



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA-PROCESAMIENTO DIGITAL DE SEÑALES

NEURO-DETECTOR DE ÁNGULO DE ARRIBO DE SEÑALES DE VOZ

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
DAVID FRÓMETA RAMÍREZ

TUTOR
M. I. LARRY HIPÓLITO ESCOBAR SALGUERO
FACULTAD DE INGENIERÍA

MÉXICO, CD. MX. OCTUBRE 2023



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Jurado Asignado

Presidente: Dr. Savage Carmona Jesús
Secretario: Dr. Rascón Estebané Caleb A.
1er. Vocal: M.I Escobar Salguero Larry H.
2do. Vocal: Dr. Negrete Villanueva Marco A.
3er. Vocal: Dr. García Ugalde Francisco J.

NEURO-DETECTOR DE ÁNGULO DE ARRIBO DE SEÑALES DE VOZ

DAVID FRÓMETA RAMÍREZ

2023

Resumen

En este trabajo de tesis se presenta el diseño e implementación de un sistema capaz de estimar la dirección de arribo de una o varias señales de voz que inciden sobre un arreglo de micrófonos en un escenario acústico determinado. El sistema implementado combina características y el funcionamiento de dos métodos convencionales para la estimación de la dirección de arribo: Beamforming y Subespacios. Para integrar estos dos métodos el sistema incorpora una red neuronal del tipo Multilayer Perceptron, la cual, a partir del procesamiento de la matriz de correlaciones obtenidas de las muestras de la señal de voz, debe estimar las variaciones estadísticas y las características de la señal recibida para determinar el rango efectivo de la matriz y delimitar los subespacios de señal y solo-ruido necesarios para realizar la estimación de la dirección de arribo.

Agradecimientos

A todo el claustro de profesores del programa de Procesamiento Digital de Señales de la UNAM que incentivan la búsqueda profunda y la integración e implementación de conocimientos, convirtiéndolo esta maestría en una verdadera oportunidad de superación y crecimiento académico y profesional. En especial a mi tutor el M.I Larry H. Escobar Salguero por su guía en la realización de este trabajo.

Al CONAHCYT por haberme brindado apoyo financiero, sin el cual no hubiera podido venir a México y dedicarme a tiempo completo a este trabajo.

A mis amigos, que son los hombros de gigantes sobre los que me sostengo y veo qué tan lejos puedo llegar.

Por supuesto, a mi familia, que son mis pilares y la razón por la que pude estudiar, soñar y proponerme metas que cumplir.

Índice general

1. Introducción	1
1.1. Objetivo	1
1.2. Metodología	1
1.3. Orden de Trabajo	2
2. Metodología de Beamforming	3
2.1. Procesamiento de Arreglo de Señales	3
2.2. Beamforming y Filtrado Espacial	4
2.2.1. Aliasing Espacial y Lóbulos Gratinados	6
2.2.2. Estadística de Segundo Orden	7
2.2.3. Clasificación de Beamformers	7
2.3. Beamformer Independiente de los Datos	8
2.4. Beamformer Estadísticamente Óptimos	9
2.4.1. Cancelador Múltiple de Lóbulos Laterales	9
2.4.2. Empleo de Señal de Referencia	10
2.4.3. LCMV	10
2.4.4. GSC	13
2.4.5. Algoritmos Adaptables para Beamforming	14
2.5. Aplicaciones a la Estimación de DOA	16
2.5.1. Métodos convencionales de estimación DOA	17
2.5.2. LCMV para estimación DOA	17
3. Principios de Redes Neuronales.	19
3.1. Aspectos Generales de NN	19
3.1.1. Beneficios de las NNs	21
3.2. Arquitecturas de NN	21
3.2.1. NNs Multi-Capas Feedforward	22
3.2.2. RNN	22
3.3. Proceso de Aprendizaje	24
3.3.1. Aprendizaje Supervisado	24
3.3.2. Aprendizaje no Supervisado	25
3.4. Beamforming como Tarea de Aprendizaje	26
3.5. Entrenamiento de NNs para Estimación DOA	27
4. Diseño e Implementación del Sistema Neuro-Detector DOA	29
4.1. Generalidades del diseño del Neuro-Detector	29
4.2. Implementación del Bf y filtrado en el espacio de representación	31
4.3. Estimador DOA	32
4.4. Red Neuronal	33
4.4.1. MLP	33
4.4.2. Extracción de características. PCA	35
4.4.3. Entrenamiento y selección de parámetros	36
4.4.4. Análisis y selección de parámetros	37
4.5. Generación de los datos	38
4.5.1. Detección de actividad de voz	39
4.5.2. Simulación de los datos	39

5. Pruebas y Resultados	42
5.1. Neuro-Detector DOA para ULA	42
5.1.1. Pruebas de medio segundo de voz	43
5.1.2. Pruebas de segmentos completos de voz	45
5.1.3. Efectos del aumento de las reverberaciones y la distancia	48
5.2. Neuro-Detector DOA para UCA	50
5.2.1. Pruebas de medio segundo de voz	52
5.2.2. Pruebas de segmentos completos de voz	53
5.2.3. Efectos del aumento de las reverberaciones y la distancia	57
5.3. Efectos de las variaciones en la SNR	60
5.4. Comparación de los resultados	61
5.5. Tiempos de respuesta	64
5.6. Estimación DOA para varias fuentes incidentes	65
6. Conclusiones	68
A. Anexos	69
A.1. Características del hardware del sistema de cómputo	69
A.2. Tiempo de respuesta para 12 segundos de voz.	70
A.3. Comparación etiquetas DOAs y reultados del sistema.	70
Acrónimos	73

Índice de figuras

2.1.	Sistema de arreglo de cuatro micrófonos.	4
2.2.	Arreglos de beamforming. (a) Banda angosta. (b) Banda ancha.	5
2.3.	Ejemplo de lóbulos gratinados.	7
2.4.	Cancelador Múltiple de Lóbulos Laterales (MSC).	9
2.5.	Estructura Bf de banda ancha basada en una señal de referencia.	10
2.6.	Respuesta o beampattern (BP) de dos LCMV Bf sobre la banda de frecuencia $[0.25\pi \ \pi]$ con respecto al ángulo DOA.	13
2.7.	La transformación del beamformer LCMV restringido en un beamformer con estructura GSC no restringido.	14
2.8.	Patrón de haces de una columna de \mathbf{C}_n . (a) Muestra una respuesta cero en $\theta = -40^\circ$. El bloqueo de las direcciones $\theta = 10^\circ, -30^\circ$ y 40° se muestra en (b)	15
2.9.	Estructura del problema estándar de filtrado adaptivo	16
2.10.	La curva muestra la potencia de salida en función del DOA.	18
3.1.	NN multicapas y la retropropagación	20
3.2.	a) NN feedforward con una capa de neuronas. b) NN feedforward completamente conectada con una capa oculta y una capa de salida	22
3.3.	NN recurrente sin retroalimentación propia ni capas ocultas	23
3.4.	NN recurrente con una capa oculta	23
3.5.	Diagrama en bloques del aprendizaje con maestro. Las partes de verde en la figura constituyen el lazo de retroalimentación.	24
3.6.	Diagrama de bloques del aprendizaje reforzado. El entorno forma parte del lazo de retroalimentación.	25
3.7.	Diagrama de bloques del aprendizaje no supervisado	26
3.8.	Diagrama de bloques del Neuro-GSC	27
4.1.	Diagrama de bloques general del Neuro-Detector	30
4.2.	Bloque funcional 1, correspondiente al Bf y al filtrado en el espacio de representación.	31
4.3.	Bloque funcional 3, correspondiente al estimador DOA	32
4.4.	Bloque funcional 2, correspondiente a la red neuronal.	33
4.5.	Gráfico de la MLP para el caso de alimentar eigenvectores de la matriz \mathbf{R}_{xx}	34
4.6.	Gráfico de la MLP para el caso de alimentar eigenvalores de la matriz \mathbf{R}_{xx}	35
4.7.	Desempeño y razón de aprendizaje para ambas opciones de datos de entrada	37
4.8.	Desempeño y número de épocas para ambas opciones de datos de entrada	37
4.9.	Desempeño para un mismo número de épocas y diferentes razones de aprendizaje.	38
4.10.	Respuesta al impulso de la habitación para una habitación simulada.	40
4.11.	Ejemplo de escenario simulado para la creación del dataset de audio.	40
5.1.	Posiciones en 2D del dataset generado para la validación ULA	44
5.2.	Algunos resultados producidos por el sistema para el dataset de 2707 ejemplos etiquetados ULA	44
5.3.	Dataset alimentado y el desempeño del sistema ULA.	45
5.4.	Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 37° . ULA	46
5.5.	Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 62° . ULA	46

5.6. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde -80° . ULA	47
5.7. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde -26° . ULA	47
5.8. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde -2° . ULA	48
5.9. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 6° . ULA	48
5.10. Efecto del aumento en $rt60$ para una diatancia de un metro en la estimación DOA. ULA .	49
5.11. Señal de voz y RIR a la salida de un canal del ULA con el menor efecto de reverberaciones y distancia.	50
5.12. Efecto del aumento en $rt60$ para una diatancia de un metro en la estimación DOA. ULA .	51
5.14. Posiciones en 2D del dataset generado para la validación UCA	53
5.14. Algunos resultados producidos por el sistema para el dataset de 2707 ejemplos etiquetados UCA	53
5.15. Dataset alimentado y el desempeño del sistema. UCA.	54
5.16. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 2° . UCA	55
5.17. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 166° . UCA	55
5.18. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 201° . UCA	56
5.19. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 265° . UCA	56
5.20. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 300° . UCA	57
5.21. Contornos de potencias y estimación DOA de cada medio segundo dentro de un segmento completo de voz que arriba desde 359° . UCA	57
5.22. Efecto del aumento en $rt60$ para una diatancia fija de $1.5m$ en la estimación DOA. UCA .	58
5.23. Efecto del aumento en la distancia entre la fuente y el UCA para una $rt60$ fijo y moderado de $0.4s$	59
5.24. Respuestas del sistema sujeto a UCA para escenario extremo de $rt60$ y distancia.	60
5.25. Desempeño del sistema sujeto a ULA (a) y a UCA (b) bajo las mismas condiciones: altos valores de $rt60$ y distancia.	63
5.26. Tiempo consumido para estimar el DOA de medio segundo de voz.	64
5.27. Tiempo consumido para estimar el DOA de un segundo de voz.	64
5.28. Estimación DOA de dos fuentes de voz. Neuro-Detector sujeto a UCA.	66
5.29. Estimación DOA de tres fuentes de voz. Neuro-Detector sujeto a UCA.	67
A.1. Sistema de cómputo empleado en la implementación y pruebas del algoritmo.	69
A.2. Tiempo de respuesta del sistema para un archivo de audio de doce segundos de duración. .	70
A.3. Conjunto de resultados producidos por el sistema para el dataset de 2707 ejemplos etiquetados ULA	71
A.4. Conjunto de resultados producidos por el sistema para el dataset de 2707 ejemplos etiquetados UCA	72

Índice de tablas

4.1. Parámetros de la MLP para el caso de alimentar eigenvectores de la matriz \mathbf{R}_{xx}	34
4.2. Parámetros de la MLP para el caso de alimentar eigenvalores de la matriz \mathbf{R}_{xx}	35
5.1. Características del dataset generado para las pruebas de medio segundo ULA	43
5.2. Desempeño y valores de rendimiento del sistema. Dataset 2707 ejemplos. ULA	44
5.3. Desempeño y valores de rendimiento del sistema. Dataset 200 ejemplos. ULA	45
5.4. Intervalos considerados para las pruebas de segmentos completos ULA	46
5.5. Forma de medir el desempeño del Nuero-Detector sujeto a UCA	51
5.6. Características del dataset generado para las pruebas de medio segundo UCA	52
5.7. Desempeño y valores de rendimiento del sistema. Dataset 2707 ejemplos. UCA	53
5.8. Desempeño y valores de rendimiento del sistema. Dataset 200 ejemplos. UCA	54
5.9. Intervalos considerados para las pruebas de segmentos completos UCA	54
5.10. Sesgo de incertidumbre en la estimación DOA del sistema para los dos casos de arreglo considerados	61
5.11. Resumen de resultados del sistema para ambas geometrías de arreglo.	62
5.12. Tiempo de respuesta del sistema para estimar el DOA de medio segundo de voz.	64

Capítulo 1

Introducción

El procesamiento de arreglo de señales es una de las áreas dentro del procesamiento de señales que ha sido estudiada extensivamente en el pasado debido a sus amplias aplicaciones en varias áreas que abarcan desde radar, sonar, arreglos de micrófonos, radio astronomía, sismología, diagnóstico médico y tratamiento, hasta las comunicaciones [1]-[8]. Este involucra múltiples sensores (micrófonos, antenas, etc) localizados en posiciones diferentes en el espacio para procesar las señales recibidas provenientes desde diferentes direcciones.

Los sistemas de arreglos diseñados para recibir señales que se propagan espacialmente, a menudo encuentran la presencia de señales interferentes. Si la señal deseada y las interferentes ocupan la misma banda de frecuencia, entonces el filtrado en frecuencia no puede ser utilizado adecuadamente para separar señal de interferencia. Sin embargo, las señales deseadas e interferentes normalmente provienen de localizaciones espaciales diferentes. Estas separaciones en el espacio pueden aprovecharse para lograr separar señal de interferencia mediante filtrado espacial en el receptor.

El cerebro humano puede hacer esta separación y por ende el filtrado de forma natural ajustando las capacidades auditivas de los oídos en dependencia de la localización de la fuente sonora de interés. Mediante un arreglo de micrófonos una máquina o computadora puede implementar una metodología de filtrado espacial llamada Beamforming (BF), que emplea el conjunto de señales provenientes del arreglo para incrementar la selectividad espacial y la razón señal a ruido (Signal to Noise Ratio (SNR)) recibida [12]. Este tipo de sistema es capaz de filtrar señales provenientes de distintas direcciones o ángulos de arribo (Direction of Arrival (DOA)), lo cual se logra direccionando el haz del patrón de respuesta hacia la dirección deseada y los valores nulos hacia las direcciones de fuentes interferentes. Sin embargo para realizar BF la máquina necesita conocer con antelación el DOA de las fuentes de sonido, que a diferencia para los humanos, no es una tarea trivial.

Diversas técnicas para la estimación de DOA han sido investigadas, siendo las más conocidas la técnica basada en BF, la técnica basada en la proyección sobre subespacios y la técnica basada en la diferencia del tiempo de arribo [31]. Acercamientos más modernos involucran el uso de redes neuronales (Neural Network (NN)) y aprendizaje profundo [17],[18] para extraer información y estimar el DOA de una fuente sonora [19],[20] o varias fuentes [21],[22].

1.1. Objetivo

Diseñar e implementar un algoritmo de estimación DOA basado en una estructura Beamformer (Bf) que integre las cualidades de una red neuronal con bajo requerimiento computacional, para lograr que sea robusto en cuanto a la resolución de detección de ángulo de arribo y ante las variaciones de las señales de interés e interferentes.

1.2. Metodología

El sistema de detección de DOA de señales de voz que se propone en este trabajo de tesis está constituido por tres bloques modulares fundamentales. En el *primero*, las muestras de las señales de voz obtenidas a la salida del arreglo de micrófonos son procesadas y se obtiene la matriz de correlaciones entre ellas, a la vez que es establecido un Bf estadísticamente óptimo por cada posible dirección de

arriba considerada de la señal de interés. En el *segundo*, se emplea una NN para estimar las variaciones estadísticas y el ruido y estimar el subespacio de señal y el subespacio de solo-ruido. Una vez delimitados los subespacios el *tercer* bloque proyecta cada Bf en el subespacio de señal determinado por la NN y calcula y compara la potencia asociada a cada Bf, siendo la mayor un indicador de la mejor estimación DOA.

1.3. Orden de Trabajo

En el capítulo 2: se describe la metodología de BF y se definen las terminologías básicas, algunas clasificaciones, conceptos y aspectos de implementación de los Bf considerados para esta tesis. Se resalta y simulan resultados que refuerzan la viabilidad del BF para la estimación del DOA.

En el capítulo 3: se presenta una breve introducción a las redes neuronales, y se describen conceptos fundamentales de la arquitectura de las redes neuronales y los paradigmas de aprendizaje. Se abordan algunas de las ventajas de incluir NN en esta tesis y queda establecido el rol y la funcionalidad de la NN empleada.

En el capítulo 4: se describe el diseño seguido para la implementación del algoritmo Neuro-Detector. Se detalla cada bloque funcional y se explica la interrelación entre ellos y cómo están integrados en el sistema.

