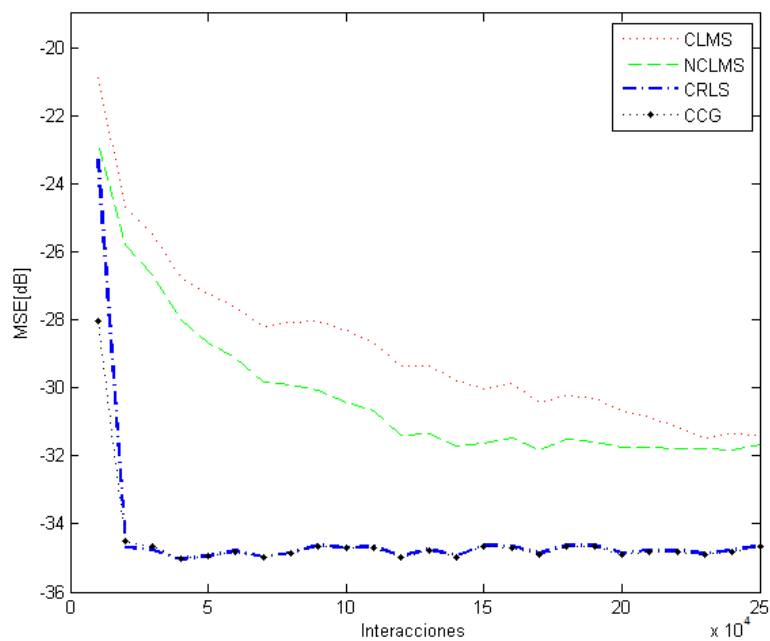


Figura 33. Comparación de los algoritmos *Constrained* respecto al MSE para un arreglo ULA

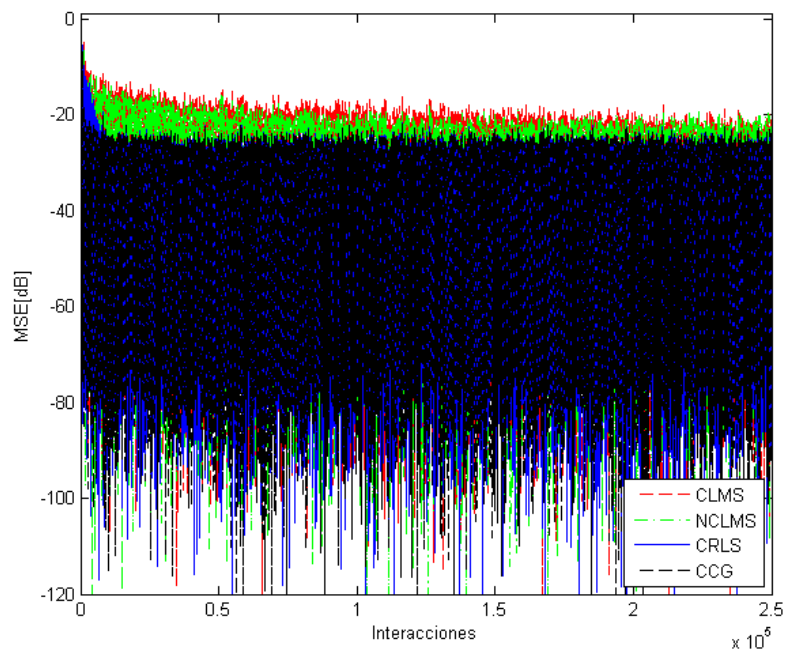


Figura 34. Comparación de los algoritmos *Constrained* respecto al MSE para un arreglo ULA con todas las interacciones

En la Figura 35 se muestra el *beam pattern* (patrón de directividad) de cada uno de los algoritmos *Constrained*, según dicha figura los algoritmos presentan una buena resolución en sus *beam patterns* y el ángulo obtenido coincide con el de la señal de entrada de 90° . El algoritmo NCLMS y CCG presentan mayor ganancia, mientras que el algoritmo CRLS posee aproximadamente 1 dB menos de ganancia que los algoritmos NCLMS y CCG. El algoritmo CLMS es el que posee menor ganancia versus a los otros tres algoritmos, aproximadamente 2.5 dB menos

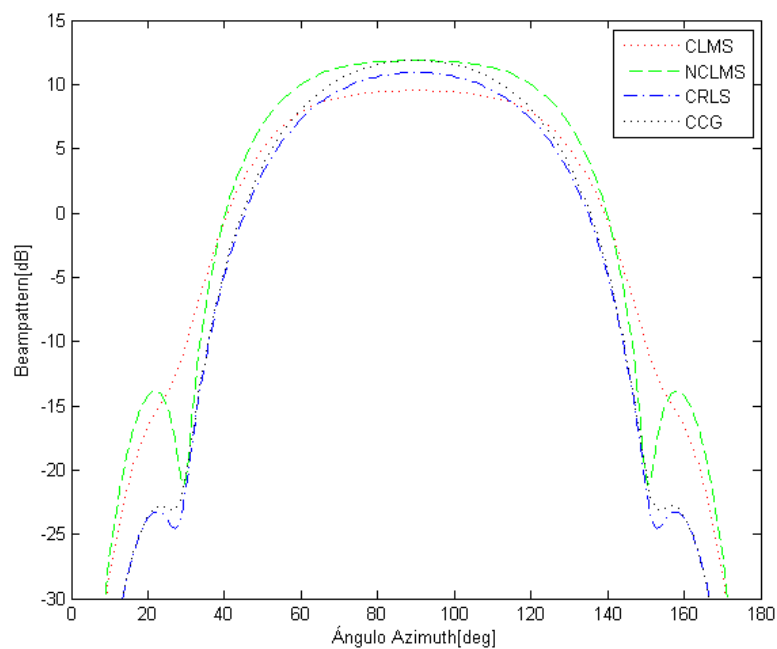


Figura 35. Comparación de los algoritmos *Constrained* respecto al *beam pattern* para un arreglo ULA

3.3.4.2. Comparación de la Estructura GSC

Los parámetros de adaptación que se utilizan para cada uno de los algoritmos en la estructura GSC se detallan en las Tablas 2, 3, 4 y las Tablas 5, 6 para los algoritmos GSC-LMS, GSC-NLMS, GSC-RLS y GSC-CG respectivamente. El MSE de los cuatro algoritmos se muestra en la Figura 36. Los algoritmos GSC-LMS y GSC-NLMS son de lenta convergencia, mientras que los algoritmos GSC-RLS y GSC-CG poseen una velocidad de convergencia alta. Este hecho se ilustra adicionalmente en la Figura 37 donde se representa el MSE de los algoritmos con todas las interacciones, al igual que en la estructura anterior no se puede aplicar el

método de Montecarlo pero se puede observar como los algoritmos convergen confirmando que los algoritmos GSC-LMS y GSC-NLMS son de lenta convergencia obteniendo aproximadamente un MSE de -30 dB versus los algoritmos GSC-RLS y GSC-CG que poseen una velocidad de convergencia alta alcanzando un MSE aproximado de -35 dB.

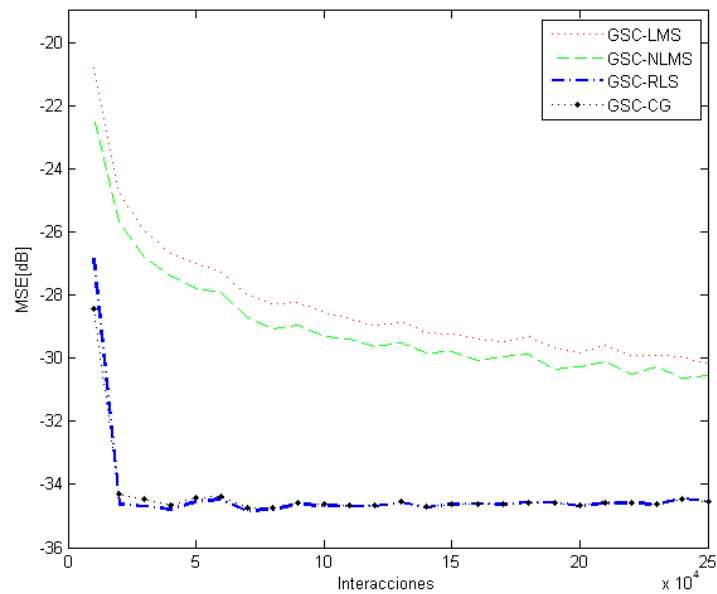


Figura 36. Comparación de los algoritmos GSC respecto al MSE para un arreglo ULA

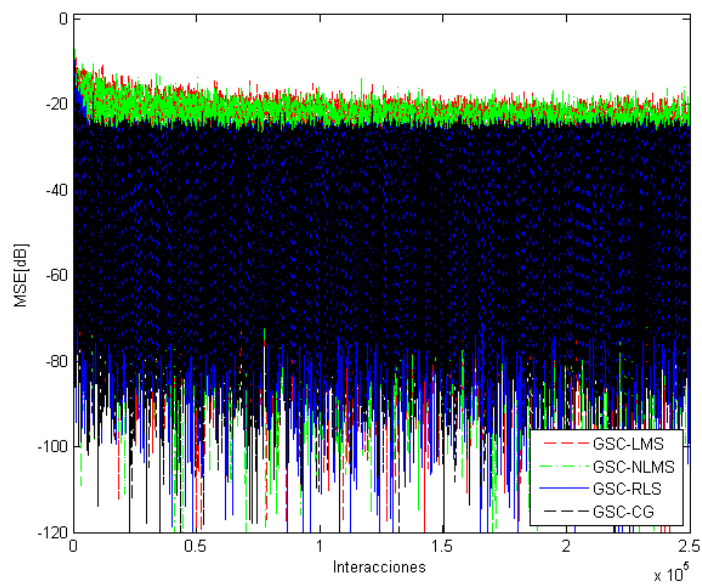


Figura 37. Comparación de los algoritmos GSC respecto al MSE para un arreglo ULA con todas las interacciones

Los *beampattern* de cada uno de los algoritmos GSC se muestran en la Figura 38. Los algoritmos GSC-RLS, GSC-CG y GSC-NLMS presentan una alta resolución, mientras que el algoritmo GSC-LMS tienen una baja resolución en comparación a los otros tres algoritmos. La ganancia de los algoritmos GSC-RLS y GSC-CG es la más alta ya que poseen 5 *dB* más que los algoritmos GSC-LMS y GSC-NLMS.