En el capítulo 5: se describen los resultados arrojados por el algoritmo Neuro-Detector para las diferentes geometrías empleadas en el arreglo de micrófonos . Se establecen comparaciones para validar y depurar los resultados.

En el capítulo 6: se presentan las conclusiones de este trabajo de tesis con base en los resultados obtenidos en el capítulo 5 y se describen brevemente algunas mejoras para incrementar la eficiencia del sistema implementado.

Capítulo 2

Metodología de Beamforming

El comienzo del BF como forma versátil de filtrado espacial está muy ligado al desarrollo de los arreglos de antenas, pero hace décadas es reconocido como uno de los pilares del área de procesamiento de arreglo de señales en general [8].

En la actualidad todos los dispositivos electrónicos de comunicaciones y otras aplicaciones tanto médicas como cotidianas desempeñan como función principal la recepción de señales para procesarlas y desempeñar una respuesta o función. Sin embargo, todos los ambientes o canales posibles en los que se obtienen dichas señales en la práctica están inmersos en ruido e interferencias. Si la señal de interés y las señales interferentes ocupan la misma banda de frecuencia, entonces mediante un filtrado en frecuencia ya sea analógico o digital, será muy difícil separar las señales interferentes de la de interés. La alternativa es emplear BF para hacer filtrado espacial aprovechando que las señales provienen de fuentes localizadas en posiciones diferentes. El beamforming brinda una forma versátil de filtrado espacial capaz de separar señales de interés en una dirección dada y rechazar las señales que provengan de otras direcciones. Sin embargo es necesario conocer de antemano las direcciones o ángulos desde donde arriban las señales.

En este capítulo se definen conceptos y terminologías básicas, se introducen las operaciones de BF y se abordan temas del filtrado espacial. Se hace un acercamiento sobre la representación de la matriz de covarianza para el arreglo de datos recibidos. Se definen algunas clasificaciones de Bf, se abordan las nociones para la estimación DOA basada en BF y se muestran y comentan algunos resultados de simulaciones hechas para este trabajo con fines de corroborar y seleccionar algoritmos para la implementación.

2.1. Procesamiento de Arreglo de Señales

Un arreglo de sensores involucra múltiples micrófonos localizados en posiciones diferentes en el espacio para procesar las señales recibidas, provenientes de diferentes direcciones. Un ejemplo de un sistema de arreglo compuesto de cuatro micrófonos con dos señales incidentes se muestra en la figura 2.1, donde la dirección de arribo (DOA) de las señales está caracterizada por dos parámetros: un ángulo de elevación θ y acimut φ . A todo lo largo de este trabajo se supone que los micrófonos o sensores tienen la misma característica y que son omnidireccionales o isotrópicos, es decir, que sus respuestas a una señal incidente son independientes a su ángulo DOA.

De acuerdo a la localización relativa de los micrófonos, los arreglos se clasifican en [8]:

- Arreglos de 1-dimensión (1-D) o arreglos lineales.
- Arreglos de 2-dimensiones (2-D) o arreglos planares.
- Arreglos de 3-dimensiones (3-D) o arreglos volumétricos.

Cada uno de ellos se puede subdividir en dos categorías:

- Espaciado regular, incluyendo los uniformes y no uniformes.
- Espaciado irregular o aleatorio.

El estudio realizado en este trabajo de tesis está basado en arreglos con espaciado regular. Para las señales de voz incidentes siempre se supone que son ondas planas, es decir, el arreglo está localizado en el campo lejano de la fuente que genera las ondas sonoras y que las señales recibidas tienen un frente de onda plano.

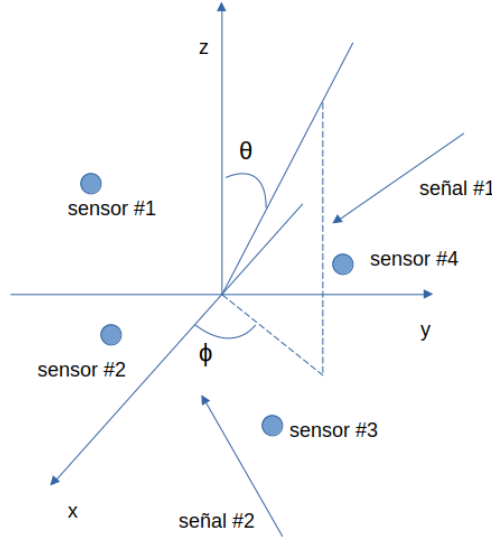


Figura 2.1: Sistema de arreglo de cuatro micrófonos.

2.2. Beamforming y Filtrado Espacial

La figura 2.2 ilustra dos Bf basados en un arreglo lineal de micrófonos. El primero muestrea los frentes de ondas en el espacio y es usado típicamente para procesar señales de banda angosta. Básicamente es una combinación lineal de los datos en los M micrófonos en un instante de tiempo dado. La salida del Bf en el instante k , $y(k)$, está dada por:

$$y(k) = \sum_{m=0}^{M-1} w_m^* x_m(k) \quad (2.1)$$

El segundo Bf en la figura 2.2 muestrea el frente de onda tanto temporal como espacialmente, y es usado cuando se utilizan señales de banda ancha, la salida en este caso puede ser expresada como:

$$y(k) = \sum_{m=0}^{M-1} \sum_{i=0}^{J-1} w_{m,i}^* x_m(k-i) \quad (2.2)$$

donde $J-1$ es el número de retardos que hay en cada uno de los M canales. Podemos emplear una notación que nos permita tratar ambos Bf en la figura 2.2 simultáneamente. De esta forma (2.1) y (2.2) pueden ser escritas como:

$$y(k) = \mathbf{w}^H \mathbf{x}(k) \quad (2.3)$$

definiendo apropiadamente un vector de pesos \mathbf{w} y un vector de datos \mathbf{x} . En este trabajo se usarán letras en minúscula y mayúscula negritas para denotar vectores y matrices respectivamente y el superíndice H para denotar transpuesta hermitiana. Los vectores se consideran vectores columna. Suponiendo que \mathbf{w} y $\mathbf{x}(\mathbf{k})$ son N dimensionales, esto implica que $N = MJ$ cuando se hace referencia a (2.2) y $N = M$ para (2.1). El índice de tiempo se obviará excepto más adelante cuando se aborde algoritmos adaptables. Entonces (2.3) es escrita como $y = \mathbf{w}^H \mathbf{x}$.

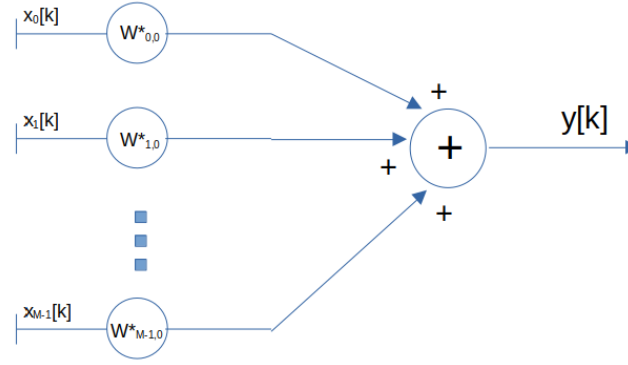
La respuesta en frecuencia de un filtro Finit Impulse Response (FIR) con coeficientes $w_m, 0 \leq m \leq M$ y retardo de T segundos está dada por:

$$r(\omega) = \sum_{m=0}^{M-1} w_m^* e^{-j\omega m T} \quad (2.4a)$$

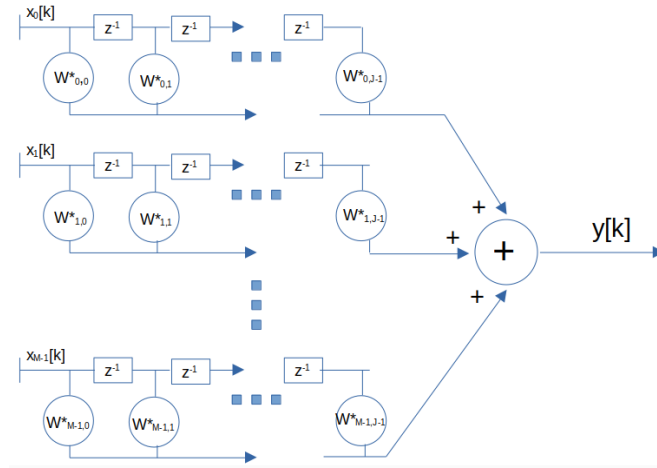
$$r(\omega) = \mathbf{w}^H \mathbf{d}(w) \quad (2.4b)$$

donde

$$\mathbf{w}^H = [w_1^* \ w_2^* \ \dots \ w_M^*]$$



(a) BF banda angosta



(b) BF banda ancha

Figura 2.2: Arreglos de beamforming. (a) Banda angosta. (b) Banda ancha.

y

$$\mathbf{d}(\omega) = [1 \ e^{j\omega T} \ e^{j\omega 2T} \ \dots \ e^{j\omega(M-1)T}]^H$$

$r(\omega)$ representa la respuesta del filtro a una señal senoidal compleja de frecuencia ω y $\mathbf{d}(\omega)$ es un vector que describe la fase de la senoidal en cada etapa (tap) del filtro FIR relativo al primer peso.

Similarmente la respuesta del Bf es definida como la amplitud y fase presentada ante una onda plana compleja como función de la frecuencia y la localización de la fuente. Suponiendo que la señal es una onda plana compleja con frecuencia ω y DOA θ y por conveniencia dejando que la fase de la onda en el primer micrófono es igual a cero. Entonces se tiene que $x_0(k) = e^{j\omega k}$ y $x_i(k) = e^{j\omega(k-\Delta\theta_m)}$, $0 \leq m \leq M-1$. $\Delta\theta$ representa el tiempo de retardo debido a la propagación desde el primer micrófono hasta el m_{esimo} micrófono. La sustitución en (2.2) resulta en la salida del Bf:

$$y(k) = e^{j\omega k} \sum_{m=0}^{M-1} \sum_{i=0}^{J-1} \mathbf{w}_m^* e^{-j\omega(\Delta\theta_m+m)} = e^{j\omega k} r(\theta, \omega) \quad (2.5)$$

donde $r(\theta, \omega)$ es la respuesta del Bf y puede escribirse de forma vectorial como:

$$r(\theta, \omega) = \mathbf{w}^H \mathbf{d}(\theta, \omega) \quad (2.6)$$

Los elementos de $\mathbf{d}(\theta, \omega)$ corresponden a la exponencial compleja $e^{-j\omega(\Delta\theta_m+m)}$ y de forma general pueden ser expresados como:

$$\mathbf{d}(\theta, \omega) = [1 e^{-j\omega\tau_1(\theta)} e^{-j\omega\tau_2(\theta)} \dots e^{-j\omega\tau_N(\theta)}]^T \quad (2.7)$$

donde los $\tau_i(\theta)$, $1 \leq i \leq N$ son los tiempos de retardo debido a la propagación entre micrófonos y a cualquier elemento de retardo desde el punto de referencia cero hasta el punto donde el $i_{ésimo}$ peso es aplicado. $\mathbf{d}(\theta, \omega)$ es referido como vector de direccionamiento o vector de dirección. Siendo más explícitos, para el caso de Bf de banda ancha (2.7) es equivalente a:

$$\mathbf{d}(\theta, \omega) = [1 \dots e^{-j\omega\tau_{M-1}(\theta)} e^{-j\omega(\tau_0(\theta)+Ts)} \dots e^{-j\omega(\tau_{M-1}(\theta)+Ts)} \dots e^{-j\omega(\tau_0(\theta)+(J-1)Ts)} \dots e^{-j\omega(\tau_{M-1}(\theta)+(J-1)Ts)}]^T \quad (2.8)$$

El patrón de haces (Beam Pattern (BP)) del Bf está definido como la magnitud al cuadrado de $r(\theta, \omega)$ y en db está definido como [9]:

$$BP = 20 \log_{10} \frac{|r(\theta, \omega)|}{\max |r(\theta, \omega)|} \quad (2.9)$$

La correspondencia entre el filtro FIR y BF es más cercana cuando el Bf opera en una sola frecuencia ω_0 y la geometría del arreglo es lineal y equiespaciado. Si llamamos al espacio entre micrófonos como d , la velocidad de propagación como c y θ el DOA relativo a la perpendicular al arreglo, tenemos $\tau_i(\theta) = i(d/c) \sin(\theta)$. En este caso la relación entre la frecuencia ω en $\mathbf{d}(\omega)$ (filtro FIR) y la dirección θ en $\mathbf{d}(\theta, \omega)$ (Bf) viene dada por: $\omega = \omega_0(d/c) \sin(\theta)$.

2.2.1. Aliasing Espacial y Lóbulos Gratinados

Basado en el vector de direccionamiento $\mathbf{d}(\theta, \omega)$ se aborda el problema de aliasing espacial encontrado en BF. En la conversión analógica-digital se muestrea la señal de tiempo continuo y se convierte en una secuencia de tiempo discreto, en el muestreo temporal el aliasing es referido como el fenómeno donde a partir de señales con diferentes frecuencias se obtienen secuencias discretas iguales, lo que ocurre cuando se muestrea a una frecuencia menor a la frecuencia de Nyquist. Para señales con aliasing temporal no será posible recuperar la señal original de tiempo continuo a partir de las muestras. En el procesamiento de arreglos y en BF los sensores o micrófonos muestrean espacialmente las señales incidentes, y si las señales incidentes desde diferentes DOA no son muestreadas lo suficientemente denso, dígame, el interespaciado de los micrófonos es muy grande, entonces las fuentes con localizaciones diferentes tendrán el mismo vector $\mathbf{d}(\theta, \omega)$ y no se podrá determinar de manera única los DOAs basado en las señales recibidas. Similar a la aliasing temporal tenemos un problema de aliasing espacial debido a la ambigüedad en las direcciones de arriba de las fuentes de señales.

Para señales con igual frecuencia angular ω y la correspondiente λ , pero diferentes DOAs θ_1 y θ_2 , tal que $(\theta_1, \theta_2) \in [-\pi/2, \pi/2]$, el aliasing espacial implica que $\mathbf{d}(\theta_1, \omega) = \mathbf{d}(\theta_2, \omega)$, es decir:

$$e^{-j\omega\tau_i(\theta_1)} = e^{-j\omega\tau_i(\theta_2)} \quad (2.10)$$

Para un arreglo lineal uniformemente espaciado, con un espacio inter-elemento d , tenemos que el tiempo de retardo entre micrófonos es $\tau_m = m\tau_1 = m(d/c) \sin(\theta)$ y $\omega\tau_m = m(2\pi d \sin(\theta))/\lambda$, entonces (2.11) cambia a:

$$e^{-jm(2\pi d \sin(\theta_1))/\lambda} = e^{-jm(2\pi d \sin(\theta_2))/\lambda} \quad (2.11)$$

Para evitar el aliasing la condición $|2\pi d \sin(\theta)/\lambda|_{\theta=\theta_1, \theta_2} < \pi$ debe ser satisfecha. Entonces tenemos $|d \sin(\theta)/\lambda| < 1/2$. Dado que $|\sin(\theta)| \leq 1$ esto requiere que $d \leq \lambda/2$.

Suponieno un arreglo de seis micrófonos con pesos con amplitud uno y fase cero, el BP se obtiene de (2.9). Graficando el resultado de las simulaciones hechas para esta tesis con $d = 0.5\lambda, 0.6\lambda, 2\lambda$ en la expresión (2.9) se obtiene los patrones de haces que se muestran en la figura 2.3. Se aprecia que a medida que el espacio entre micrófonos se hace mayor, el lóbulo principal se hace más angosto y, cuando $d > 0.5\lambda$, lóbulos adicionales aparecen con una cantidad creciente de energía en ellos a medida que aumenta d . Estos lóbulos adicionales debido al aliasing son referidos como lóbulos gratinados [1].

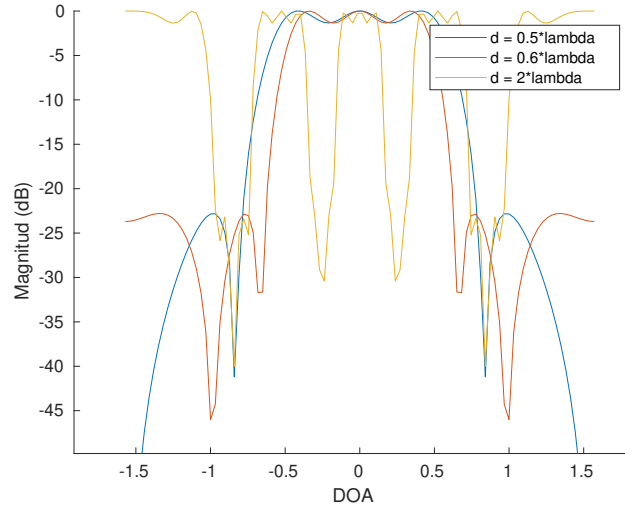


Figura 2.3: Ejemplo de lóbulos gratinados.

2.2.2. Estadística de Segundo Orden

La evaluación del desempeño del Bf involucra la potencia o varianza, de este modo la estadística de segundo orden juega un papel importante. Este trabajo como sugiere su título esta enfocado en señales de voz, por tanto a lo largo de este se asume que los datos recibidos por los micrófonos son de media cero. La varianza o potencia de la salida de un Bf está dada por:

$$E \{ |y|^2 \} = \mathbf{w}^H E \{ \mathbf{x}\mathbf{x}^H \} \mathbf{w} \quad (2.12)$$

Si los datos son estacionarios en sentido amplio la matriz de covarianza es independiente del tiempo y está dada por:

$$\mathbf{R}_{\mathbf{x}\mathbf{x}} = E \{ \mathbf{x}\mathbf{x}^H \} \quad (2.13)$$

A pesar que frecuentemente se encuentren datos no estacionarios, la suposición de estacionalidad en sentido amplio es usada en el desarrollo de Bf estadísticamente óptimos y en la evaluación del desempeño de estado estable del Bf.

Suponiendo que \mathbf{x} representa las muestras de una serie de tiempo muestreada uniformemente, con densidad de potencia espectral $S(\omega)$ y con energía nula fuera de la banda $[\omega_a, \omega_b]$. $\mathbf{R}_{\mathbf{x}\mathbf{x}}$ se puede expresar en términos de la densidad de potencia espectral del arreglo de datos usando la relación con la transformada de Fourier como:

$$\mathbf{R}_{\mathbf{x}\mathbf{x}} = \frac{1}{2\pi} \int_{\omega_a}^{\omega_b} S(\omega) \mathbf{d}(\omega) \mathbf{d}^H(\omega) d\omega \quad (2.14)$$

con $\mathbf{d}(\omega)$ como se definió en (2.4b). Ahora suponiendo un arreglo de datos \mathbf{x} debido a una fuente con DOA θ ; de igual manera que la serie de tiempo se puede obtener la matriz de covarianza del arreglo de datos como:

$$\mathbf{R}_{\mathbf{x}\mathbf{x}} = \frac{1}{2\pi} \int_{\omega_a}^{\omega_b} S(\omega) \mathbf{d}(\theta, \omega) \mathbf{d}^H(\theta, \omega) d\omega \quad (2.15)$$

2.2.3. Clasificación de Beamformers

Bf pueden ser clasificados ya sea como independientes de los datos o como estadísticamente óptimos, dependiendo de cómo sean elegidos los pesos en la combinación lineal. Los pesos en un Bf independiente de los datos no dependen del arreglo de datos y son elegidos para presentar una respuesta específica para todos los escenarios señal/interferencia. Los pesos en un Bf estadísticamente óptimos son elegidos basado en la estadística del arreglo de datos para “optimizar” la respuesta del Bf de modo que se minimicen las

contribuciones de señales interferentes provenientes de direcciones diferentes a las de interés, maximizando así la relación señal-ruido [12].

2.3. Beamformer Independiente de los Datos

En numerosas aplicaciones es de interés aplicar diseños de Bf que aproximen una respuesta arbitraria. Por ejemplo, se pudiera desear recibir cualquier señal arribando desde un intervalo de direcciones, en dicho caso la respuesta es unitaria sobre todo el intervalo de direcciones. Como otro ejemplo, pudiera conocerse que existe una fuente de interferencia arribando desde un cierto intervalo de direcciones, en dicho caso la respuesta es cero en dicho intervalo.

Estos dos casos son análogos a filtros FIR pasabandas y supresor de bandas respectivamente. En el diseño se considera seleccionar \mathbf{w} tal que la respuesta real $r(\theta, \omega) = \mathbf{w}^H \mathbf{d}(\theta, \omega)$ de modo que se minimice la norma ponderada L_p de la diferencia entre la respuesta deseada y la real. La norma ponderada L_p se utiliza en diversas técnicas establecidas de diseño de filtros FIR. Las normas más usadas son L_α (minmax) y L_2 (mínimos cuadrados o least squares).

Las técnicas específicas incluyen [12]:

1. Aplicar la técnica de windowing a la respuesta pulso unitario de un filtro ideal. (minimiza la norma L_2 sobre un intervalo continuo de ω)
2. Muestreo de la respuesta en frecuencia y ponderación lineal de mínimos cuadrados (minimiza L_2 sobre un intervalo discreto de ω)
3. Diseño minmax con algoritmos basados en Remez (minimiza la norma L_α sobre un intervalo discreto de ω)
4. Diseño minmax de respuesta compleja y amplitud de respuesta (minimiza la norma L_α sobre un intervalo discreto de ω)

El diseño de filtros FIR corresponde a un problema de aproximación polinomial dado que la respuesta en frecuencia (2.4b) es la transformada discreta de Fourier de la secuencia de coeficientes del filtro FIR.

Excluyendo los casos donde la geometría del Bf puede ser reducida a lineal y equiespaciada, el diseño de un Bf no es un problema de aproximación polinomial. En general la respuesta del Bf en (2.6) es una suma ponderada de exponenciales elevadas a potencias no enteras. De este modo los métodos L_α (3 y 4) no son aplicables ya que están basados en aproximación polinomial. El método de windowing (1) tampoco, ya que se basa en la transformada discreta de Fourier. Sin embargo el procedimiento L_2 (2) que emplea mínimos cuadrados ponderados linealmente es aplicable.

Para el diseño de un Bf independiente de los datos vía la optimización de L_2 se considera la minimización del error al cuadrado de la diferencia entre la respuesta deseada y la real en P puntos (θ_i, ω_i) , $1 \leq i \leq P$, si $P < N$ entonces se obtiene un problema sobredeterminado de mínimos cuadrados:

$$\min_{\mathbf{w}} \left| \mathbf{A}^H \mathbf{w} - \mathbf{r}_d \right|^2 \quad (2.16)$$

donde

$$\mathbf{A} = [\mathbf{d}(\theta_1, \omega_1) \ \mathbf{d}(\theta_2, \omega_2) \ \cdots \ \mathbf{d}(\theta_P, \omega_P)] \quad (2.17)$$

$$\mathbf{r}_d = [r_d(\theta_1, \omega_1) \ r_d(\theta_2, \omega_2) \ \cdots \ r_d(\theta_P, \omega_P)]^H \quad (2.18)$$

Garantizando que $\mathbf{A}\mathbf{A}^H$ sea inversible, es decir que \mathbf{A} tenga rango completo, entonces la solución de (2.16) viene dada por:

$$\mathbf{w} = \mathbf{A}^+ \mathbf{r}_d \quad (2.19)$$

donde $\mathbf{A}^+ = (\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{A}$ es la pseudo inversa de \mathbf{A} . Aproximaciones de rango pequeño deben ser usadas para \mathbf{A} y \mathbf{A}^+ siempre que el rango sea menor que N . Esto también asegura una norma pequeña de \mathbf{w} , lo cual es importante pues $\mathbf{w}^H \mathbf{w}$ representa la ganancia de ruido blanco.

2.4. Beamformer Estadísticamente Óptimos

En los Bf estadísticamente óptimos los pesos son elegidos basados en la estadística del arreglo. De forma general se puede decir que la meta del Bf de este tipo es optimizar la respuesta minimizando las contribuciones de señales interferentes provenientes de direcciones diferentes a las de interés a través de la ubicación de los valores nulos en estas direcciones; maximizando así la relación señal-ruido.

2.4.1. Cancelador Múltiple de Lóbulos Laterales

El Cancelador Múltiple de Lóbulos Laterales (Multi-Sidelobes Canceller (MSC)) es quizá el primero de los Bf estadísticamente óptimos. Un MSC consiste de un canal principal y uno o varios canales auxiliares como se muestra en la figura 2.4. El canal principal puede ser un Bf fijo, el cual debe tener una respuesta altamente direccional, “apuntando” en la dirección de la señal de interés. Las señales interferentes se suponen que entran a través de los lóbulos laterales del canal principal y además son recibidas por los canales auxiliares. La meta es encontrar los pesos de los canales auxiliares para cancelar los componentes de interferencia del canal principal, por ende las respuestas de ambos canales a las interferencias deben ser idénticas, esto se muestra en la figura 2.4a. En 2.4b se muestra las salidas del canal principal, la rama auxiliar y del sistema general ante interferencias con DOA θ_i .

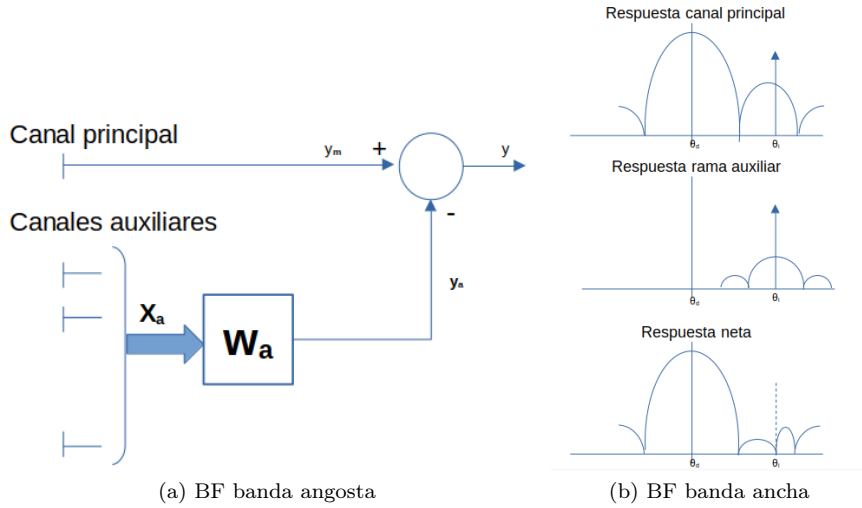


Figura 2.4: Cancelador Múltiple de Lóbulos Laterales (MSC).

Tener respuesta cero para todas las señales interferentes es imposible o bien resulta en una ganancia de ruido blanco muy alta. De esta forma al determinar los pesos mediante la minimización del valor esperado de la potencia de salida se crea un compromiso entre supresión de interferencias y ganancia de ruido blanco.

Definiendo el arreglo de datos obtenido de los canales auxiliares como \mathbf{x}_a y los datos primarios como y_m , entonces la salida del MSC vendrá dada por:

$$y = y_m - \mathbf{w}_a^H \mathbf{x}_a \quad (2.20)$$

De este modo y como se mencionó anteriormente el criterio de selección de los pesos es la minimización del valor esperado de la potencia de salida del MSC dado por:

$$\min_{\mathbf{w}} E \left\{ |y_m - \mathbf{w}_a^H \mathbf{x}_a|^2 \right\} \quad (2.21)$$

de donde se desprende que los pesos óptimos están dados por:

$$\mathbf{w}_a = \mathbf{R}_a^{-1} \mathbf{r}_{ma} \quad (2.22)$$

donde $\mathbf{R}_a = E \{ \mathbf{x}_a \mathbf{x}_a^H \}$ es la matriz de covarianza de los datos auxiliares y $\mathbf{r}_{ma} = E \{ \mathbf{x}_a y_m^* \}$ es el vector de correlación cruzada de los datos del canal principal y los canales auxiliares. Aquí es importante notar, que elegir los pesos para minimizar la potencia de salida puede causar la cancelación de la señal

deseada, ya que también contribuye a la potencia de salida. Más aún, a medida que se fortalece la señal deseada contribuye en mayor medida a la potencia total de salida y se incrementa el porcentaje de cancelación. El MSC es muy efectivo para casos en que la señal deseada es muy débil en comparación a las interferencias.

Una de las ventajas del MSC es su simplicidad, mientras que sus desventajas estriban en que requiere la ausencia de la señal deseada en los canales auxiliares para determinar los pesos.

2.4.2. Empleo de Señal de Referencia

Si la señal deseada fuera conocida, los pesos pueden ser elegidos para minimizar el error entre la salida del Bf y la señal deseada. En este caso el Bf es realizado usando un filtro adaptativo multicanal (Multi-Channel Adaptive Filter (MCAF)), como se muestra en la figura 2.5, donde las M señales son alimentadas al MCAF y sus coeficientes o pesos son ajustados mediante la minimización de una función de costo basada en este caso en el error cuadrático medio (Mean Square Error (MSE)). La longitud de los filtros adaptables en cada uno de los M canales es J y por tanto existen MJ coeficientes. Tenemos:

$$\begin{aligned} e[k] &= y_d[k] - y[k] \\ &= y_d[k] - \mathbf{w}^H \mathbf{x}[k] \end{aligned} \quad (2.23)$$

$$\underset{\mathbf{w}}{\text{mín}} = E \{ e[k] e^*[k] \} \quad (2.24)$$

donde los pesos óptimos están dados por:

$$\mathbf{w} = \mathbf{R}_x^{-1} \mathbf{r}_{xd} \quad (2.25)$$

siendo $\mathbf{R}_x = E \{ \mathbf{x}[k] \mathbf{x}^H[k] \}$ la matriz de covarianza del arreglo de datos del Bf y $\mathbf{r}_{xd} = E \{ \mathbf{x}[k] y_d^*[k] \}$ el vector de correlación cruzada del arreglo de datos del Bf y la señal deseada.

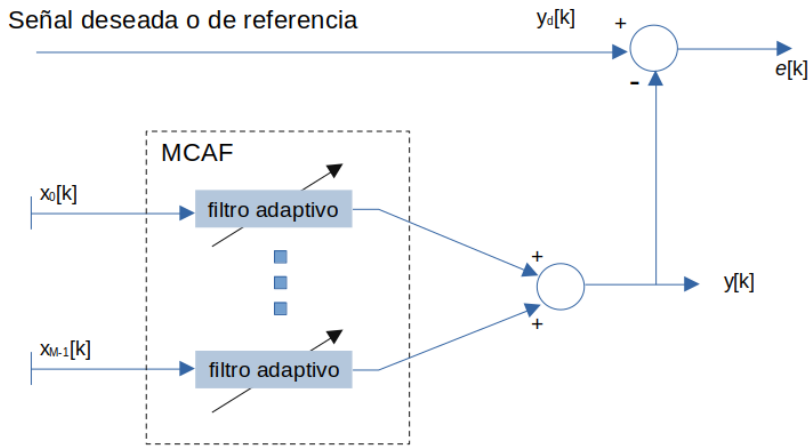


Figura 2.5: Estructura Bf de banda ancha basada en una señal de referencia.

Es claro que el conocimiento de la señal deseada elimina la necesidad de BF, sin embargo, brinda un punto de partida y de base para revisar los algoritmos adaptables aplicados a BF en la Sección 2.4.5. Además para algunas aplicaciones es posible tener suficiente conocimiento de la señal deseada para crear una señal similar que la represente. El hecho que no sea necesario conocer el DOA de la señal deseada es una ventaja de la implementación de esta estructura basada en una referencia.

2.4.3. Beamformer de Mínima Varianza con Restricciones Lineales

En esta subsección se aborda cómo las restricciones lineales se pueden emplear para controlar la respuesta de un Bf. La idea básica detrás del Bf de mínima varianza con restricciones lineales (Linearly Constrained Minimum Variance (LCMV)) [15] es restringir la respuesta, de modo que las señales desde la dirección de interés son pasadas con una ganancia y fase determinadas. Los pesos son elegidos para minimizar la varianza o potencia de salida, sujetos a las restricciones impuestas. Esto tiene el efecto de

preservar la señal deseada mientras que se minimizan las contribuciones a la salida de interferencias y ruidos desde otras direcciones distintas a la de interés.

En la sección 2.2 se abordó que la respuesta de un Bf a una fuente con ángulo DOA θ y frecuencia ω está dada por $\mathbf{w}^H \mathbf{d}(\theta, \omega)$. Para asegurar que cualquier señal que tenga frecuencia ω_0 y ángulo DOA θ_0 pase el Bf con una respuesta específica G_0 (G_0 es un número complejo) se puede establecer esta restricción como:

$$\mathbf{w}^H \mathbf{d}(\theta_0, \omega_0) = G_0 \quad (2.26)$$

La minimización de las contribuciones a la respuesta del Bf debido a interferencias (señales con frecuencia ω_0 pero que no arriban desde θ_0) se logra mediante la minimización del valor esperado de la varianza o potencia de salida $E\{y[k]y^*[k]\} = E\{|y[k]|^2\} = \mathbf{w}^H \mathbf{R}_x \mathbf{w}$. El problema del LCMV viene dado por la selección de los pesos como:

$$\underset{\mathbf{w}}{\text{mín}} \mathbf{w}^H \mathbf{R}_x \mathbf{w} \quad \text{suje}to \ a \quad \mathbf{d}^H(\theta_0, \omega_0) \mathbf{w} = G_0^* \quad (2.27)$$

Para resolver 2.27 puede usarse el método de Lagrange, resultando en el vector de pesos óptimo:

$$\mathbf{w}_{opt} = G_0^* \frac{\mathbf{R}_x^{-1} \mathbf{d}(\theta_0, \omega_0)}{\mathbf{d}^H(\theta_0, \omega_0) \mathbf{R}_x^{-1} \mathbf{d}(\theta_0, \omega_0)} \quad (2.28)$$

donde la presencia de ruido blanco incorrelacionado debe asegurar que \mathbf{R}_x^{-1} exista. El Bf denominado de mínima varianza con respuesta sin distorsión (Minimum Variance Distortionless Response (MVDR)) se obtiene de (2.28) haciendo $G_0 = 1$.