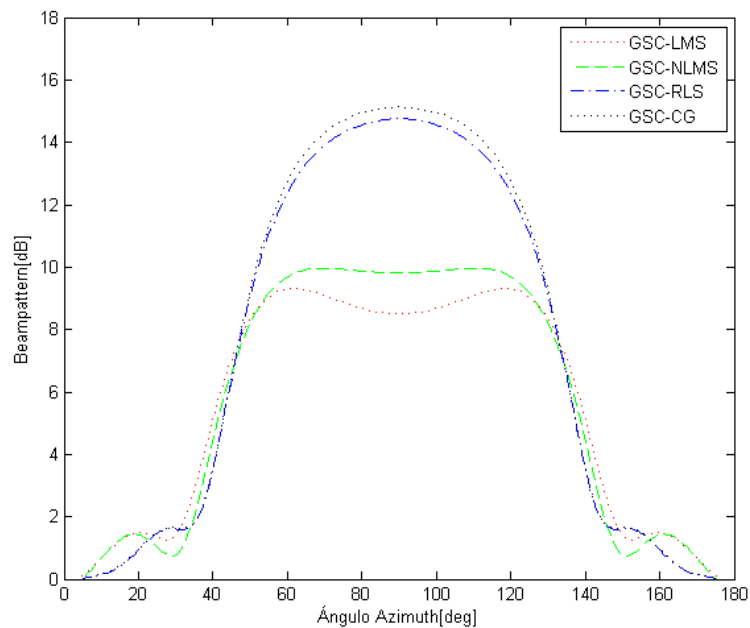


Figura 38. Comparación de los algoritmos GSC respecto al *beam pattern* para un arreglo ULA

3.3.4.3. Comparación de la Estructura *Householder*

Para la estructura *Householder* los parámetros de adaptación que se utilizan en cada uno de los algoritmos se detallan en las Tablas 2, 3, 4 y las Tablas 5, 6 para los algoritmos HCLMS, HNCLMS, HCRLS y HCCG respectivamente. La respuesta del MSE de los cuatro algoritmos se presenta en la Figura 39. La estructura *Householder* posee un comportamiento similar a las otras dos estructuras, siendo los algoritmos HCLMS y HNCLMS de lenta convergencia, mientras que los algoritmos HCRLS y HCCG poseen una alta velocidad de convergencia alta. Este hecho adicionalmente se muestra en la Figura 40, en la cual se representa el MSE de los algoritmos con todas las interacciones, al igual que en las estructuras anteriores no se puede aplicar el método de Montecarlo pero se puede analizar como el MSE disminuye desde 0 *dB* hasta aproximadamente 30 *dB* en el caso de los algoritmos HCLMS y HNCLMS y

los algoritmos HCRLS y HCCG disminuyen su MSE de 0 dB hasta aproximadamente 35 dB.

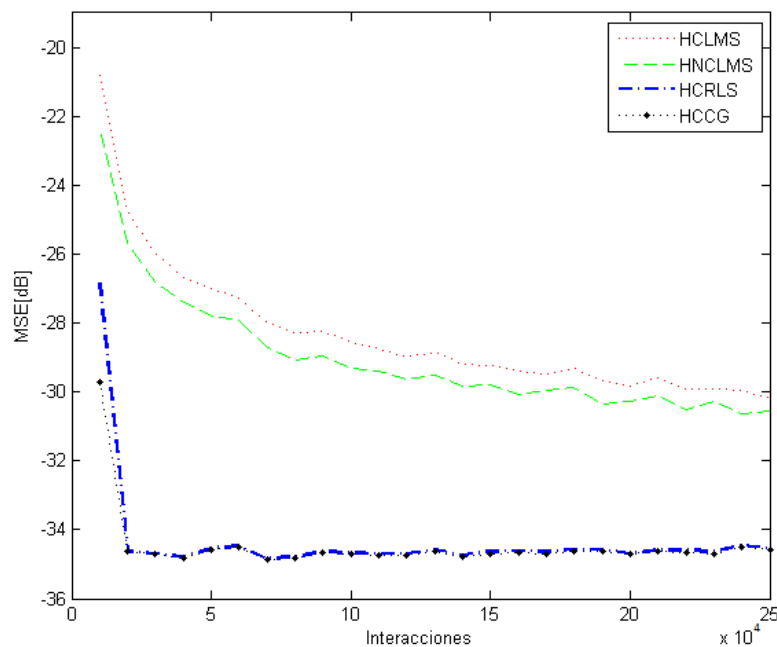


Figura 39. Comparación de los algoritmos *Householder* respecto al MSE para un arreglo ULA

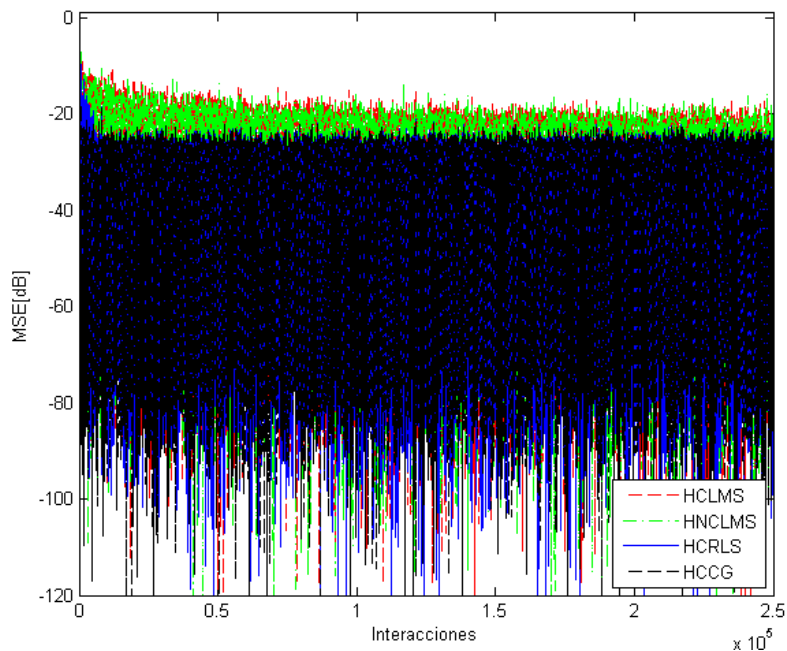


Figura 40. Comparación de los algoritmos *Householder* respecto al MSE para un arreglo ULA con todas las interacciones

El *beam pattern* en la estructura *Householder* tiene una mejor resolución que las otras dos estructuras como se puede observar en la Figura 41. Los algoritmos HCRLS y HCCG poseen un similar comportamiento, por tanto una mayor ganancia, de aproximadamente 6 dB más que los algoritmos HCLMS y HNCLMS.

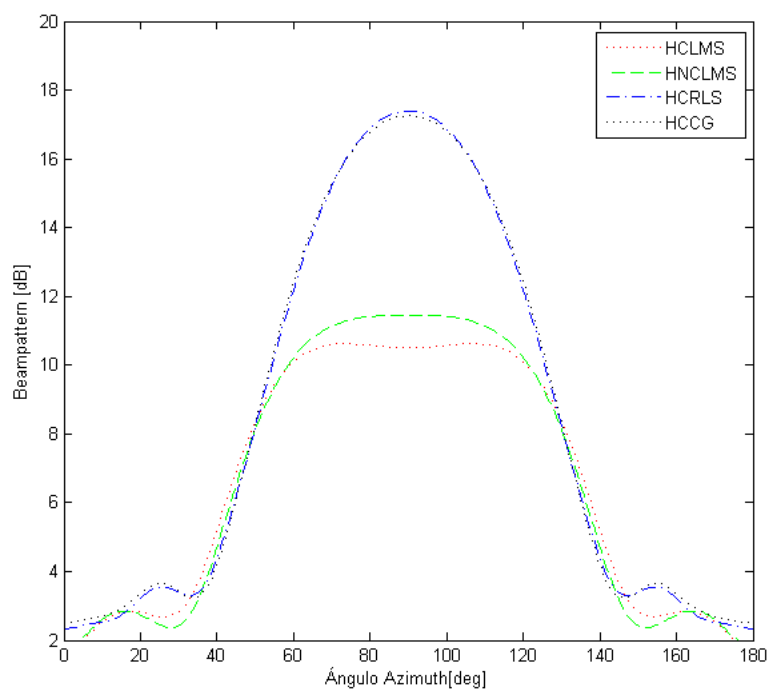


Figura 41. Comparación de los algoritmos *Householder* respecto al *beam pattern* para un arreglo ULA

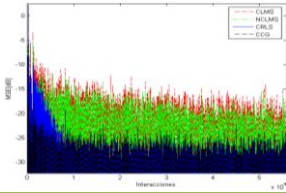
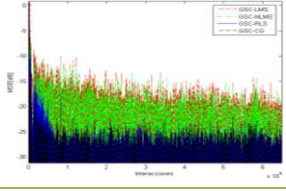
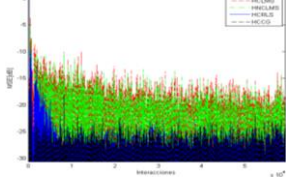
3.3.4.4. Velocidad de convergencia de los algoritmos LCMV

La velocidad de convergencia de un algoritmo depende del tamaño de su parámetro de adaptación. Para el caso del algoritmo LMS y NLMS que son algoritmos de convergencia lenta dependen del *step size* que a medida que este aumenta de tamaño su velocidad de convergencia es mayor. Los algoritmos RLS y CG poseen un *forgetting factor* entre 0 y 1, cuando más cercano este valor se encuentre a uno más rápida es la convergencia y si por el contrario se acerca a cero más lenta es la convergencia.

Los algoritmos RLS y CCG según la Tabla 7 presentan una mayor velocidad de convergencia al necesitar aproximadamente 10000 interacciones para convergir,

mientras que los algoritmos LMS y NLMS necesitan aproximadamente 40000 interacciones siendo estos más lentos. Lo indicado se cumple para las tres estructuras.

Tabla 7. Velocidad de convergencia de los algoritmos LCMV

| Estructura LCMV | Algoritmo | N° Aproximado de Interacciones | MSE | Figuras |
|--------------------|-----------|--------------------------------|----------|---------------------------------------------------------------------------------------|
| <i>Constrained</i> | LMS | 40 000 | -27.9188 |  |
| | NLMS | 40 000 | -29.4207 | |
| | RLS | 10 000 | -32.9463 | |
| | CG | 10 000 | -34.1853 | |
| GSC | LMS | 40 000 | -27.6321 |  |
| | NLMS | 40 000 | -28.4688 | |
| | RLS | 10 000 | -33.8434 | |
| | CG | 10 000 | -34.0706 | |
| <i>Householder</i> | LMS | 40 000 | -27.6321 |  |
| | NLMS | 40 000 | -28.4688 | |
| | RLS | 10 000 | -33.8434 | |
| | CG | 10 000 | -34.3253 | |

3.3.4.5. Costo computacional de los algoritmos LCMV

La complejidad computacional de un algoritmo no solo se determina por sí mismo como ya se indicó anteriormente. Para determinar el costo computacional de los algoritmos LCMV se considera el número de multiplicaciones.

La inversión de una matriz es la peor operación $O[N^3]$ por lo cual se emplea el proceso de la matriz de Lema. El número de multiplicaciones de cada uno de los algoritmos se presenta en la Tabla 8.

Al preconditionar la matriz de restricciones \mathbf{C} y alcanzar un mejor desempeño en la estructura *Constrained* como se indicó en el ítem anterior, trae como desventada el

aumento en el costo computacional en $O[N^2]$ multiplicaciones, por tanto el costo computacional en la estructura *Constrained* aumenta (Golub & Van Loan, 2013). En la Tabla 8 se considera en la estructura *Constrained* el costo computacional de la matriz *toeplitz*.

Tabla 8. Costo computacional de los algoritmos LCMV

| Algoritmo | <i>Constrained</i> | GSC | <i>Householder</i> |
|-----------|-------------------------|--------------------------------------|------------------------------------|
| LMS | $N^2 + (2p + 2)N + 1$ | $N^2 + (3 - p)N - (2p - 1)$ | $(2p + 2)N - (p^2 - 1)$ |
| NLMS | $N^2 + (3p + 3)N + 1$ | $N^2 + (4 - p)N - (3p - 1)$ | $(2p + 3)N - (p^2 + p - 1)$ |
| RLS | $6N^2 + (19p + 4)N + p$ | $5N^2 + (6 - 9p)N + 4p^2 - 5p$ | $4N^2 + (5 - 6p)N + 2p^2 - 3p + 2$ |
| CG | $4N^2 + (4p + 10)N + 1$ | $4N^2 + (11 - 7p)N + 3p^2 - 10p + 1$ | $3N^2 + (10 - 4p)N + p^2 - 8p + 2$ |

Donde se determina que N representa el número de *taps* y p el número de restricciones. Cada costo computacional debe ser multiplicado por el número de micrófonos utilizados. Según la Tabla 8 se puede analizar que los algoritmos de menor costo computacional son el LMS y NLMS y los de mayor costo computaciones los algoritmos RLS y CG. La estructura que tiene el menor costo computacional es la *Householder*, seguida de la estructura GSC y la de mayor costo computacional es la estructura *Constrained*.

Por tanto, los algoritmos LMS y NLMS presentan velocidades bajas de convergencia pero el algoritmo NLMS presenta un menor MSE, esto sucede para las tres estructuras como se pudo observar en la Tabla 7. En cuanto a la resolución de los *beam patterns*, varía dependiendo de la estructura utilizada, pero el algoritmo NLMS presenta una mejor resolución versus al algoritmo LMS. Lo que corresponde al costo computacional entre estos dos algoritmos, los mismos presentan un bajo costo computacional como se detalló en la Tabla 8, pero entre estos dos algoritmos el LMS es el de menor costo.

Si comparamos los algoritmos RLS y CG se tiene que la velocidad de convergencia está directamente relacionada a sus *forgetting factor*, los dos

algoritmos presentan una rápida velocidad de convergencia como se mostró en la Tabla 7. El algoritmo CG presenta un mejor comportamiento puesto que el algoritmo RLS puede divergir en cualquier instante. Para las tres estructuras los algoritmos RLS y CG presentan una alta resolución como se pudo observar en las Figuras 35, 38 y 41. El costo computacional de los dos algoritmos es alto, siendo el algoritmo RLS el que mayor complejidad computacional versus al CG como se detalló en la Tabla 8.

Hasta el momento los algoritmos del tipo LCMV han sido comparados bajo un arreglo ULA. En el siguiente capítulo se analiza la geometría de un arreglo semiesférico puesto que a futuro se desea la implementación en su primera fase en un casco, para lo cual se utilizarán los algoritmos NLMS y CG por las ventajas antes mencionadas.

3.4. Análisis del Número de *Taps*

El costo computacional de los algoritmos depende directamente de N que es el número de *taps* del filtro, es por ello que en esta sección se pretende variar el número de *taps* y verificar que desempeño tiene las estructuras al variar dicho número de *taps*. Al variar el número de *taps* N la respuesta de los algoritmos también varía.

En la Figura 42 se presenta el MSE de los algoritmos *Constrained* NLMS y CG cuando se varía el número de *taps* en $N = 5$, $N = 10$ y $N = 15$. El algoritmo NCLMS presenta una menor velocidad de convergencia al aumentar el número de *taps*. Para $N = 5$ el algoritmo converge más rápido con un valor en su MSE de -29 dB y para $N = 10$ converge con un poco más de interacciones pero con un valor en el MSE de -31 dB . Esto se debe que a media que se tienen un menor número de coeficientes el algoritmo realiza menos interacciones y por lo tanto su velocidad de convergencia será mayor.

En el algoritmo CCG al variar el número de *taps* se obtiene un comportamiento diferente, en comparación al algoritmo NCLMS puesto que para $N = 5$, $N = 10$ y $N = 15$ se tiene la misma velocidad de convergencia pero para $N = 5$ y $N = 15$ se

obtiene un valor en el MSE aproximado de -30 dB y para $N = 10$ el desempeño es mejor al obtener un valor del MSE de -35 dB. El comportamiento del CCG se debe a la propia estructura del algoritmo CG que por ser de rápida convergencia puede compensar la falta de algunos coeficientes.

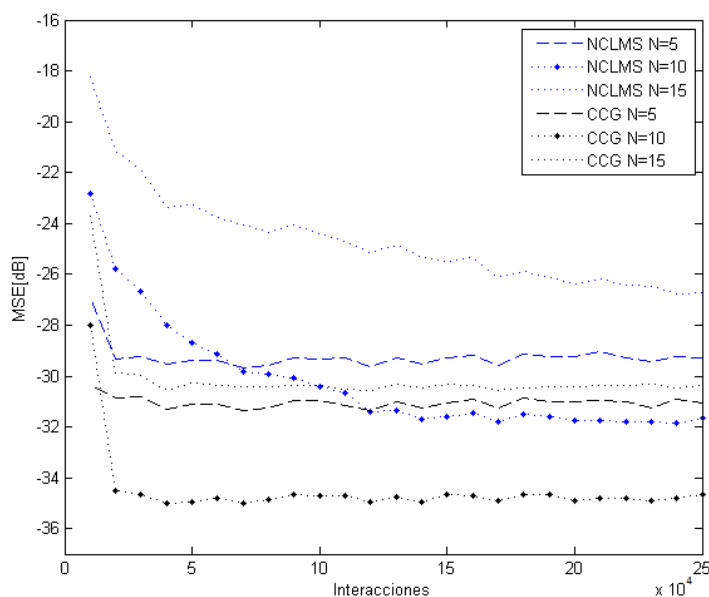


Figura 42. Algoritmos *Constrained* respecto al MSE con varios valores de N para un arreglo ULA

La resolución de los *beam pattern* para los dos algoritmos va mejorando a medida que el número de *taps* aumenta en $N = 5$, $N = 10$, y $N = 15$. Debido a que se tiene más coeficientes en el vector de pesos y los algoritmos tienen una mejor respuesta, como se puede observar en la Figura 43. La ganancia del algoritmo NCLMS es mayor en comparación al algoritmo CCG para cada uno de los valores de N utilizados. Para $N = 10$ se posee 3 dB más que con $N = 15$ para los dos algoritmos. Para $N = 5$ se obtiene una menor ganancia en comparación de $N = 10$ aproximadamente menos 3 dB para el algoritmo NCLMS y de menos 6 dB para el algoritmo CCG.