Para generalizar (2.27) de forma sencilla a múltiples restricciones lineales y tener más control sobre el BP, supongamos que se conoce que hay una interferencia fija en la dirección conocida ϕ_0 , y por tanto se desea que en ese DOA la respuesta sea cero mientras se mantiene la respuesta G_0 a la señal deseada. Esto se expresa como:

$$\begin{bmatrix} \mathbf{d}^H(\theta_0, \omega_0) \\ \mathbf{d}^H(\phi_0, \omega_0) \end{bmatrix} \mathbf{w} = \begin{bmatrix} G_0^* \\ 0 \end{bmatrix} \quad (2.29)$$

Si existen $r < N = MJ$ restricciones lineales independientes impuestas a \mathbf{w} podemos formular las restricciones de forma matricial como:

$$\mathbf{C}^H \mathbf{w} = \mathbf{f} \quad (2.30)$$

donde la matriz \mathbf{C} de dimensión $MJ \times r$ es llamada matriz de restricción y \mathbf{f} , de dimensión r es el vector respuesta. Las restricciones se suponen linealmente independientes de modo que la matriz \mathbf{C} tiene rango r . El problema del LCMV con esta ecuación de restricción más general viene dado por:

$$\underset{\mathbf{w}}{\text{mín}} \mathbf{w}^H \mathbf{R}_x \mathbf{w} \quad \text{suje}to \ a \quad \mathbf{C}^H \mathbf{w} = \mathbf{f} \quad (2.31)$$

$$\mathbf{w}_{opt} = \mathbf{R}_x^{-1} \mathbf{C} \left[\mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \right]^{-1} \mathbf{f} \quad (2.32)$$

Diseño de Restricciones por Eigenvectores

La matriz de restricción se puede formular simplemente muestreando la banda de frecuencia de interés de la señal y haciendo que el vector respuesta en esos puntos tome los valores deseados, usualmente estos son retardos puros o ceros si queremos anular esta señal. Este acercamiento es conocido como restricciones lineales de múltiples puntos y puede ser extendido facilmente a varias direcciones de fuentes con diferentes anchos de banda [29]. Mientras más puntos de muestras tengamos en la formulación de restricciones mejor será el control sobre la respuesta del Bf sobre esas direcciones y frecuencias. Sin embargo una matriz directamente formada de esta forma no es la manera más eficiente de utilizar los grados de libertad del arreglo por lo que una versión modificada, por lo que una restricción por eigenvectores, basada en una representación de rango pequeño de señales de banda ancha, fue desarrollada por [11]; la cual es una extensión al diseño propuesto para bloquear eficientemente señales interferentes [30].

Si queremos formar r restricciones linealmente independientes para la dirección θ , es decir, \mathbf{C} tendrá rango r y la banda de frecuencia de interés es $\omega \in [\omega_{min} \ \omega_{max}]$, primero muestreamos la banda de frecuencia en $\hat{r} \gg r$ puntos de frecuencia $\omega_i, i = 0, 1, \dots, \hat{r} - 1$. Los valores desados de la respuesta del Bf a los puntos de frecuencia ω_i en la dirección θ están dados por $G(\theta, \omega_i)$. Con los vectores de direccionamiento dados como (2.8) las restricciones para \hat{r} puntos de frecuencia se expresan como [9]:

$$[\mathbf{d}(\theta, \omega_0) \ \mathbf{d}(\theta, \omega_1) \ \dots \ \mathbf{d}(\theta, \omega_{\hat{r}-1})]^H \mathbf{w} = \begin{bmatrix} G^*(\theta, \omega_0) \\ G^*(\theta, \omega_1) \\ \vdots \\ G^*(\theta, \omega_{\hat{r}-1}) \end{bmatrix} \quad (2.33)$$

Esto es la formulación de restricción por múltiples puntos y la matriz de restricción \mathbf{C} y el vector respuesta \mathbf{f} están dados por:

$$\mathbf{C} = [\mathbf{d}(\theta, \omega_0) \ \mathbf{d}(\theta, \omega_1) \ \dots \ \mathbf{d}(\theta, \omega_{\hat{r}-1})] \quad (2.34)$$

y

$$\mathbf{f} = [G^*(\theta, \omega_0) \ G^*(\theta, \omega_1) \ \dots \ G^*(\theta, \omega_{\hat{r}-1})]^T \quad (2.35)$$

Como segundo paso, la matriz es descompuesta en tres matrices mediante una operación de descomposición de valores singulares (Singular Value Decomposition (SVD)) [4] dada por:

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \quad (2.36)$$

donde $\mathbf{\Sigma}$ contiene los valores singulares de \mathbf{C} .

Para encontrar una matriz \mathbf{C}_r que sea una aproximación de rango r de la matriz \mathbf{C} se separan las matrices convenientemente de modo que se obtiene:

$$\mathbf{C}_r = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^H \quad (2.37)$$

donde \mathbf{U}_r contiene las primeras columnas de \mathbf{U} , al igual que \mathbf{V}_r con relación a \mathbf{V} y $\mathbf{\Sigma}_r$ contiene r valores singulares más grandes. Entonces el planteamiento original de la restricción de puntos múltiples cambia a:

$$\mathbf{C}_r^H \mathbf{w} = \mathbf{V}_r \mathbf{\Sigma}_r \mathbf{U}_r^H \mathbf{w} = \mathbf{f} \quad (2.38)$$

que se simplifica a:

$$\mathbf{U}_r^H \mathbf{w} = \mathbf{f}_r \quad (2.39)$$

con $\mathbf{f}_r = \mathbf{\Sigma}_r^{-1} \mathbf{V}_r^H \mathbf{f}$

Ahora se puede usar \mathbf{U}_r como la matriz de restricción final y \mathbf{f}_r como vector respuesta en el LCMV. Las ecuaciones de arriba pueden ser ampliadas con facilidad a múltiples direcciones incluyendo los vectores $\mathbf{d}(\theta, \omega)$ correspondientes en la matriz \mathbf{C} . Para la selección de r dado el ancho de banda B_θ y el ángulo DOA θ de la señal se emplea como guía una función del producto ‘tiempo-ancho de banda’ $B_\theta T_\theta$ de la señal. En [11] se demuestra que una fuente de banda ancha puede ser representada de forma exacta por $D_\theta = \lceil B_\theta T_\theta / \pi + 1 \rceil$ bases ortogonales, donde $\lceil \cdot \rceil$ es la función que redondea al siguiente valor entero mayor. Entonces eligiendo $r \geq D_\theta$ como guía se expande efectivamente el espacio de restricciones. $T_\theta = \tau_{M-1}(\theta) + (J-1)Ts$ como en (2.8) y B_θ es la componente de frecuencia mayor para señales reales en banda base, $B_\theta = \omega_{max} - \omega_{min}$ para señales reales fuera de banda base y $B_\theta = (\omega_{max} - \omega_{min})/2$ para señales complejas [11].

En la figura 2.6 se muestran dos resultados de simulaciones realizadas para esta tesis, basadas en un arreglo lineal uniforme para $M = 8$ y señales de banda ancha con una frecuencia normalizada $\Omega \in [0.25\pi \ \pi]$. En las figuras 2.6a y 2.6b la longitud del filtro FIR por cada canal fue $J = 48$, la frecuencia de muestreo fue $f_s = 16 \text{ KHz}$ y la meta fue recibir una señal arribando desde un DOA $\theta = -40^\circ$. En las figuras 2.6c y 2.6d la longitud del filtro FIR por cada canal fue $J = 96$, la frecuencia de muestreo fue $f_s = 48 \text{ KHz}$ y la meta fue recibir una señal arribando desde un DOA $\theta = 10^\circ$ y anular dos señales interferentes arribando desde $\theta = -30^\circ$ y 40°

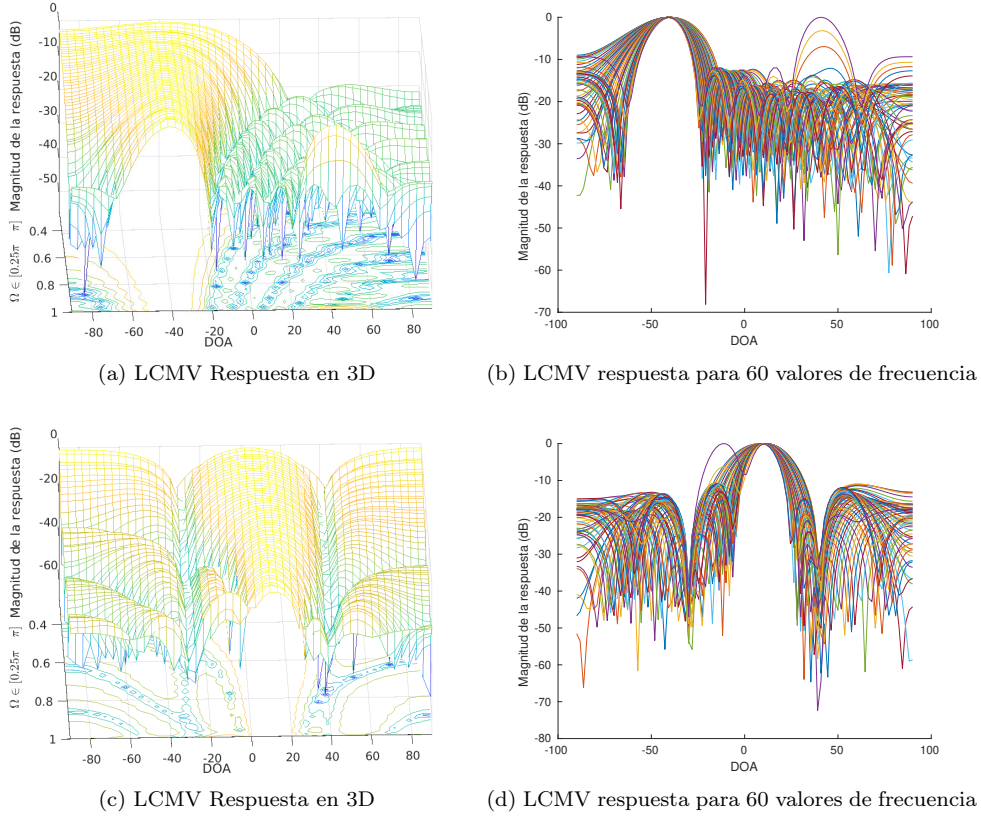


Figura 2.6: Respuesta o beampattern (BP) de dos LCMV Bf sobre la banda de frecuencia $[0.25\pi \ \pi]$ con respecto al ángulo DOA.

2.4.4. Cancelador Generalizado de Lóbulos Laterales

El cancelador generalizado de lóbulos laterales (Generalized Sidelobe Canceller (GSC)) [16] representa una formulación alternativa del problema del LMS. El GSC es un mecanismo para transformar el problema de minimización con restricción de (2.31) en uno sin restricción, de modo que en esta nueva estructura de Bf se puede aplicar algoritmos adaptivos bien conocidos como Least Mean Square (LMS), LMS normalizado (Normalized Least Mean Square (NLMS)) y LMS recursivo (RLS) [4],[14] entre otros.

Descomponiendo el vector de pesos \mathbf{w} en dos componentes ortogonales \mathbf{w}_0 y $-\mathbf{v}$ ($\mathbf{w} = \mathbf{w}_0 - \mathbf{v}$) que yacen en el rango y el espacio nulo de \mathbf{C} respectivamente (el rango y el espacio nulo de una matriz expanden todo el espacio, por tanto esta descomposición representa cualquier \mathbf{w}), dado que $\mathbf{C}^H \mathbf{v} = 0$ si \mathbf{w} satisface las restricciones se tendrá que:

$$\mathbf{w}_0 = \mathbf{C}(\mathbf{C}^H \mathbf{C})^{-1} \mathbf{f} \quad (2.40)$$

Si \mathbf{C}_n forma una base del espacio nulo de \mathbf{C} entonces su dimensión es $(N = MJ) \times (N - r)$ y \mathbf{v} se puede expresar como una combinación lineal de sus columnas ($\mathbf{v} = \mathbf{C}_n \mathbf{w}_n$). \mathbf{C}_n se puede obtener a partir de \mathbf{C} mediante cualquier procedimiento de ortogonalización como Gram-Schmidt, SVD u otros. En la figura 2.7 se muestran las descomposiciones del vector de pesos $\mathbf{w} = \mathbf{w}_0 - \mathbf{C}_n \mathbf{w}_n$. La existencia y elección de \mathbf{w}_0 y \mathbf{C}_n implican que \mathbf{w} satisface las restricciones independientemente de \mathbf{w}_n y se reduce el problema del LCMV a un problema sin restricción:

$$\min_{\mathbf{w}} = [\mathbf{w}_0 - \mathbf{C}_n \mathbf{w}_n]^H \mathbf{R}_x [\mathbf{w}_0 - \mathbf{C}_n \mathbf{w}_n] \quad (2.41)$$

La solución es:

$$\mathbf{w}_n = (\mathbf{C}_n^H \mathbf{R}_x \mathbf{C}_n)^{-1} \mathbf{C}_n^H \mathbf{R}_x \mathbf{w}_0 \quad (2.42)$$

La transformación del beamformer LCMV restringido en un beamformer con estructura GSC no restringido [9] se observa en la figura 2.7, donde 2.7a muestra el LCMV con restricción, 2.7b muestra la descomposición del vector de pesos en dos componentes ortogonales, una en el rango de \mathbf{C} y otra en su espacio nulo y en 2.7c se muestra la descomposición adicional de la componente del espacio nulo en una parte fija y una adaptativa, que resulta finalmente en el problema no restringido.

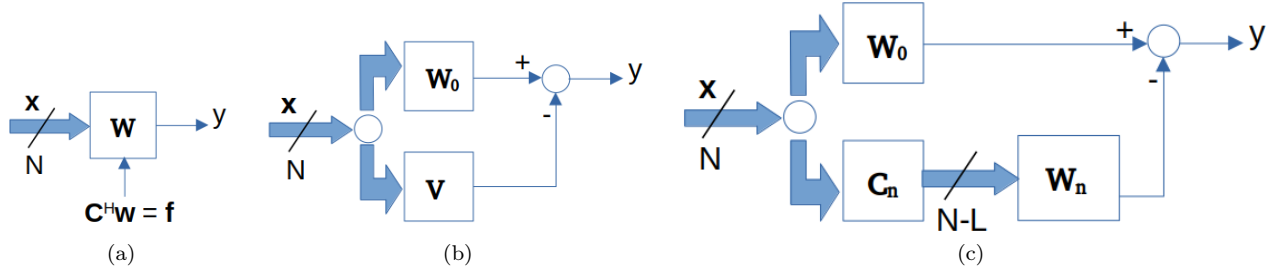


Figura 2.7: La transformación del beamformer LCMV restringido en un beamformer con estructura GSC no restringido.

Podemos considerar al vector \mathbf{w}_0 y las columnas de \mathbf{C}_n como vectores de pesos o Bf fijos aplicados al arreglo de datos, de modo que la rama alta del GSC en la figura 2.7c permite el paso de las señales deseadas exactamente con las características que se requiere. Se puede tomar como ejemplos de respuestas de \mathbf{w}_0 las simulaciones de la figura 2.6b y 2.6d. Por otro lado se tiene $\mathbf{C}^H \mathbf{C}_n = \mathbf{0}$, entonces la rama baja del GSC tendrá respuesta cero estas señales deseadas y entonces se puede suponer que las columnas de \mathbf{C}_n bloquearon o cancelaron esas direcciones y frecuencias. Por esta razón \mathbf{C}_n es referida como matriz de bloqueo.

La figura 2.8 muestra las respuestas de una columna de las matrices de bloqueo obtenidas de simulaciones realizada en este trabajo. Estas matrices componen los GSC resultantes de transformar los LCMV de las simulaciones mostradas en la figura 2.6. Se observa que la respuesta es cero en $\theta = -40^\circ$ en el caso de figura 2.8a y en $\theta = 10^\circ, -30^\circ$ y 40° en la figura 2.8b, tal como se pretendía con estas simulaciones y en correspondencia con las figuras 2.6b y 2.6d respectivamente. Así el GSC estima por su rama baja el ruido y las señales interferentes que pasan por la rama alta para luego sustraerlas.

2.4.5. Algoritmos Adaptables para Beamforming

El cálculo de los vectores óptimos en las secciones anteriores requiere el conocimiento de la estadística de segundo orden. Esta estadística es usualmente desconocida pero suponiendo ergodicidad se pueden estimar y por tanto estimar los vectores óptimos a partir de los datos disponibles. Además la estadística puede variar en el tiempo, por ejemplo, debido a fuentes móviles. Para resolver estos problemas los pesos son típicamente estimados mediante algoritmos adaptativos.

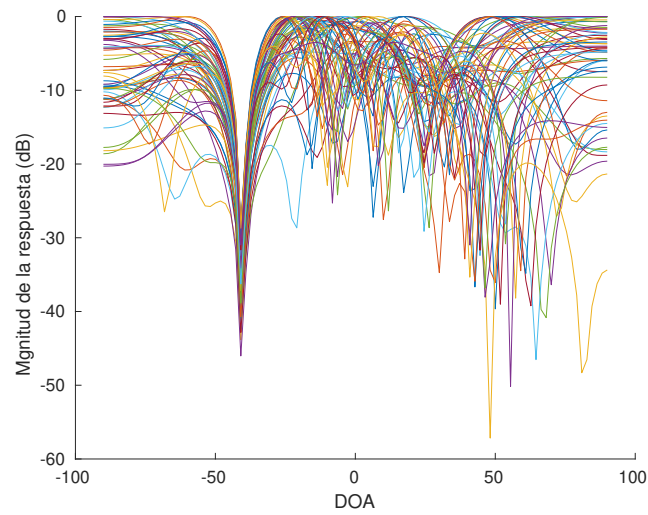
Existen dos procedimientos adaptativos básicos:

1. Adaptación en bloque, donde la estadística es estimada a partir de un bloque temporal de arreglos de datos y luego usado en las ecuaciones para los pesos óptimos.
2. Adaptación continua, en la que los pesos son ajustados al mismo tiempo que se adquieren las muestras de señales; de manera que la secuencia de vectores de pesos converge a la solución óptima.

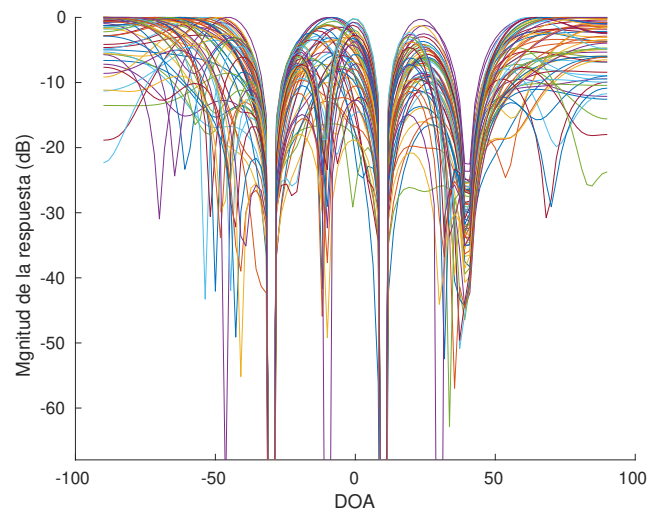
Si se espera un escenario no estacionario entonces la adaptación en bloque puede emplearse proveyendo que se recalculen los pesos periódicamente. La adaptación continua se emplea preferiblemente cuando la estadística cambia en el tiempo o cuando el número de pesos es grande.

Relación entre GSC y filtro adaptivo estandar

El problema estándar de filtrado adaptivo se muestra en la figura 2.9



(a)



(b)

Figura 2.8: Patrón de haces de una columna de \mathbf{C}_n . (a) Muestra una respuesta cero en $\theta = -40^\circ$. El bloqueo de las direcciones $\theta = 10^\circ$, -30° y 40° se muestra en (b)

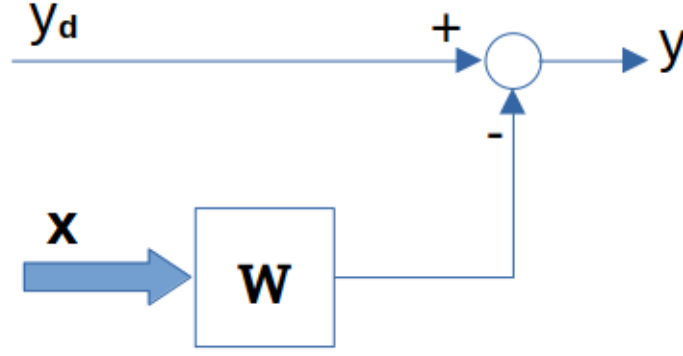


Figura 2.9: Estructura del problema estándar de filtrado adaptivo

En la figura 2.9 los pesos son elegidos para estimar la señal deseada y_d como una combinación lineal de los elementos del vector de datos \mathbf{x} . \mathbf{w} se selecciona para minimizar el error cuadrático medio (Mean Square Error (MSE)):

$$\begin{aligned}\xi(\mathbf{w}) &= E \left\{ |y_d - \mathbf{w}^H \mathbf{x}|^2 \right\} \\ &= \sigma_d^2 - \mathbf{w}^H \mathbf{r}_{xd} - \mathbf{r}_{xd}^H \mathbf{w} + \mathbf{w}^H \mathbf{R}_x \mathbf{w}\end{aligned}\quad (2.43)$$

donde $\sigma_d^2 = E \left\{ |y_d|^2 \right\}$, $\mathbf{r}_{xd} = E \left\{ |\mathbf{x} y_d^*|^2 \right\}$ y $\mathbf{R}_x = E \left\{ |\mathbf{x} \mathbf{x}^H|^2 \right\}$. (2.43) es minimizado por:

$$\mathbf{w} = \mathbf{R}_x^{-1} \mathbf{r}_{xd} \quad (2.44)$$

Comparando este problema con el problema del MSC se desprende que son equivalentes. El problema del LCMV puede parecer diferente, sin embargo, examinando las figuras 2.7 y 2.9 y las ecuaciones (2.42) y (2.44) se concluye que el problema estándar de filtrado adaptivo es equivalente al LCMV implementado con la estructura GSC.

La adaptación continua es aplicada en términos de la figura 2.9 y la ecuación (2.44) notando que el MSE $\xi(\mathbf{w})$ es una función de los pesos y tiene forma de una hiperparábola con un mínimo único para una matriz de covarianza con rango completo. El punto mínimo puede ser encontrado mediante una regla de actualización que involucra correcciones sucesivas del vector de pesos $\mathbf{w}[k]$ desde un punto de inicio en la dirección del gradiente negativo del MSE el cual puede ser expresado como [4],[14]:

$$\mathbf{w}[k+1] = \mathbf{w}_0 - \mu \nabla \xi[k] \quad (2.45)$$

despejando la ecuación anterior y haciendo una estimación instantánea de la matriz de covarianza y el vector de correlación cruzada a partir de una sola realización del arreglo de datos como $\mathbf{R}_x = \mathbf{x} \mathbf{x}^H$ y $\mathbf{r}_{xd} = \mathbf{x} y_d^*$ se obtiene la ecuación de conocido algoritmo LMS [14]:

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \mu \mathbf{x}[k] y^*[k] \quad (2.46)$$

2.5. Aplicaciones a la Estimación de DOA

La Estimación DOA es una de las mayores áreas de aplicación en el procesamiento de arreglo de señales. Para señales de banda angosta, dos algoritmos bien conocidos son Multiple Signal Classification (MUSIC) y Estimation of Signal Parameters via Rotational Invariant Techniques (ESPIRIT), para señales de banda ancha el procedimiento más usual es descomponerlas en bandas estrechas de frecuencias y aplicar los métodos DOA antes mencionados a cada banda de frecuencia. Sin embargo, el BF también brinda una alternativa viable al problema de la estimación de DOA, siendo lo más simple emplear una serie de Bf fijos apuntando a diferentes direcciones, para luego estimar sus potencias de salida y compararlas para decidir la dirección correcta de las fuentes de señales.

2.5.1. Métodos convencionales de estimación DOA

Método basado en Beamforming

Los métodos de BF obtienen un espectro de potencia espacial mediante la formación de haces en múltiples direcciones y la localización de una fuente en función de la dirección que tiene la potencia máxima. Para estimar el DOA para múltiples fuentes, se encuentran los picos más grandes en el espectro de potencia espacial, que corresponden al número de fuentes. Cualquier técnica de formación de haz podría usarse para la estimación de DOA, siendo una popular el formador de haz MVDR. Los lóbulos laterales del patrón de haz limitan la resolución y la precisión de la estimación que se puede lograr a partir de una técnica específica de BF [31].

Método basado en subespacios

Los métodos subespaciales descomponen la matriz de covarianza de las señales en sus eigenvectores y eigenvalores. Los N eigenvalores más grandes, donde N es el número de fuentes, corresponden a los eigenvectores de la señal que forman el subespacio de señal ortogonal al resto de eigenvectores que corresponden al subespacio de ruido. Las técnicas populares incluyen la clasificación de señales múltiples (MUSIC) y la estimación de parámetros de señales a través de técnicas de invariancia rotacional (ESPIRIT). Sus limitaciones para el escenario de fuentes múltiples es que requieren que el número de fuentes se conozca de antemano para que el número correcto de eigenvectores de la señal se elija y que funcionan solo para señales no correlacionadas e incoherentes [31].

Método basado en TDOA

Los métodos basados en la diferencia del tiempo de arribo (Time Delay of Arrival (TDOA)) se basan en encontrar el TDOA de la señal de origen para los diferentes pares de micrófonos de la matriz, ya que una señal de sonido incidente llegará a cada micrófono en un momento ligeramente diferente según el ángulo de llegada. Mediante el uso de técnicas como la correlación cruzada generalizada (Generalized Cross-Correlation (GCC)) o Generalised Cross-Correlation with Phase Transform (GCC-PHAT), que emplea la transformada de fase como función de peso para mejorar la SNR; es posible encontrar el TDOA para cada par de micrófonos y luego combinar esta información con la posición de los micrófonos para ubicar el DOA [31].

2.5.2. LCMV para estimación DOA

Una mejora adicional a la idea de emplear Bf fijos es emplear el algoritmo LCMV. Para señales de banda ancha, por cada dirección de arribo supuesta se diseña un conjunto de restricciones lineales sobre la banda de frecuencia de interés usando el método de los eigenvectores y se aplica el LCMV resultante. La potencia de salida del Bf con respecto al ángulo DOA se obtiene con su pico correspondiendo al ángulo DOA real [9],[10].

Para estimar la dirección de arribo de una señal incidente se muestrea uniformemente el intervalo de direcciones desde $\theta = -90^\circ$ a 90° con 181 puntos θ_i , $i = 0, 1, \dots, 180$. Para cada θ_i se formula \mathbf{C}_r , eligiendo r de acuerdo a la subsección 2.4.3 y calculando el vector óptimo \mathbf{w}_{opt} de acuerdo con (2.32). Por último se calcula la potencia asociada al DOA θ_i , la cual se puede simplificar como:

$$\begin{aligned}
 p_\theta &= \mathbf{w}_{opt}^H \mathbf{R}_x \mathbf{w}_{opt} \\
 &= \mathbf{f}^H \left[\mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \right]^{-1} \mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{R}_x \mathbf{R}_x^{-1} \mathbf{C} \left[\mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \right]^{-1} \mathbf{f} \\
 &= \mathbf{f}^H \left[\mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \right]^{-1} \mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \left[\mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \right]^{-1} \mathbf{f} \\
 &= \mathbf{f}^H \left[\mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \right]^{-1} \mathbf{f}
 \end{aligned} \tag{2.47}$$

Sustituyendo por sus correspondientes en el método de eigenvector se obtiene:

$$p_\theta = \mathbf{f}_r^H \left[\mathbf{U}_r^H \mathbf{R}_x \mathbf{U} \right] \mathbf{f}_r \tag{2.48}$$

En la figura 2.10 se muestra la simulación realizada basada en un arreglo lineal uniformemente espaciado con $M = 8$, $J = 96$ para cada canal, al cual incide una señal real con un rango de frecuencia

normalizada $\Omega \in [0.008\pi \quad 0.17\pi]$ desde una dirección $\theta = 22^\circ$, un SNR de 5db y la frecuencia de muestreo fue de $f_s = 48\text{ KHz}$.

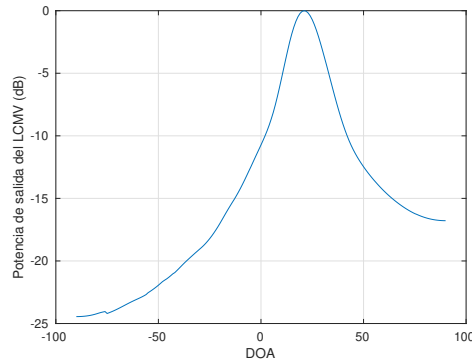


Figura 2.10: La curva muestra la potencia de salida en función del DOA.

Resumen

En este capítulo se abordaron aspectos generales de BF que sientan las bases de su uso en este trabajo. Se analizaron los tipos de beamformer que se consideraron para la implementación del Neuro-Detector, realizándose simulaciones con fines demostrativos para esta tesis, específicamente se comprobó la eficacia del LCMV para filtrar espacialmente a partir de la selección de la matriz de restricciones mediante el método de eigenvectores. Las simulaciones hechas en el caso del GSC demuestran la acción de la matriz de bloqueo y centran el análisis en la funcionalidad de aislar o separar las señales de interés del ruidos y señales interferentes. Además, se hizo notar como esta idea de separar subespacios de interés y ruido mediante descomposición ortogonal del vector de pesos y de la matriz de restricciones convierte al GSC en una alternativa de implementación del LCMV que permite incorporar a su funcionamiento técnicas adaptables bien establecidas. Se mostraron de forma concisa los principales métodos para la estimación DOA y se resaltó, a través de los resultados de la simulación realizada, el uso del método basado en BF como una alternativa viable y efectiva.

Capítulo 3

Principios de Redes Neuronales.

La inteligencia artificial (Artificial Intelligence (AI)) [17] es una de las áreas de investigación más exitosas del siglo 21. Los nuevos métodos de acercamiento a la AI llamados aprendizaje profundo [18] usan modelos compuestos por muchas capas que pueden extraer información sobre características de datos sin procesar. Estas capas están formadas por nodos inspirados por las neuronas en el cerebro humano, lo cual es una de las razones por las que se les llaman redes neuronales artificiales (NN). En los últimos años, los trabajos y resultados basados en aprendizaje profundo son cuantiosos [26]. Arquitecturas de aprendizaje profundo como redes neuronales profundas, aprendizaje profundo reforzado, redes neuronales recurrentes y convolucionales han sido aplicadas en diversos campos, y específicamente para estimación de DOA han sido utilizadas con muchos ejemplos exitosos [19]-[25].

En este capítulo se presenta una breve introducción a las redes neuronales, se describen conceptos fundamentales de la arquitectura de las redes neuronales y los paradigmas de aprendizaje. Se abordan algunas de las ventajas de incluir Neural Networks (NNs) en esta tesis y se debaten temas relacionados a las NNs que fueron analizados y tomados en cuenta para este trabajo.

3.1. Aspectos Generales de NN

En su forma más general una red neuronal está diseñada para modelar la manera en la que el cerebro desempeña una tarea particular o función de interés. La red suele ser implementada mediante componentes electrónicos o simulada por software en una computadora digital. Para este trabajo, el interés estuvo en la clase de NNs que ejecuta tareas o cálculos mediante un proceso de aprendizaje. Para lograr un buen desempeño las NNs emplean una interconexión masiva de las unidades de procesamiento o cómputo llamadas neuronas. Desde el punto de vista adaptivo una NN es un procesador masivamente distribuido en paralelo, compuesto de unidades simples de procesamiento que tienen una tendencia intrínseca a guardar conocimiento y hacerlo disponible para su uso. Tiene dos aspectos parecidos al cerebro humano: que el conocimiento es adquirido por la NN desde su medio ambiente a través de un proceso de aprendizaje y que la fuerza o intensidad de las conexiones interneuronales, conocida como peso sináptico, es usada para guardar los conocimientos adquiridos [17].