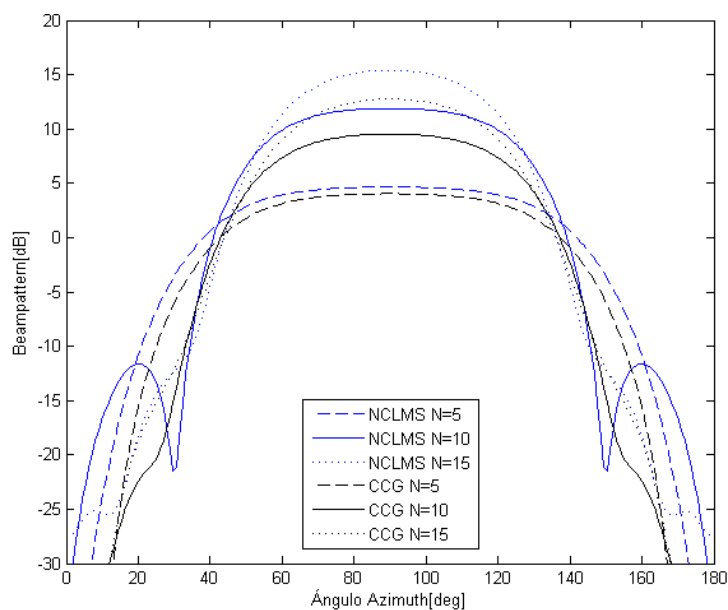


Figura 43. Algoritmos *Constrained* respecto al *beampattern* con varios valores de N para un arreglo ULA

Los algoritmos NLMS y CG en las estructuras GSC y *Householder* poseen un similar comportamiento en su valor del MSE como se puede observar en la Figuras 44 y en la Figura 46 respectivamente, al variar el número de *taps* en $N = 5$, $N = 10$, y $N = 15$. Para los algoritmo GSC-NLMS y HNCLMS la velocidad de convergencia disminuye a medida que el número de *taps* aumenta, como ya se indicó en la estructura *Constrained* esto se debe que a medida que se tiene un menor número de *taps* el algoritmo tiene menor interacciones por lo tanto su velocidad de convergencia aumenta. El algoritmo GSC-CG y HCCG tienen un comportamiento diferente puesto que al variar el número de *taps* se obtiene una misma velocidad de convergencia pero el MSE disminuye a medida que el número de *taps* aumenta. Para $N = 5$ el valor del MSE es de aproximadamente -32 dB y para $N = 10$ y $N = 15$ se tiene un valor del MSE de -35 dB. Como ya se indicó en la estructura *Constrained* esto se debe a que el algoritmo CG posee una alta velocidad de convergencia que permite compensar la falta de coeficientes.

La estructura GSC tiene una respuesta diferente en los *beampatterns* en comparación a la estructura *Constrained*, como se puede observar en la Figura 45. El algoritmo GSC-CG es el que posee mayor ganancia en comparación al algoritmo GSC-NLMS para cada uno de los *taps* utilizados. El algoritmo GSC-NLMS tiene una

buena resolución para $N = 5$ y $N = 10$, mientras que para $N = 15$ la resolución empieza a disminuir. Para el algoritmo GSC-CG la mayor resolución y ganancia se obtiene para $N = 10$ ya que para $N = 5$ se tiene 3 dB menos de ganancia pero con una buena resolución. Al igual que el GSC-NLMS la resolución para el algoritmo GSC-CG empieza a disminuir para $N = 15$.

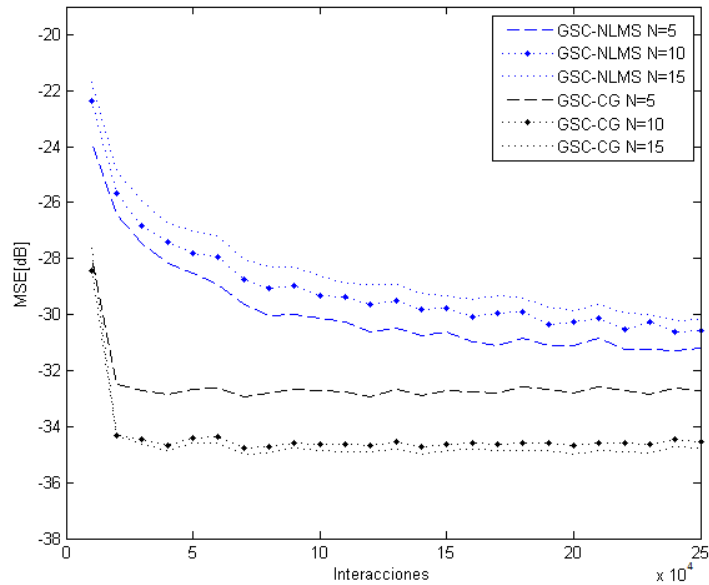


Figura 44. Algoritmos GSC respecto al MSE con varios valores de N para un arreglo ULA

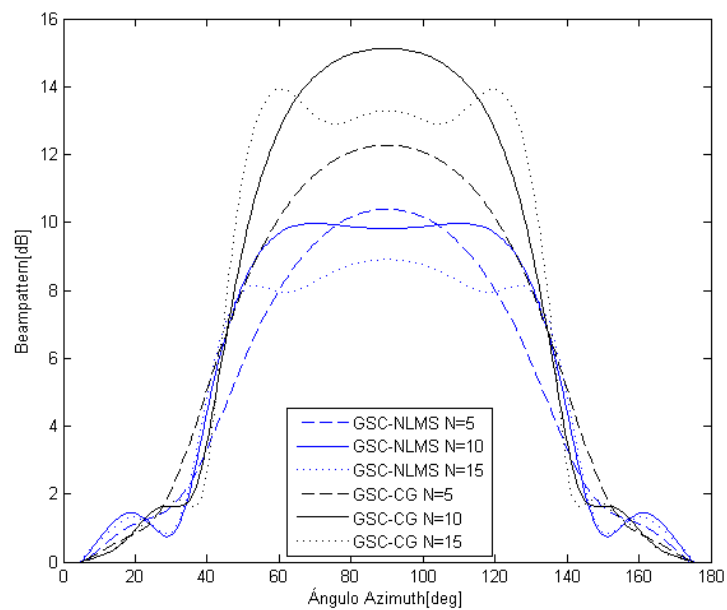


Figura 45. Algoritmos GSC respecto al *beam pattern* con varios valores de N para un arreglo ULA

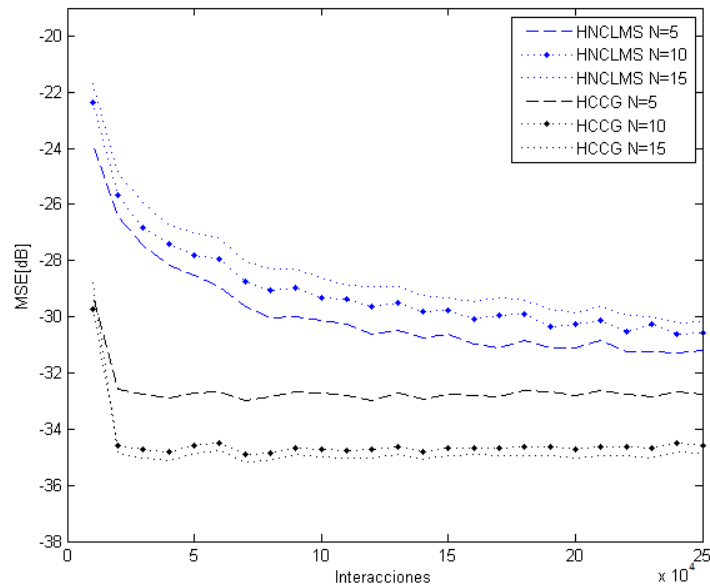


Figura 46. Algoritmos *Householder* respecto al MSE con varios valores de N para un arreglo ULA

Los *beampatterns* en la estructura *Householder* presenta una alta resolución para los algoritmos HNCLMS y HCCG. Para $N = 5$, $N = 10$, y $N = 15$ en el algoritmo HNCLMS la ganancia es similar, pero para el algoritmo HCCG la ganancia es mayor a media que el número de *taps* aumenta, para $N = 15$ la ganancia es 2 dB más que $N = 10$ y este 4 dB más que $N = 5$.

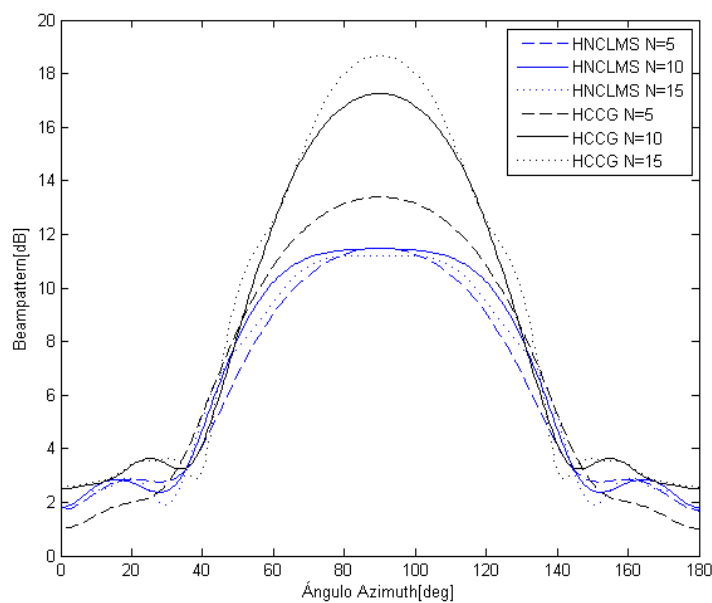


Figura 47. Algoritmos *Householder* respecto al *beampattern* con varios valores de N para un arreglo ULA

Del anterior análisis se concluye que para los algoritmos de convergencia alta, tal es el caso del algoritmo CG al variar el número de *taps* se sigue logrando un valor de MSE aceptable, mientras que para los algoritmos de convergencia lenta como es el algoritmo NLMS el número de coeficientes si afecta a su velocidad de convergencia, siendo este más lento a medida que se aumenta este número de coeficientes. Además a medida que el número de coeficientes aumenta en el algoritmo CG se obtiene una mayor ganancia porque al tener un mayor número de coeficientes le permite realizar las actualizaciones más rápidas.

CAPÍTULO IV

4. ANÁLISIS Y SIMULACIÓN DE LOS ALGORITMOS EN LA GEOMETRÍA PROPUESTA

4.1. Diseño del arreglo

En la literatura existen tres opciones para la configuración de un arreglo esférico (Huang & Balanis, 2005). La geometría que se utiliza en la presente investigación se basa en la configuración de L anillos con igual espaciamento, donde los M_L micrófonos tienen una distancia de separación uniforme.

El arreglo de micrófonos para la presente investigación consta de un semianillo $l = 1$ con 4 micrófonos espaciados uniformemente con una apertura entre cada micrófono adyacente de 60° , donde r es el radio con dimensión de 15 cm , posicionado en un plano (r_x, r_y) como se muestra en la Figura 48.

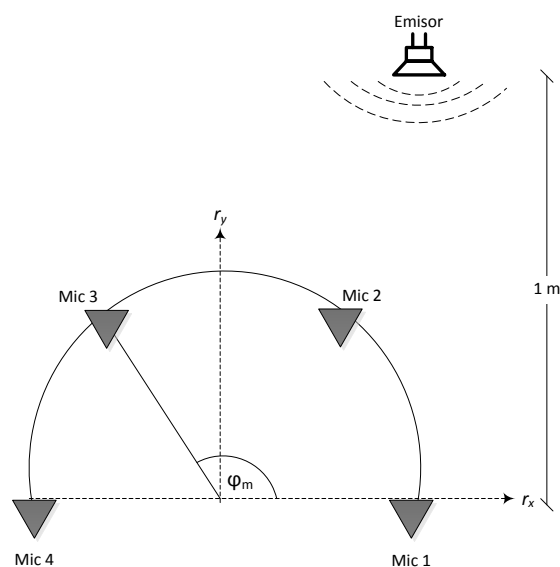


Figura 48. Arreglo Semiesférico

Las coordenadas esféricas de cada micrófono está dado por (r, θ, ϕ_m) . Donde $m = 0, \dots, 3$. Se utiliza 4 micrófonos por las aclaraciones indicadas en el capítulo anterior.

$$\theta = \frac{\pi}{2} \quad (73)$$

$$\varphi_m = \frac{m\pi}{M} \quad (74)$$

Si se supone un medio ideal, la velocidad de propagación de las ondas sonoras es constante y el tiempo de propagación depende solamente de la distancia recorrida. Para la presente investigación, el medio de interés es el aire cuyo valor es de 340 m/s (no se considera las alteraciones debido a la altura en relación al nivel del mar). Según la Figura 48 la distancia d_m establece un camino directo entre la fuente emisora y el eje r_x del arreglo de los micrófonos. La distancia $d_m = 1\text{m}$.

$$t_m = \frac{d_m}{g} \quad (75)$$

El conocimiento del tiempo de propagación del emisor a los micrófonos constituye la base para el cálculo de los diferentes tiempos de llegada ya que la señal llega a los micrófonos en distintos tiempos, por tanto el tiempo de propagación desde la fuente emisora a los micrófonos *Mic 1* y *Mic 4* será el siguiente:

$$t_1 = t_4 = \frac{1\text{m}}{340 \text{ m/s}} = 2,94 \text{ ms} \quad (76)$$

El tiempo de propagación desde la fuente emisora a los micrófonos *Mic 2* y *Mic 3* es:

$$t_2 = t_3 = \frac{0,87\text{m}}{340 \text{ m/s}} = 2,56 \text{ ms} \quad (77)$$

Entonces, la señal llega al micrófono *Mic 1* con un retardo de tiempo de $0,38\text{ ms}$ con relación al micrófono *Mic 2* y de manera similar sucede con la señal que arriba al micrófono *Mic 4* que lo hace con un retardo de $0,38\text{ ms}$ con relación al micrófono *Mic 3*.

Las señales recibidas por los micrófonos son muestreadas a una tasa $f = 22\ 050\text{ Hz}$, por tanto los micrófonos *Mic 1* y *Mic 4* sufrirán un retardo de 8 muestras en relación con los micrófonos *Mic 2* y *Mic 3*, respectivamente. Este retardo se obtiene median una regla de tres.

4.2. Comparación del desempeño de los algoritmos

Para la geometría semiesférica se analizan los algoritmos NLMS y CG por sus características, como se indicó en el capítulo anterior. El análisis del desempeño de los algoritmos en un arreglo semiesférico se debe a su futura implementación en un casco.

4.2.1.1. Comparación de la Estructura *Constrained*

La Figura 49 muestra el MSE de los algoritmo NCLMS y CCG, como se puede observar la curva del algoritmo NCLMS es de más lenta convergencia en comparación al algoritmo CCG que posee una velocidad de convergencia más alta.

El MSE se ilustra adicionalmente en la Figura 50 donde se representa los dos algoritmos con todas las interacciones, no se aplica el método de Montecarlo porque como ya se indicó se está trabajando en tiempo real y la señal es muy ruidosa, pero se puede observar como los algoritmos convergen alcanzando el algoritmo NCLMS un MSE promedio de -28.8352 dB , mientras que el algoritmo CCG alcanza un MSE menor de -33.8471 dB . En la Figura 51 se muestra el *beam pattern* de los algoritmo NCLMS y CCG, según dicha figura el ángulo obtenido coincide con el de la señal de entrada de 90° . Los dos algoritmos presentan una alta resolución y una misma ganancia.

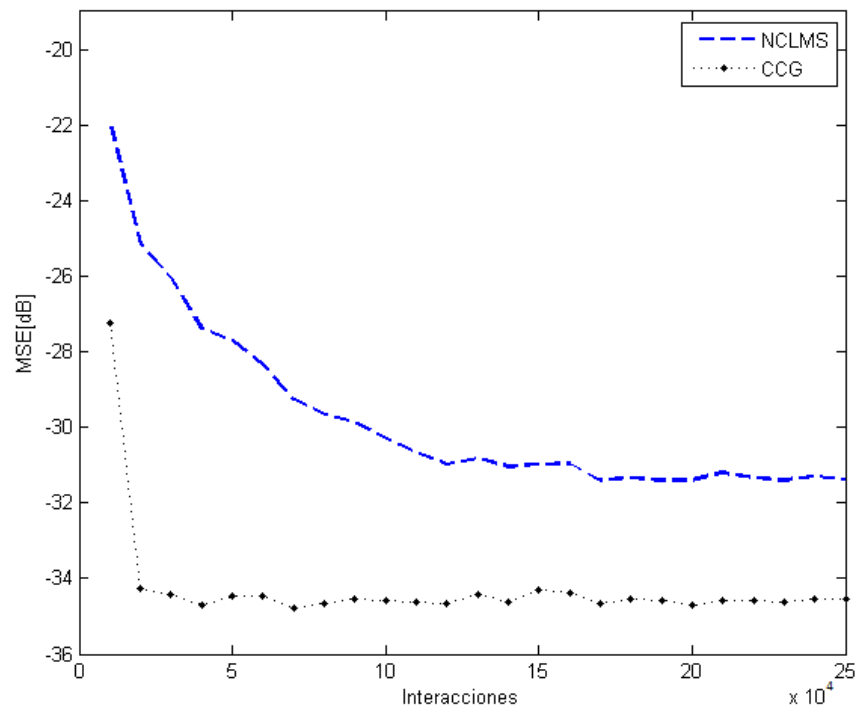


Figura 49. Comparación de los algoritmos *Constrained* respecto al MSE para un arreglo semiesférico