El aprendizaje de máquina y en consecuencia las NN emplean modelos compuestos por varias capas para extraer información de las características de datos sin procesar. Estas capas están compuestas de neuronas también llamadas nodos. Cada uno de esos nodos opera obteniendo múltiples señales de entrada, que son multiplicadas por pesos y sumadas a un bias o sesgo para luego aplicar al resultado una transformación mediante una función de activación no lineal, lo que le permite mapear salidas no lineales [18], esto se aprecia en la figura 3.1b. Conectando los nodos entre capas y apilando muchas de estas capas (de ahí el nombre Aprendizaje Profundo), se pueden crear funciones muy complejas. Las capas que se encuentran entre las capas de entrada y de salida se les conocen como capas ocultas (Hidden Layer (HL)) y son responsables de extraer el mayor volumen de características abstractas de los datos de entrada [18]. Usando una técnica llamada retropropagación la NN puede encontrar el error entre los parámetros del modelo y los valores de salida y posteriormente modificar los pesos de cada nodo para minimizar este error y aprender [17], parte de las ecuaciones de esta técnica se muestran en la figura 3.1c. NNs profundas compuestas por nodos completamente conectados entre ellos se llaman **Multilayer Perceptron (MLP)**, un ejemplo se muestra en la figura 3.1 y son las redes profundas feedforward (subsección: 3.2.1)

más básicas que pueden aprender a mapear valores de salida a partir de los datos de entrada [18]. Otros tipos de NNs profundas incluyen las **Convolutional Neural Network (CNN)**, las que aplican pesos llamados filtros con una operación discreta de convolución a diferentes conjuntos de valores de entrada para extraer sus características [28],[32]. Las **Recurrent Neural Network (RNN)**(subsección 3.2.2) son muy usadas para resolver tareas secuenciales ya que las salidas de los modelos anteriores son almacenadas en las HL para ser tomadas en cuenta en la siguiente iteración [32].

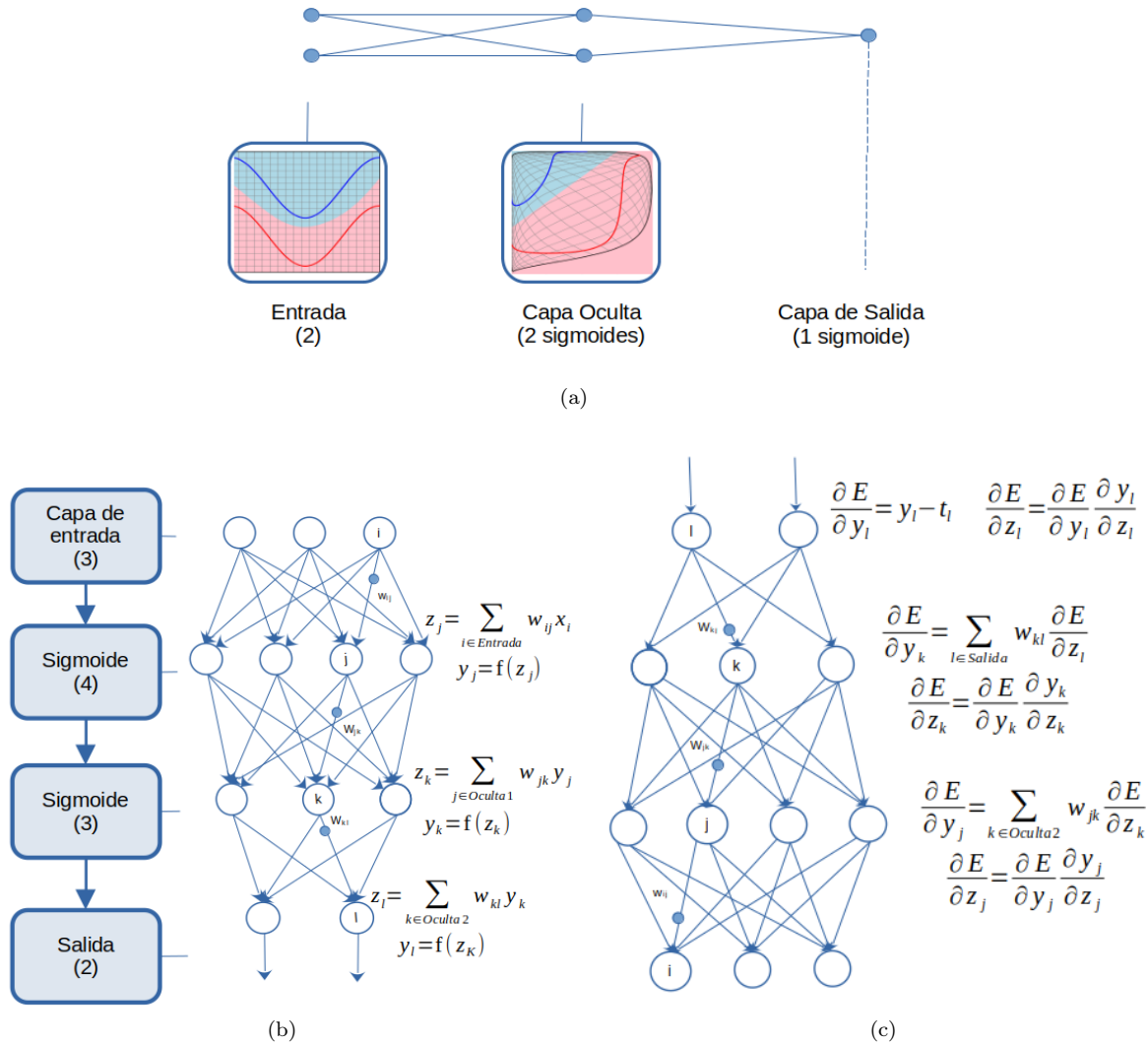


Figura 3.1: NN multicapas y la retropropagación

En la figura 3.1a se muestra una NN multicapa y se resalta que la integración de varias capas dota a la NN con la capacidad de distorsionar el espacio de entrada para lograr hacer las clases de datos, mostradas en líneas azul y roja, linealmente separables. Se muestra también cómo las cuadrillas dispuestas uniformemente en el espacio de entrada se transforman por los nodos ocultos. Este constituye un ejemplo ilustrativo con solo dos nodos de entrada, una sola capa oculta compuesta de dos nodos y un solo nodo en la capa de salida, sin embargo, las redes empleadas para tareas complejas como reconocimiento de objetos y otras están compuestas de decenas o cientos de miles de nodos [32]. En la figura 3.1b se muestran las ecuaciones usadas en el cálculo de las señales hasta obtener la respuestas de una NN con dos capas ocultas y una capa de salida; cada ecuación constituye un módulo a través del cual se retropropaga el gradiente. En cada capa se calcula primero la entrada total z de cada nodo, la cual es una suma ponderada de las salidas de los nodos de la capa anterior. Entonces una función no lineal $f(\cdot)$ es aplicada a z para obtener la salida del nodo en cuestión. Las funciones no lineales usadas en una NN incluyen la unidad lineal rectificadora (ReLU) $f(z) = \max(0, z)$, comúnmente usada en años recientes, y las más convencionales funciones sigmoides, tales como la tangente hiperbólica $f(z) = (\exp(z) - \exp(-z))/(\exp(z) + \exp(-z))$

y la función logística $f(z) = 1/(1 + \exp(-z))$ [32]. En la figura 3.1c se muestran las ecuaciones usadas para la retropropagación del error. En cada capa oculta se calcula la derivada del error con respecto a la salida de cada nodo, la cual es la suma ponderada de las derivadas del error con respecto a las entradas totales de los nodos de la capa posterior. Luego se convierte la derivada del error con respecto a la salida en la derivada del error con respecto a la entrada multiplicando la primera por el gradiente de $f(z)$. Una vez calculada $\partial E/\partial z_k$, la derivada del error para el peso w_{jk} en la conexión desde el nodo j se calcula como $y_j \partial E/\partial z_k$ [32].

El procedimiento para realizar el proceso de aprendizaje se denomina algoritmo de aprendizaje, y tiene como función modificar los pesos sinápticos en la red de manera organizada para lograr cierto objetivo. La modificación de los pesos sinápticos proporciona el método tradicional para el diseño de la NN. Este acercamiento es el más cercano a la teoría de filtrado lineal adaptivo, la cual está bien establecida y aplicada exitosamente en diversos campos [4],[14]. Sin embargo, es posible para una NN modificar su propia topología, lo cual está motivado por el hecho de que las neuronas en el cerebro humano pueden morir y nuevas conexiones sinápticas pueden crecer.

3.1.1. Beneficios de las NNs

Es evidente que las NNs derivan sus fortalezas de cálculo a través de su estructura de distribución masiva en paralelo en primer lugar, y en segundo en su habilidad para aprender y por tanto generalizar. La generalización se refiere a que la NN produce salidas razonables ante entradas que nunca estuvieron presente durante el entrenamiento o aprendizaje. Para este trabajo se pueden identificar las siguientes ventajas de emplear una NN:

- No linealidad: la NN hecha de interconexiones de neuronas no lineales es no lineal. Esta no linealidad es en cierto modo especial pues está distribuida a lo largo de la NN. La no linealidad es una propiedad muy importante en este caso ya que el mecanismo responsable de generar las señales de entrada (señales de voz) de la NN es inherentemente no lineal.
- Mapeo de entrada-salida: en este trabajo para la NN se empleó un paradigma de aprendizaje llamado aprendizaje supervisado o con maestro, el cual involucra la modificación de los pesos sinápticos mediante la aplicación de un conjunto de ejemplos de entrenamiento etiquetados. La importancia de este mapa de entradas-salidas, que crea la NN a partir del entrenamiento, es que es un mecanismo no paramétrico, es decir, que no hay ninguna suposición previa del modelo estadístico de las señales de voz de entrada.
- Adaptabilidad: la NN tiene la capacidad intrínseca de adaptar los pesos sinápticos a los cambios del medio circundante. En particular los pesos pueden ser actualizados en tiempo real cuando opera en un entorno no estacionario, es decir, que la estadística cambia con el tiempo; lo cual es el caso de una habitación con hablantes o interferencias en movimiento.
- Implementación VLSI: la naturaleza en paralelo de la NN la hace muy rápida en el cálculo de ciertas tareas. Esta misma característica la hace muy adecuada para su implementación en tecnologías de escala muy grande de integración (Very Large-Scale Integration (VLSI)) como los Digital Signal Processor (DSP), arquitecturas Advanced RISC Machine (ARM) como las Raspberry Pi, entre otros.

Las redes neuronales y el aprendizaje profundo en general, muestran marcadas ventajas adaptativas y computacionales como son su estructura distribuida masivamente en paralelo y además su habilidad para aprender y generalizar. Este paralelismo hace a las redes neuronales potencialmente rápidas para el cálculo de ciertas tareas y muy adecuadas para su implementación usando tecnología de muy alta escala de integración (VLSI), así como procesamiento en paralelo. A su vez esta virtud de VLSI le provee de formas para capturar comportamientos complejos de una forma altamente jerárquica.

3.2. Arquitecturas de NN

La manera en que las neuronas están estructuradas en una NN está íntimamente relacionado al algoritmo de aprendizaje usado para entrenar la red. En esta sección se abordan las arquitecturas o estructuras más generales de NN. En general se pueden identificar dos clases diferentes fundamentales [17]:

3.2.1. NNs Multi-Capas Feedforward

La forma más sencilla de una NN, es una capa de nodos de entradas que proyectan directamente sobre la capa de neuronas de salida. Cuando la interconexión de capas es estrictamente en un sentido, hacia adelante, la NN es del tipo feedforward. en la figura 3.2a se muestra este caso para cuatro nodos en ambas capas. Este tipo de NN es llamada de una-capas, donde este término solo hace referencia a la capa de salida de neuronas. No se toma en cuenta la primera capa de nodos de entradas pues en esta no se realizan cálculos.

La clase de NNs feedforward multi-capas se distingue por tener una o más de una capa oculta de neuronas. El término “oculta” se refiere a que estas capas no son vistas directamente desde la entrada ni la salida de la NN. Añadiendo una o más capas ocultas la NN es capaz de extraer más características y estadísticas de orden superior de la información de entrada.

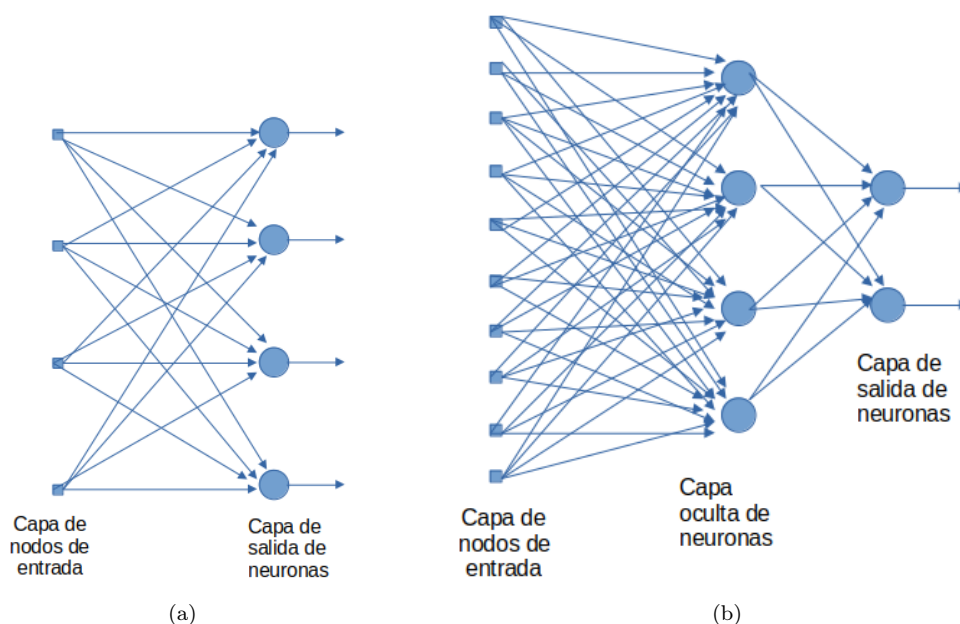


Figura 3.2: a) NN feedforward con una capa de neuronas. b) NN feedforward completamente conectada con una capa oculta y una capa de salida

La NN de la figura 3.2b se dice que es completamente conectada en el sentido que cada nodo en cada capa esta conectado con todos los nodos de la siguiente capa hacia adelante. Si alguna interconexión entre capas adyacentes no está presente se dice que la NN es parcialmente conectada.

3.2.2. NN Recurrentes

Una NN recurrente se distingue de una feedforward por la presencia de al menos un lazo de retroalimentación. Por ejemplo una red recurrente puede estar compuesta por una sola capa de neuronas, con cada neurona retroalimentando su salida a la entrada del resto de las neuronas, como se muestra en la figura 3.3. En la estructura que se muestra en esta figura no hay lazos propios de retroalimentación; retroalimentación propia se refiere a alimentar la salida de la neurona a su propia entrada. La NN recurrente de la figura 3.3 tampoco tiene capas ocultas.

En la figura 3.4 se muestra otro ejemplo de NN recurrente pero en este caso si existe una capa oculta y el lazo de retroalimentación tiene contribuciones de la capa de salida así como de la capa oculta.

La presencia de retroalimentación tiene un marcado impacto en la capacidad de la red para aprender y también en su desempeño. Además como se muestra, involucra ramas que incluyen elementos de retardo de una unidad de tiempo, denotados por z^{-1} .

La cuestión de cómo es representado el conocimiento en una NN está directamente relacionado a su arquitectura. Sin embargo no existe una teoría bien desarrollada para la selección y optimización de la arquitectura de una NN requerida para interactuar en un entorno de interés, o para evaluar la forma en

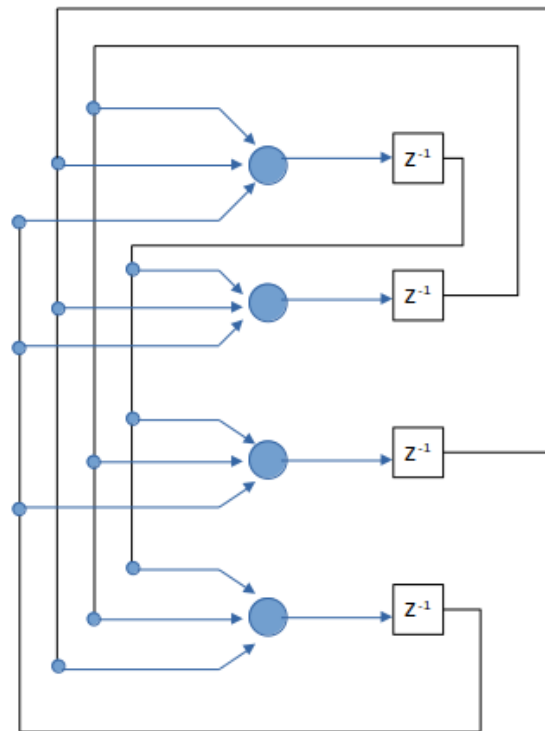


Figura 3.3: NN recurrente sin retroalimentación propia ni capas ocultas

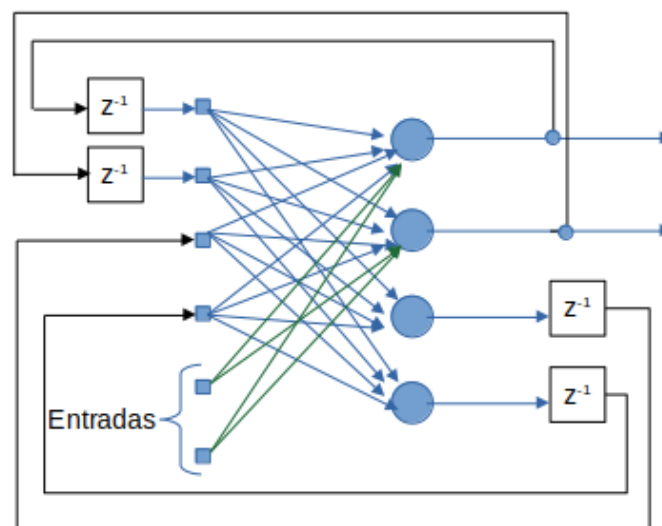


Figura 3.4: NN recurrente con una capa oculta

que los cambios en la arquitectura de la red afectan la representación del conocimiento dentro de esta [17]. Es por medio de estudios experimentales exhaustivos que se llegan a conclusiones satisfactorias para casos de interés particulares.

3.3. Proceso de Aprendizaje

En un sentido amplio se identifican dos categorías en el proceso de aprendizaje de una NN las cuales son: aprendizaje supervisado o con maestro y aprendizaje sin maestro. De igual forma el aprendizaje sin maestro puede ser categorizado en dos: aprendizaje no supervisado y aprendizaje reforzado [17]. Estas formas de realizar el entrenamiento o aprendizaje de la NN son similares a los del ser humano.

3.3.1. Aprendizaje Supervisado

La figura 3.5 muestra un diagrama de bloques que ilustra esta forma de aprendizaje. En forma conceptual se puede pensar que el maestro conoce el entorno, es decir, el mapeo entrada-salida del entorno, sin embargo, es desconocido para la NN. Durante el entrenamiento el maestro y la NN son expuestos a un vector de entrenamiento y el maestro provee a la NN con la respuesta deseada. La respuesta deseada es la salida óptima que debería dar la NN. Entonces los parámetros de la red son ajustados bajo la combinación de la influencia del vector de entrenamiento y la señal de error. La señal de error es definida como la diferencia entre la respuesta deseada y la respuesta dada por la red. El ajuste es realizado iterativamente con el fin de emular al maestro, la emulación se supone que sea óptima en algún sentido estadístico. De esta forma el conocimiento del entorno es pasado a la NN mediante el entrenamiento y almacenado en los pesos sinápticos. Cuando se alcanza este estado se puede prescindir del maestro y permitir a la NN tratar con el entorno por sí sola.

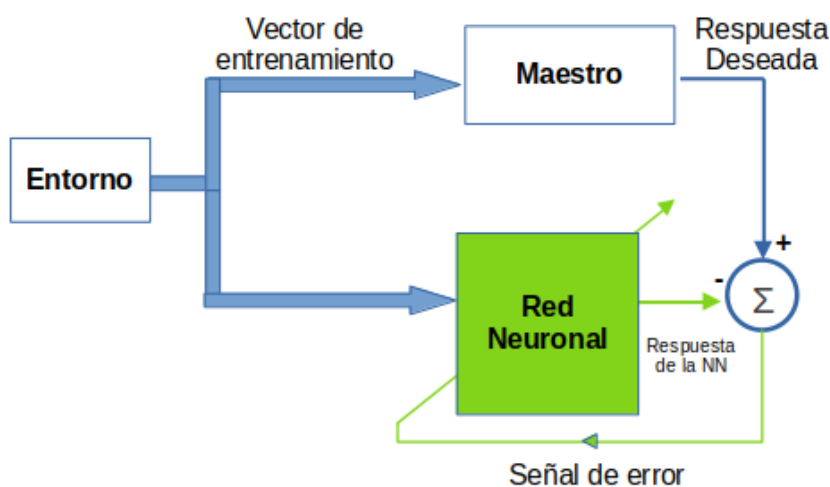


Figura 3.5: Diagrama en bloques del aprendizaje con maestro. Las partes de verde en la figura constituyen el lazo de retroalimentación.

En la figura 3.5 se observa que el proceso de aprendizaje supervisado es un lazo cerrado de retroalimentación donde no se incluye al entorno. La medida del desempeño de la NN bien puede ser en términos del error cuadrático medio o de la suma de errores cuadráticos sobre los ejemplos de entrenamiento, definiendo una función de los pesos de la red. Esta función constituye una superficie de desempeño de error multidimensional donde las coordenadas son los pesos sinápticos. Con cada vector de entrenamiento y cada ajuste resultante se obtiene un punto de la superficie. Para que la NN optimice su desempeño y aprenda del maestro, este punto de operación debe ir moviéndose sucesivamente hacia abajo en la dirección del menor valor de la superficie de error, ya sea un mínimo local o global. El sistema debe ser capaz de lograr esto a partir del conocimiento de la dirección del gradiente de la superficie de error, de modo que el punto de operación se mueva en el sentido contrario del gradiente o en sentido del descenso más inclinado.

3.3.2. Aprendizaje no Supervisado

A diferencia del aprendizaje con maestro, en este caso no existe una supervisión del aprendizaje o entrenamiento, lo cual es equivalente a decir que no existen ejemplos etiquetados para entrenar la NN. Bajo este paradigma se identifican dos categorías:

Aprendizaje Reforzado

El aprendizaje por refuerzo consiste en colocar un modelo llamado agente en un entorno, observar el estado del entorno y realizar una acción. Esta acción produce un cambio en el estado del entorno el cual es recompensado positiva o negativamente dependiendo del resultado deseado y la NN aprende tratando de realizar acciones que maximicen las recompensas.

En el aprendizaje reforzado, el mapeo entrada-salida es aprendido a partir de la interacción continua con el entorno en aras de minimizar un índice escalar de rendimiento. La figura 3.6 muestra una forma de aprendizaje reforzado construida alrededor de un sistema “crítico” o agente que convierte una señal de refuerzo primaria, escalar, en una de mayor calidad llamada señal heurística de refuerzo, también escalar. El sistema observa secuencias temporales de estímulo provenientes del entorno, las cuales eventualmente resultan en la generación de la señal heurística de refuerzo creada por el crítico o agente, es decir, el sistema aprende bajo una señal de refuerzo retardada.

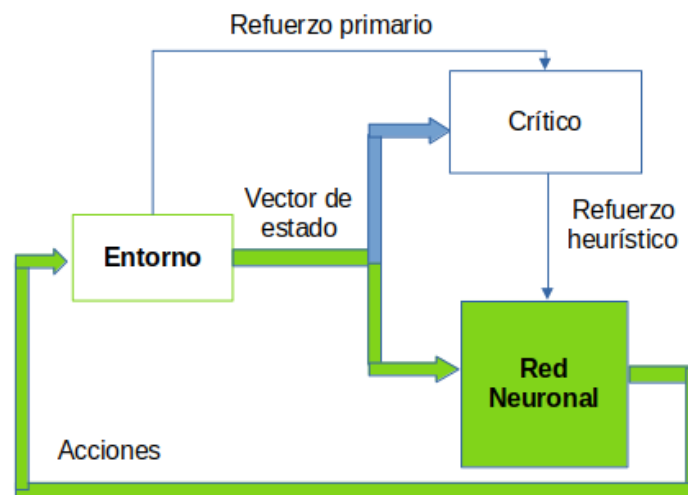


Figura 3.6: Diagrama de bloques del aprendizaje reforzado. El entorno forma parte del lazo de retroalimentación.

La meta del aprendizaje reforzado es minimizar una función de costos, definida como el valor esperado de la sumatoria de los costos de las acciones tomadas sobre una secuencia de estímulos del entorno. La función del sistema es encontrar las acciones que constituyan los mejores determinantes del comportamiento general del sistema y retroalimentarlas al entorno.

El mecanismo de refuerzo retardado es difícil debido a que el sistema debe ser capaz de otorgar costos que “aprueben” o den crédito y costos que “desaprueben” o penalicen las acciones individuales que se tomaron sobre las secuencias de estímulos para decidir la respuesta final; mientras que esta es evaluada por la señal de refuerzo. A pesar de estas dificultades el refuerzo retardado representa una opción atractiva en muchos casos ya que brinda las bases para que la NN interactúe por sí sola con el entorno basado en su experiencia durante la interacción.

Aprendizaje no supervisado

En el aprendizaje no supervisado, o autoorganizado, no existe un maestro ni agente externo para corregir o evaluar el proceso de aprendizaje, tal como se muestra en la figura 3.7. En su lugar se dispone de una medida de la calidad de representación que requiere la NN para aprender, esta medida es independiente de la tarea. Los parámetros de la NN son optimizados con respecto a esta medida. Para una medida determinada, una vez que la NN se ha sintonizado con las regularidades estadísticas de los

datos de entrada, entonces la red desarrolla habilidades para formar representaciones internas de las características de la entrada y para formar nuevas clases.

Por ejemplo para llevar a cabo aprendizaje no supervisado se puede utilizar una regla de competitividad, mediante la cual las neuronas de la última capa, encargadas de dar los resultados de la NN, compiten entre ellas por la “oportunidad” de responder a las características de los datos de entrada. De esta forma la NN opera bajo la estrategia de que “el ganador se lo lleva todo”, mediante la cual la neurona con el mayor resultado se enciende o se dispara mientras las demás permanecen apagadas.

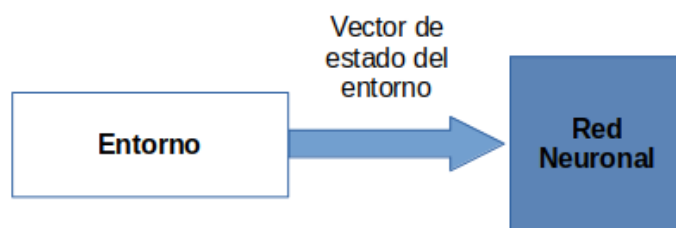


Figura 3.7: Diagrama de bloques del aprendizaje no supervisado

3.4. Beamforming como Tarea de Aprendizaje

En la sección anterior se analizaron los diferentes paradigmas de aprendizaje de una NN. La selección de uno de éstos en particular depende de la tarea que queremos que resuelva la red o de la aplicación dentro de la cual desempeña un rol. Por otra parte, en el capítulo 2, sección 2.4.4 se abordó el GSC como una alternativa de implementación del LCMV y se resalta que este permite la utilización de algoritmos adaptables bien establecidos como parte de su funcionamiento .

Para este trabajo, específicamente para decidir la funcionalidad y la integración de la NN en el detector DOA se tomó en cuenta las características y el análisis de una estructura para un Neuro-GSC brevemente descrita en [17] y que se muestra en la figura 3.8. Este sistema GSC se explica a través de sus componentes como sigue:

- Un arreglo de micrófonos para muestrear la señal de voz en puntos discretos del espacio.
- Un combinador lineal que actúa como filtro espacial, cuyos coeficientes serán calculados a partir de la descomposición ortogonal hecha como parte del GSC y usando la ecuación 2.40 . La salida del combinador lineal, denotada por $d(n)$, será la señal de referencia del Bf y de la NN.
- La matriz de bloqueo, \mathbf{C}_n , la cual impide el paso de la señal deseada y otras señales de la rama del combinador lineal, mientras que permite el paso de ruido e interferencias a través de los lóbulos laterales para así cancelarlas.
- Una NN cuyos parámetros serán ajustados en cada iteración para acomodar las variaciones estadísticas de las señales interferentes y el ruido, minimizando así la señal de error y por tanto cancelando las señales interferentes y el ruido de la rama principal del combinador lineal.

En este caso los pesos sinápticos de la NN son ajustados mediante un algoritmo de aprendizaje de corrección de error que opera sobre la señal de error $e(k)$, definida como la diferencia entre la salida del combinador lineal $d(k)$ y la salida actual $y(k)$ de la NN. Entonces el GSC y en concreto la NN operan bajo la supervisión del combinador lineal, tomando el rol de maestro. Tal como se mostró en la sección

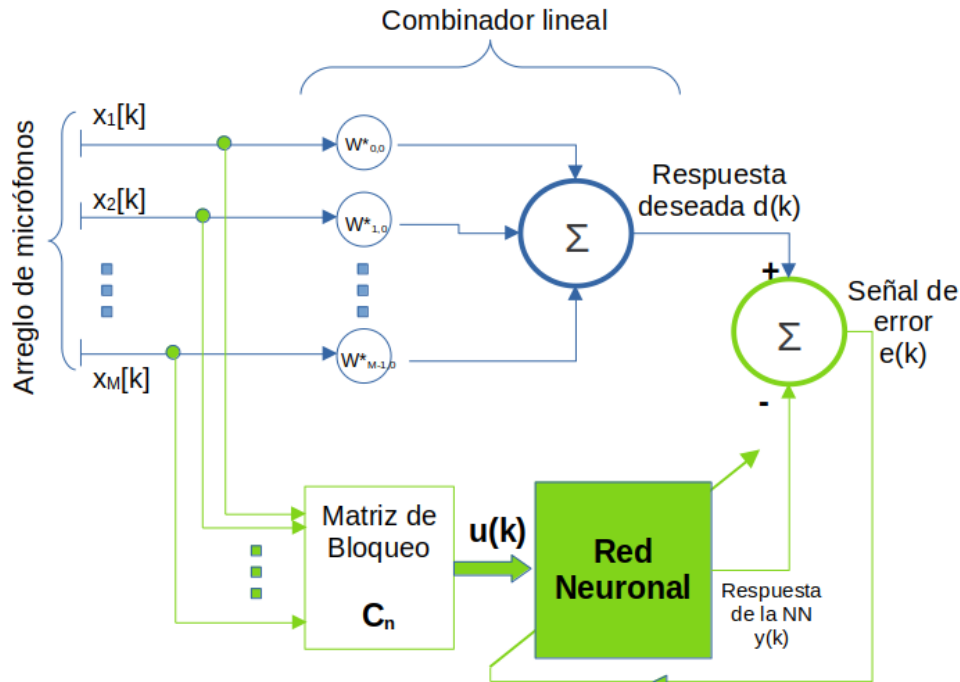


Figura 3.8: Diagrama de bloques del Neuro-GSC

anterior bajo el paradigma de aprendizaje supervisado el combinador lineal queda fuera del lazo de retroalimentación.

Como factor importante y de extremo interés para esta tesis, del análisis de esta estructura, se desprende que la funcionalidad de la red está en generalizar a partir de sus parámetros la mayor cantidad de variaciones estadísticas y las características de todas las señales interferentes o de ruido que inciden en el arreglo desde direcciones diferentes a la de interés y lograr resumir todo esto en un valor apropiado para ser utilizado ya sea para mejorar la relación señal-ruido a la salida del GSC o la separación de fuentes. Es decir, la NN es empleada como un agente de apoyo entre la entrada y la salida del sistema, capaz de interpretar las interferencias y el ruido y resumir una amplia variedad de características empleando aprendizaje supervisado.

3.5. Entrenamiento de NNs para Estimación DOA

Las NNs han sido empleadas en numerosas ocasiones para la estimación DOA y existen muchos ejemplos de los resultados logrados en este sentido. La mayoría de los investigadores emplean CNNs para extraer características espaciales a partir de GCC o de la transformada discreta de Fourier (Discrete Fourier Transform (DFT)) [19]-[22], mientras que algunos emplean MLPs [21],[33],[34]; logrando mediante ambos casos resultados similares.

En todos los casos el audio a la salida de los canales del arreglo de micrófonos debe ser procesado de forma que represente la información DOA antes de ser alimentado a la NN. El artículo [22] alimenta la DFT de cada canal a una CNN para extraer las características espaciales. Este método no entrega directamente el DOA, lo que es interesante para esta tesis, además, el hecho que deba enlazar las características de la DFT con la información espacial exige el uso de más datos de entrenamiento y de una NN con capacidades y complejidad mayores como es el caso de una CNN. Los artículos [33],[34] utilizan la técnica GCC-PHAT mencionada y referenciada en la subsección 2.5.1 para alimentar una MLP, en estos casos dicha técnica entrega picos que son el TDOA de la señal de entrada para diferentes combinaciones de micrófonos. A pesar que este no es el acercamiento que sigue esta tesis, sí es de interés hacer notar que en estos casos se le alimentó a la red información que corresponde más concretamente a información espacial y en concreto a tiempos de retardo lo que deriva directamente en DOA. El análisis más importante de estos últimos casos es, que el empleo de redes MLP es muy acertado y efectivo para

casos de estimación DOA siempre y cuando haya una correspondencia, en cierto grado explícita, entre los datos de entrada y el mapeo de salida de la NN.