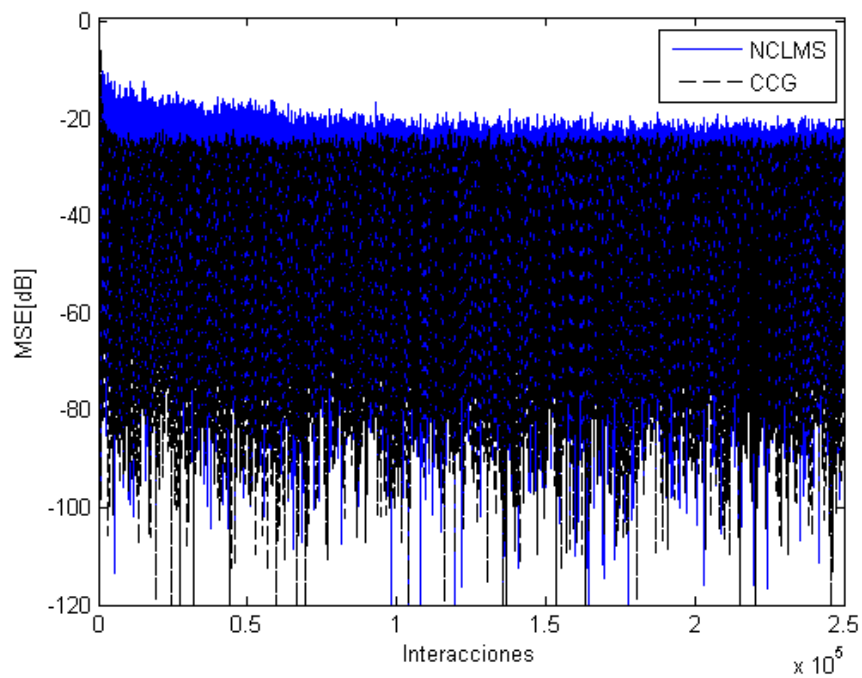


Figura 50. Comparación de los algoritmos *Constrained* respecto al MSE para un arreglo semiesférico con todas las interacciones

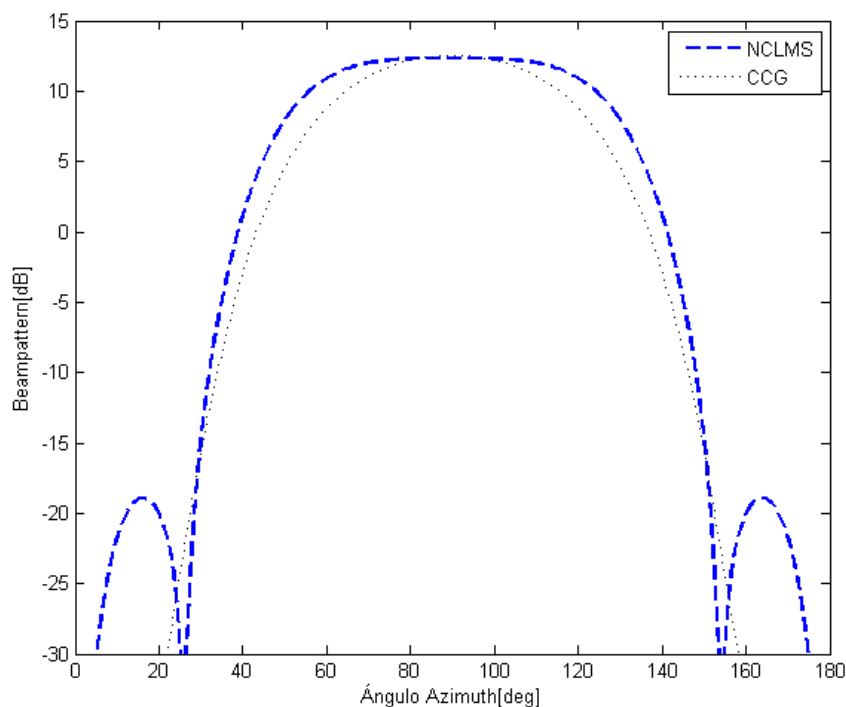


Figura 51. Comparación de los algoritmos *Constrained* respecto al *beampattern* para un arreglo semiesférico

4.2.1.2. Comparación de la Estructura GSC

En la Figura 52 se presenta el MSE de los algoritmos GSC-NLMS y GSC-CG, como se puede observar posee un comportamiento similar a la estructura *Constrained*. La curva del algoritmo NCLMS es de más lenta convergencia en comparación al algoritmo CCG que posee una alta velocidad de convergencia.

El MSE se ilustra adicionalmente en la Figura 53 donde se representa los dos algoritmos con todas las interacciones en la que se puede observar como los algoritmos convergen alcanzando de 0 dB un MSE promedio de -30.0257 dB para el algoritmo GSC-NLMS y para el algoritmo GSC-CG un MSE menor de -36.0157 dB. La Figura 54 muestra el *beampattern* de los algoritmos. El algoritmo GSC-NLMS posee una baja resolución, mientras que el algoritmo GSC-CG tienen una alta resolución y una ganancia mayor de 4 dB.

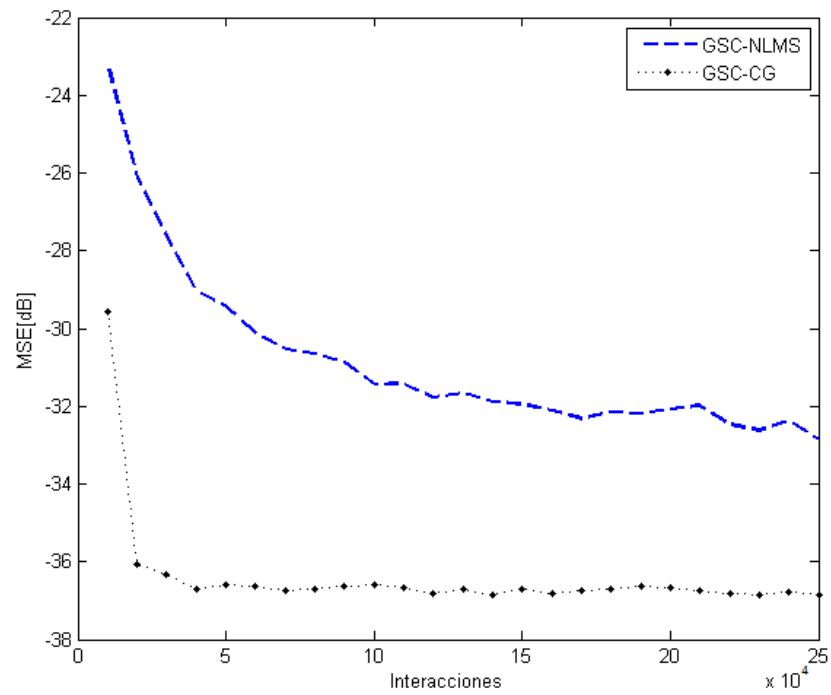


Figura 52. Comparación de los algoritmos GSC respecto al MSE para un arreglo semiesférico

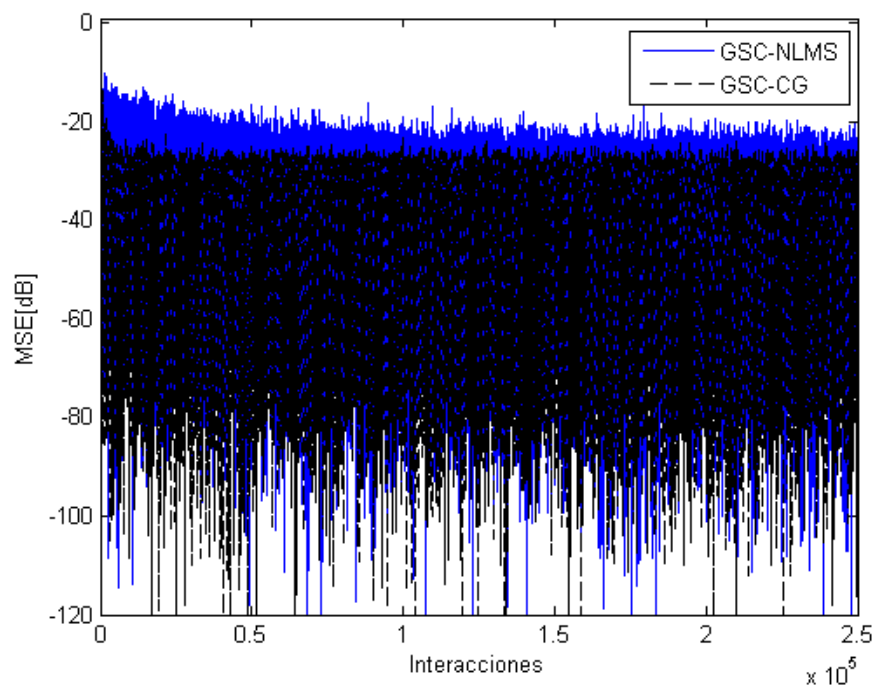


Figura 53. Comparación de los algoritmos GSC respecto al MSE para un arreglo semiesférico con todas las interacciones

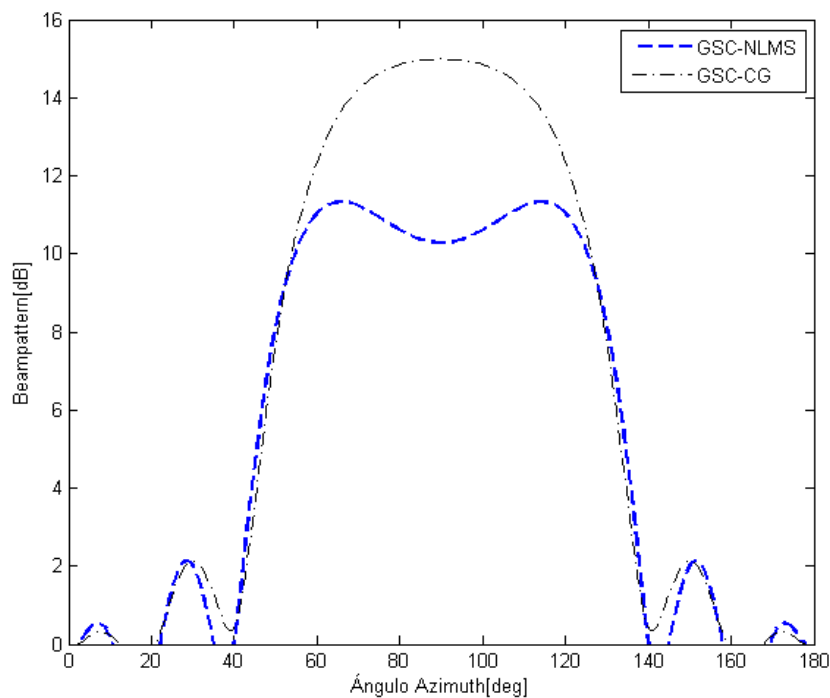


Figura 54. Comparación de los algoritmos GSC respecto al *beam pattern* para un arreglo semiesférico

4.2.1.3. Comparación de la Estructura *Householder*

Los algoritmos *Householder* presentan una similar respuesta en el MSE en comparación a las otras dos estructuras. En la Figura 55 se presenta el MSE de los algoritmos HNCLMS y HCCG. El algoritmo HNCLMS es de lenta convergencia y el algoritmo HCCG posee una alta velocidad de convergencia.

El MSE se ilustra adicionalmente en la Figura 56 con todas las interacciones en la que se puede observar como los algoritmos convergen alcanzando de 0 dB un MSE promedio de -30.2792 dB para el algoritmo HCNLMS y -36.5247 dB para el algoritmo HCCG. El *beam pattern* de los algoritmos *householder* se muestra en la figura 57. El algoritmo HNCLMS posee una baja resolución en comparación del algoritmo HCCG que además posee una ganancia mayor de 4 dB .

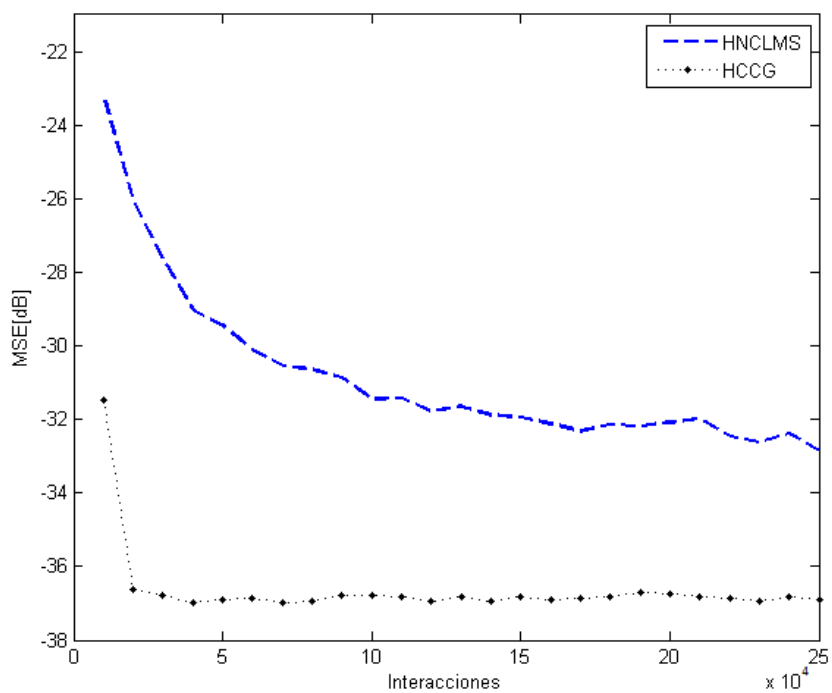


Figura 55. Comparación de los algoritmos *Householder* respecto al MSE para un arreglo semiesférico

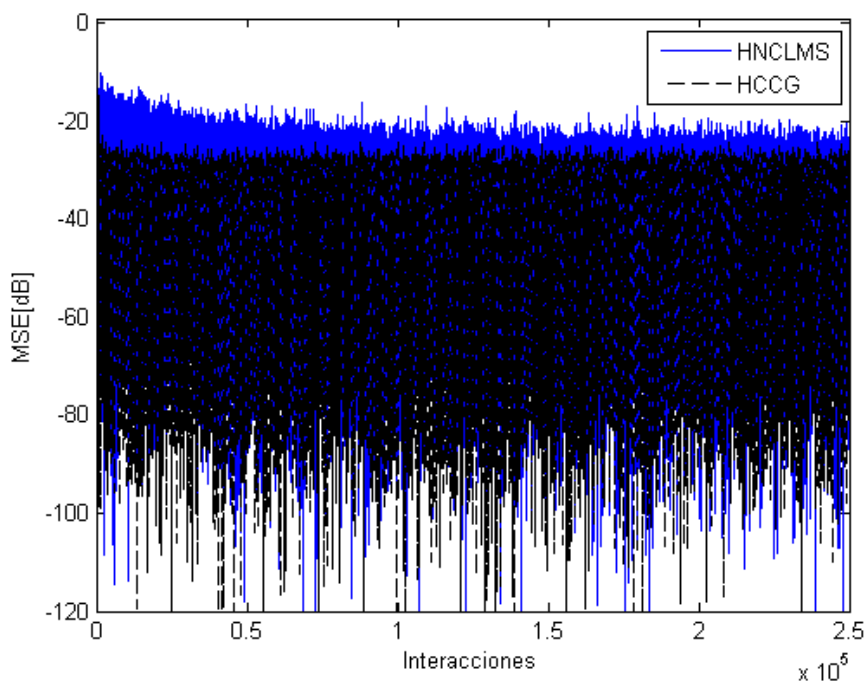


Figura 56. Comparación de los algoritmos *Householder* respecto al MSE para un arreglo semiesférico con todas las interacciones

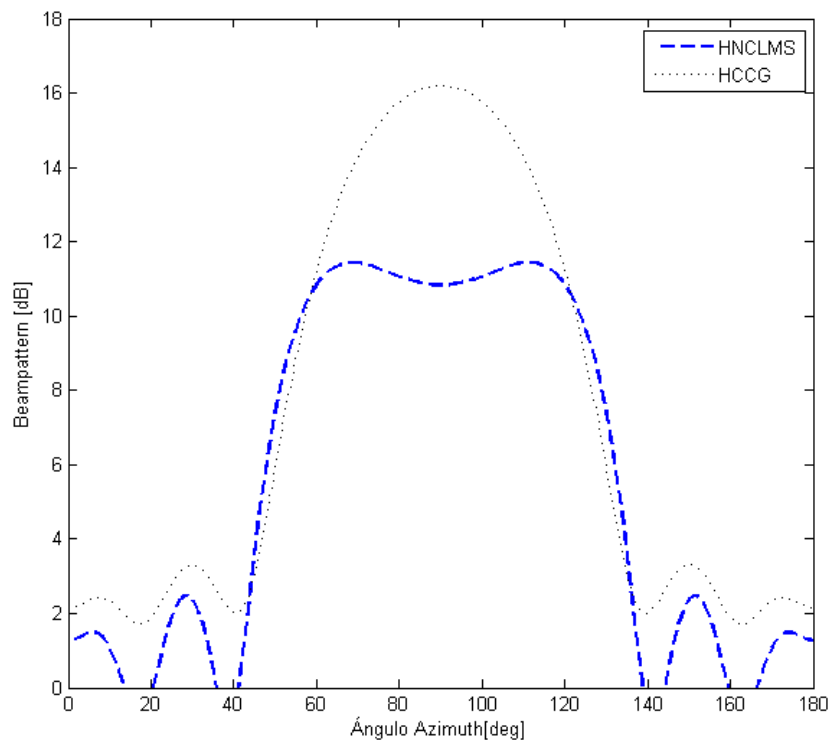


Figura 57. Comparación de los algoritmos *Householder* respecto al *beampattern* para un arreglo semiesférico

4.3. Comparación del Diseño Propuesto

Los algoritmos NLMS y CG presentan un valor en sus MSE similar para las tres estructuras y un buen desempeño porque se obtiene un valor del MSE pequeño. El *beampattern* varía en las tres estructuras, por tanto se tiene que el algoritmo GSC-NLMS presenta una baja resolución en comparación a las otras dos estructuras, aunque el algoritmo CG posee una alta resolución para las tres estructuras esto se debe a las características propias del algoritmo.

En los primeros capítulos de la presente investigación se verificó el desempeño de las tres familias de algoritmos LCMV: *Constrained*, GSC y *Householder* en un arreglo ULA de micrófono con $M = 4$, como resultado de este análisis se pudo constatar con el estado del arte las ventajas y limitaciones de cada una de las familias. De las características de los algoritmos se pudo rescatar dos algoritmos NLMS y CG en las tres estructuras.

Se escoge el algoritmo NLMS porque posee mayor velocidad de convergencia versus al algoritmos LMS sin un aumento significativo en su costo computacional, lo cual lo torna deseable para implementación de dispositivos con prestaciones de hardware limitadas. Mientras que el algoritmo CG posee una alta velocidad de convergencia al igual que el algoritmo RLS pero con las ventajas de ser un algoritmo estable y con un costo computacional menor que el RLS en sus tres estructuras, el único inconveniente es presenta el algoritmo CG es que se debe ajustar dos parámetros de adaptación versus a los otros algoritmos que ajusta un parámetro.

También se varió el número de *taps* y como era de esperar el desempeño de los algoritmos NLMS y CG depende de su versatilidad. Los de rápida convergencia pueden manejar mayor número de *taps* con un valor del MSE pequeño, mientras que algoritmos de lenta convergen con un número grande de *taps* se obtiene un valor de MSE grande.

La familia *Constrained* presenta el inconveniente en su punto inicial de partida, al no ser el adecuado, por lo que se tuvo que condicionar la matriz mediante la matriz *toeplitz* lo que involucro un mayor costo computacional a la estructura *Constrained*. La estructura GSC posee un desempeño aceptable pero siendo para la presente aplicación muy sensible a cambios pequeños, por lo que esta estructura es recomendable en aplicaciones con arreglo de antenas, es decir señales electromagnética. En cambio la estructura *Householder* al emplear los reflectores del mismo nombre presentan un menor costo computacional. Con el algoritmo CG se tiene un costo de $3N^2 + N(10 - 4p) + p^2 - 8p + 2$ en la estructura *Householder* versus a la estructura GSC que tiene un costo computacional igual a $4N^2 + N(11 - 7p) + 3p^2 - 10p + 1$ y a la estructura *Constrained* que presenta un costo computacional mayor de $4N^2 + N(4p + 10) + 1$, por ello es que el algoritmo HCCG es deseable cuando su implementación es en un hardware con mayores recursos.

Debido al rápido desarrollo de la electrónica sobre todo en el campo del VLSI el más idóneo para su implementación es la estructura *Householder* con el algoritmo CG.

El algoritmo HCCG presenta un alto desempeño y esto se puede corroborar cuando se compara el *beam pattern* de este algoritmo con el algoritmo MUSIC, como se puede observar en la Figura 58, en la cual se tiene que la ganancia del algoritmo HCCG es aceptable y aunque su lóbulo principal no es altamente directivo en comparación al algoritmo MUSIC cumple con las condiciones del sistema.

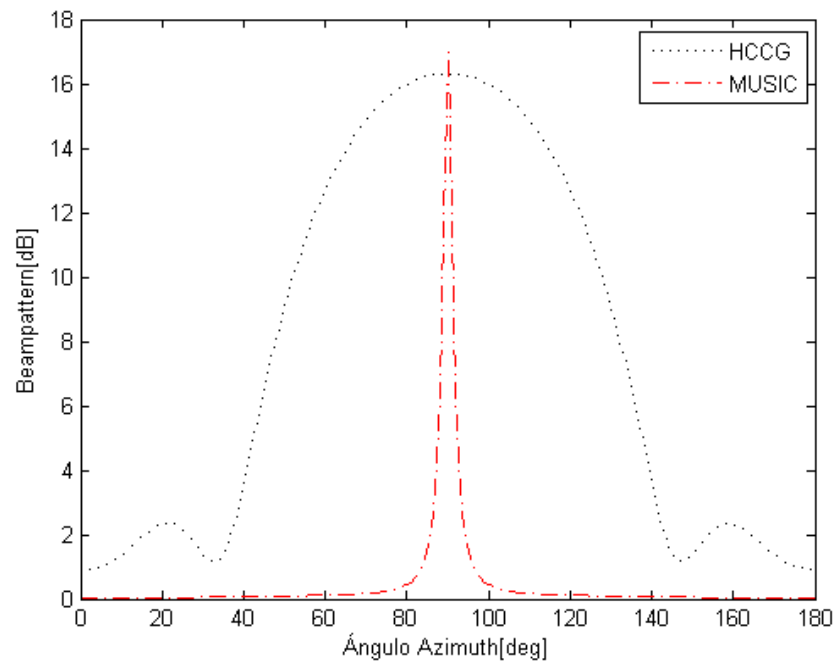


Figura 58. Comparación de los algoritmos HCCG y MUSIC respecto al *beam pattern* para un arreglo semiesférico.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Si bien la teoría de arreglo de sensores indica que estos sensores deben ser isotrópicos se pudo confirmar que, esto nunca es así en casos prácticos. Para la presente investigación los micrófonos no presentan un comportamiento similar y esto es propio de su fabricación, de las propiedades físico-químicas de los materiales empleados, por lo que se aproximó su comportamiento mediante la ecualización de los mismos tomando como referencia el *Mic 1*.
- En la estructura *Constrained* se tuvo problemas al utilizar la matriz de restricciones **C**, ya que los algoritmos presentaban un valor en el MSE muy alto por lo que se supuso que los algoritmos al converger se encontraban en un mínimo local, puesto que al hacer cambios en sus respectivos parámetros de adaptación no se obtenía ningún cambio y se optó por precondicionar la estructura de la matriz **C** mediante la utilización de la matriz *toeplitz*. Los resultados que se obtuvieron con la matriz restricciones **C** precondicionada fueron buenos, ya que los algoritmos alcanzaron un valor del MSE aceptable.
- La nueva estructura de la matriz de restricciones **C** tienen como desventada el aumento en el costo computacional de $O[N^2]$ multiplicaciones, por tanto el costo computacional en la estructura *Constrained* aumento siendo con esta nueva matriz más cara computacionalmente que la estructura *Householder*.
- Al comparar las tres estructuras: *Constrained*, GSC y *Householder* algoritmo por algoritmo se obtuvo que la estructura GSC y *Householder* presentan una respuesta bastante próxima en su MSE a pesar de que el comportamiento de

los micrófonos no son parecidos pero la estructura *Constrained* posee una respuesta diferente en comparación a las otras dos estructuras debido a que no existen condiciones iniciales iguales entre los algoritmos. Además de que debido al alto costo computacional que implica el uso de una matriz de *blocking* y que la matriz \mathbf{C} fue preconditionada.

- El algoritmo RLS presenta un buen desempeño pero un alto costo computacional y problemas de divergencia.
- El algoritmo CG posee un alto desempeño en comparación al algoritmo RLS, pero con la desventaja de que dependen de dos parámetros de adaptación en comparación a los otros algoritmos estudiados en la presente investigación que dependen de un parámetro de adaptación.
- Al variar el número de *taps* se obtuvo diferentes respuestas en los algoritmos LCMV pero se utiliza un número de *taps* igual a 10 porque se obtiene un equilibrio en el valor del MSE y la resolución en los *beam patterns*.

5.2. Recomendaciones

- Para seguir con el estudio del relace de la voz se debe incluir el estudio de precisión finita para confirmar el desenvolvimiento de la estructura *Householder*.
- Se recomienda ampliar el estudio de la matriz \mathbf{C} para las estructuras GSC y *Householder*.
- Se recomienda ampliar el estudio de las señales acústicas en otros ambientes ya que en la presente investigación se utiliza señales del tipo AWGN pero sin considerar las condiciones de altura y se trabajó bajo la referencia de un ambiente a nivel del mar y una temperatura promedio de 20°C.

BIBLIOGRAFÍA

- Apolinário, J., de Campos, M., & Bernal, C. (2000). The Constrained Conjugate Gradient Algorithm. *IEEE SIGNAL PROCESSING LETTERS*, VOL. 7, NO. 12,.
- Papoulis , A., & Pillai , U. (2002). *Probability, Random Variables and Stochastic Processes*. International Edition.
- Apolinário, J. A. (2009). *QDR-RLS Adaptive Filtering*. Springer.
- Apolinario, J., & de Campos, M. (2011). *Instituto Militar de Engenharia*. Acesso em 21 de 12 de 2014, disponível em <http://aquarius.ime.eb.br/~apolin>
- Benesty, J., Chen, J., & Huang, Y. (2008). *Microphone Array Signal Processing*. USA: Springer-Verlag Berlin Heidelberg.
- Bernal Oñate, C. (2000). *Principios y Aplicaciones de CDMA "Code Division multiple access" con implementación de algoritmos para la detección de multiusuarios*. Tesis , Quito.
- Chandran, S. (2004). *Adaptive Antenna Arrays Trends and Applications*. Springer-Verlag Berlin Heidelberg.
- Chandran, S. (2004). *Adaptive Antenna Arrays: Trends and Applications*. Malaysia: Springer.
- de Campos, M., Werner, S., & Apolinário, J. (September 2002). Constrained Adaptation Algorithms Employing Householder Transformation. *IEEE TRANSACTIONS ON SIGNAL PROCESSING VOL. 50, NO. 9, 9*.
- Diniz, P. (2013). *Adaptive Filtering: Algorithms and Practical Implementation* (Fourth ed.). Rio de Janeiro, Brazil: Springer New York Heidelberg Dordrecht London.
- Golub, G., & Van Loan, C. (2013). *Matrix Computations Fourth Edition*. The Johns Hopkins University Press.
- Gundersen, K., & Hakon Husoy, J. (2006). Preconditioner structures for the CLMS adaptive filtering algorithm. *IEEE*.
- Huang, Z., & Balanis, C. (2005). Adaptive Beamforming Using Spherical Array. *IEEE*.
- Liu, W., & Weiss, S. (2010). *WIDEBAND BEAMFORMING: Concepts and Techniques*. John Wiley & Sons, Ltd.
- Madisetti, V. (2010). *The Digital Signal Processing Handbook* (SECOND ed.). Taylor and Francis Group, LLC.

- Monzingo, R., Haupt, R., & Miller, T. (2011). *Introduction to Adaptive Arrays* (Second ed.). Published by SciTech Publishing, Inc.
- Rabiner, L., & R. W., S. (2009). *Theory and Application of Digital Speech Processing*.
- Valle, S. d. (2009). *Manual Prático de Acústica* (Terceira ed.). Rio de Janeiro : Música & Tecnologia.
- Van Veen, B., & Buckley, K. (2000). *Beamforming Techniques for Spatial Filtering. CRCnetBASE*.
- Werner, S., Apolinário, J., & Marcello L. R. , d. (s.d.). On the Equivalence of the Constrained RLS and the GSC-RLS Beamformers. *Helsinki University of Technology, Instituto Militar de Engenharia, and Universidade Federal do Rio de Janeiro*.
- Yang, W. Y., Cho, Y. S., & Choo, C. Y. (2012). *MATLAB /Simulink for Digital Signal Processing*. Seoul, Korea: Hongrung Publishing Company.