En todos los artículos anteriores se menciona como principal limitante la dificultad para adquirir suficientes muestras de entrenamiento, especialmente en el caso de escenarios con múltiples fuentes sonoras, sobretodo a medida que estas aumentan en número. Basado en el análisis y la simulación de la subsección 2.5.2 considerando un arreglo lineal uniforme y suponiendo que queremos lograr una resolución de un grado, cada fuente de voz puede estar en cualquiera de 181 grados diferentes. La forma en que se pueden distribuir fuentes de voz en 181 grados, sin importar el orden, es una combinación de 181 objetos distintos tomados k a la vez, donde k es el número de fuentes en este caso. La ecuación para el número de combinaciones con respecto a k es:

$$\binom{181}{k} = \frac{181!}{k!(181-k)!} \quad (3.1)$$

Tomando en cuenta las simulaciones hechas y dado que el número de fuentes debe ser menor al número de micrófonos en el arreglo [31], el número de combinaciones requeridas para entrenar la NN para hasta 7 fuentes sería:

- $k_1 = 181$
- $k_2 = 16290$
- $k_3 = 971970$
- $k_4 = 43 \times 10^6$
- $k_5 = 1 \times 10^9$
- $k_6 = 44 \times 10^9$
- $k_7 = 1.12 \times 10^{12}$

Como se aprecia el número de combinaciones aumenta abrumadoramente y para el caso de estimación DOA cada combinación correspondiente a k_i se considera una clase de entrenamiento y en [18] se sugiere que el aprendizaje supervisado requiere aproximadamente 5000 ejemplos de entrenamiento por clase. Incluso con los modelos más sofisticados de arquitecturas que tengan la capacidad requerida, el poder computacional necesario para obtener la cantidad de ejemplos de entrenamiento y realizar el entrenamiento limitan el número de fuentes considerados en un escenario en la práctica.

Resumen

En este capítulo se mencionaron algunos de los beneficios de incluir una NN en esta tesis. Se explicaron las dos grandes clases de arquitecturas de redes que existen, haciendo notar que no existe una teoría bien desarrollada para la selección y optimización de la arquitectura de una NN requerida para interactuar en un entorno de interés. La representación del conocimiento también influencia sobre la elección de la arquitectura, siendo un ejemplo de esto las redes convolucionales CNNs, las cuales incorporan conocimiento previo del entorno o tarea y la técnica de compartición de pesos [27]. Se hizo notar que la arquitectura menos compleja para el caso de NNs profundas son las MLPs y mediante el análisis de varios artículos se resaltó su viabilidad para emplearla en la estimación DOA. Además, se trataron aspectos cruciales en el entrenamiento de las NNs para la estimación DOA donde quedó establecido la importancia del procesamiento adecuado de los datos de entrada a la red, el número de fuentes y de ejemplos etiquetados por clases como determinantes en la complejidad y las capacidades de la NN, lo que es un aspecto importante en los objetivos de esta tesis.

Capítulo 4

Diseño e Implementación del Sistema Neuro-Detector DOA

El objetivo de este trabajo es implementar un algoritmo de estimación DOA integrando BF y NN con la menor complejidad y requerimiento computacional posible. Primeramente, en la sección 4.1 se crean las bases y se fijan las ideas generales del funcionamiento del algoritmo y la implementación propuestos. Luego, en la sección 4.2, se fija la estructura del Bf empleado y se describe cómo se crearon las restricciones para el filtrado espacial. Posteriormente en la sección 4.3 se detalla el estimador empleado y en las subsecciones 4.4.1 y 4.4.3 se debate sobre la elección del modelo de arquitectura de la NN y el entrenamiento respectivamente. Los detalles sobre la generación de los datos, la detección de actividad de voz empleada y la simulación de los datos se discuten la sección 4.5 y las subsecciones 4.5.1 y 4.5.2 respectivamente. En la subsección 4.4.2 se abordan la forma en que se procesaron los datos alimentados a la red.

4.1. Generalidades del diseño del Neuro-Detector

El sistema de detección de DOA de señales de voz que se implementó en este trabajo está constituido por tres bloques funcionales fundamentales como se muestra en figura 4.1. Un primer bloque donde las muestras de las señales de voz obtenidas a la salida del arreglo de micrófonos son procesadas y se obtiene la matriz de correlaciones entre ellas, a la vez que quedan establecidas las matrices de restricciones asociadas a cada LCMV y que expanden el espacio de representación de las señales de interés provenientes de cada dirección de arribo considerada. En el segundo, se emplea una NN MLP para estimar las variaciones estadísticas y el ruido y así estimar el subespacio de señal y el subespacio de solo-ruido. Una vez delimitados los subespacios, el tercer bloque proyecta cada espacio de representación de las señales sobre el subespacio de señal determinado por la NN, para así calcular y comparar la potencia asociada a cada LCMV; siendo la mayor un indicador de la mejor estimación DOA.

A lo largo de todo el diseño e implementación se consideró un arreglo de $M = 8$ micrófonos y en la totalidad de las simulaciones e implementaciones realizadas para este capítulo, la geometría del arreglo de micrófonos considerada fue lineal con espaciado uniforme (Uniform Linear Array (ULA)). Sin embargo, cabe destacar que el algoritmo no es restrictivo a esta única configuración y que, tomando en cuenta las especificidades correspondientes se puede emplear con arreglos de micrófonos con otras configuraciones, como podría ser circular uniforme (Uniform Circular Array (UCA))(sección 5.2). Además la configuración del Bf es tal que se emplean líneas de retardo de muestras (Sample Delay Lines (SDL)) o filtros FIR de orden $J = 48$ por cada canal M (vea sección 2.2).

La selección del orden J de los filtros FIR o SDL de cada canal fue hecha basado en las simulaciones que se realizaron para esta tesis y con las siguientes premisas:

- el número de restricciones que conforma cada matriz de restricción se toma con base al producto Tiempo-Ancho de banda de la señal en dicha dirección (D_θ), es decir, $r \geq D_\theta$ (subsección 2.4.3)
- las magnitudes de los eigenvalores correspondientes a la base del espacio de representación caen dramáticamente para valores mayores al producto Tiempo-Ancho de banda de la señal [11],[12]
- el empleo de aproximaciones de rango bajo de la matriz de restricciones, como el que se hace en este trabajo, es factible siempre y cuando el rango numérico de la matriz sea menor a MJ [12]

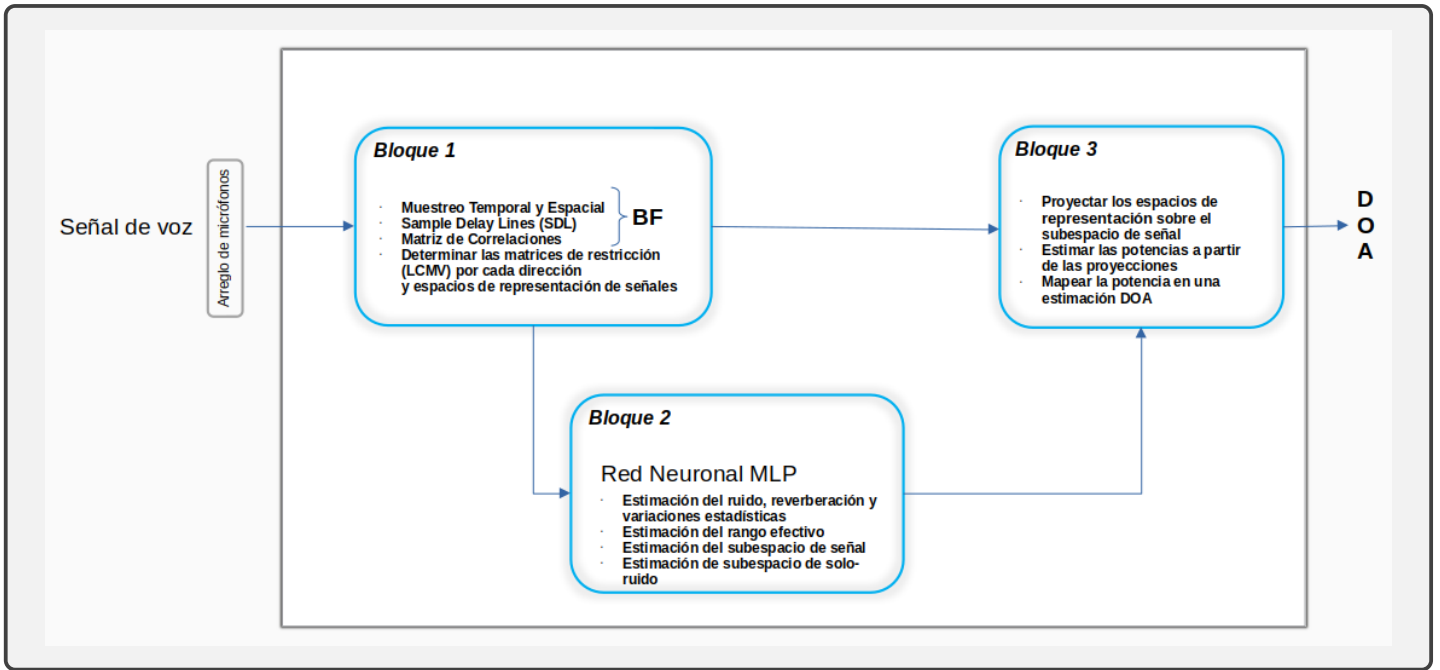


Figura 4.1: Diagrama de bloques general del Neuro-Detector

- garantizar que la norma del vector de pesos \mathbf{w} no sea innecesariamente grande y por tanto disminuir la ganancia de ruido blanco y aumentar la SNR [12]

Las cuestiones anteriores se garantizan asegurando la desigualdad $D_\theta \leq r \ll MJ$ mediante una elección apropiada de J . De modo que J se eligió para granatizar que MJ sea lo suficientemente grande para cumplir la desigualdad pero no demasiado como para requerir una mayor complejidad computacional debido al aumento considerable de las operaciones necesarias.

La implementación del algoritmo Neuro-Detector DOA tiene la característica que realiza una estimación cada medio segundo de señal de voz recibida. Esto se hizo con la intención de dotar al sistema con la capacidad de lograr hacer estimaciones a partir de una matriz de autocorrelaciones $\mathbf{R}_{\mathbf{x}\mathbf{x}}$ conformada con un número relativamente pequeño de muestras como corresponde a medio segundo. Además podría considerarse una desventaja si para lograr una estimación DOA se necesitara segmentos completos de voz, es decir, una cantidad de muestras correspondiente a una expresión coherente o una palabra completa, ya que mientras el hablante completa una palabra o expresión bien pudiera estarse desplazando; por otro lado, la cantidad de muestras correspondientes pudiera ser considerablemente grande tomando en cuenta que además se multiplica por la cantidad de canales del arreglo, que para este caso es $M = 8$. Así, con esta característica de implementación el Neuro-Detector puede rastrear fuentes de voz en movimiento y hacerlo sin requerir demasiadas muestras ni tiempo. El tiempo de medio segundo fue elegido ya que se considera que es un tiempo menor al requerido por una persona para desplazarse de un punto a otro (en caso de que la fuente de voz sea una persona) o bien un tiempo que genera poco desplazamiento en caso que la fuente no sea una persona. Esta característica de medio segundo de estimación, sumada a la detección de actividad de voz (Voice Activity Detection (VAD)) empleada en la generación de los datos de entrenamiento y validación (subsección 4.5.1) garantizan la escalabilidad del sistema para implementaciones en tiempo real.

Por último cabe destacar que en el algoritmo Neuro-Detector implementado se diseña un conjunto de restricciones lineales, del mismo modo que se procedería para un Bf LCMV, sobre la banda de frecuencia de interés por cada dirección de arriba supuesta, pero en vez de calcular y comparar la potencia asociada a cada Bf según las ecuaciones 2.47 y 2.48 para estimar el DOA, en su lugar, se proyectan los espacios de representación expandidos por los vectores de restricción que conforman la matriz de restricciones resultante del diseño por eigenvectores (sección 2.4.3) sobre los subespacios de señal de la matriz de correlaciones para cada dirección. De esta forma se considera que este algoritmo es una combinación de dos métodos convencionales de estimación DOA (subsección 2.5.1): el método basado en BF y el método

basado en subespacios.

4.2. Implementación del Bf y filtrado en el espacio de representación

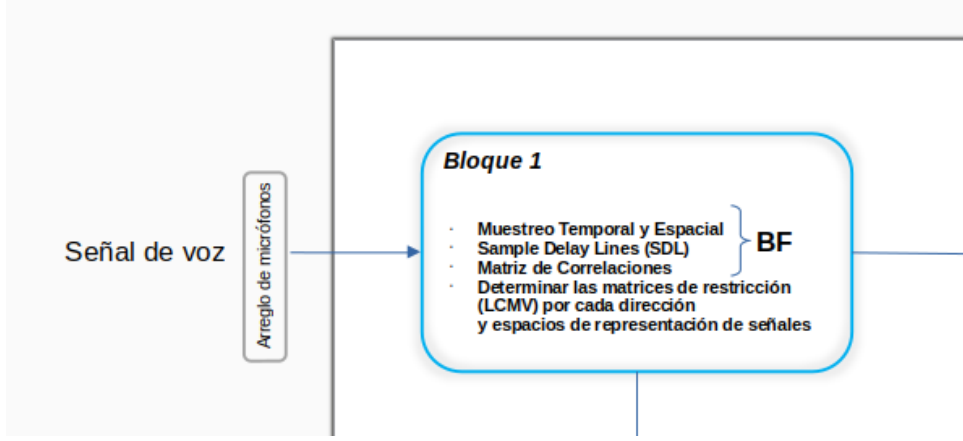


Figura 4.2: Bloque funcional 1, correspondiente al Bf y al filtrado en el espacio de representación.

Basado en un arreglo ULA con $M = 8$ micrófonos y con filtros FIR de orden $J = 48$ en cada canal y tomando una resolución de dirección de arriba de un grado se muestreó uniformemente el intervalo de direcciones $\theta = -90^\circ$ a 90° con 181 puntos θ_i , $i = 0, 1, \dots, 180$ y se diseñó un conjunto de 181 matrices ($m \times n = MJ \times r_{\theta_i}$) de restricciones lineales, una por cada dirección, sobre la banda de frecuencia de audio normalizada utilizando el acercamiento de restricciones por eigenvectores descrito en la subsección 2.4.3.

Para llevar a cabo el filtrado en el espacio de representación de las señales se considera el problema de controlar la respuesta de cada LCMV en la dirección θ_i correspondiente sobre la frecuencia angular normalizada en la banda de frecuencia audible. Para la implementación se consideró el caso de señales senoidales $\cos(\omega_j t)$ en dicha banda, incidiendo desde cada θ_i , lo cual es correcto ya que estas señales pertenecen al mismo espacio de representación. Para un par localización/frecuencia (θ_i, ω_j) en la banda audible, el vector de señal \mathbf{x} (ecuación 4.1) de dimensión MJ obtenido a la salida del Bf y las restricciones para esta señal (ecuaciones 4.2, 4.3, 4.4), suponiendo que la respuesta deseada presenta una ganancia unitaria y un retardo puro T_0 , quedarían como sigue [11]:

$$\begin{aligned} \mathbf{x}(\theta_i, \omega_j) = & [\cos(\omega_j(t - \tau_0)) \dots \cos(\omega_j(t - \tau_{M-1})) \\ & \cos(\omega_j(t - (\tau_0 + T_s))) \dots \cos(\omega_j(t - (\tau_{M-1} + T_s))) \\ & \dots \cos(\omega_j(t - (\tau_0 + (J-1)T_s))) \\ & \dots \cos(\omega_j(t - (\tau_{M-1} + (J-1)T_s)))]^T \end{aligned} \quad (4.1)$$

donde $\tau_m = m\tau_1$, τ_1 se define en la sección 2.2, $T_s = 1/f_s$, siendo $f_s = 16000$ la frecuencia de muestreo que se eligió en la implementación y $\Omega_j = \omega_j T_s$ la frecuencia normalizada,

$$\begin{bmatrix} \mathbf{x}_c^T(\omega_j) \\ \mathbf{x}_s^T(\omega_j) \end{bmatrix} \mathbf{w} = \begin{bmatrix} \cos(\Omega_j T_0 / T_s) \\ \sin(\Omega_j T_0 / T_s) \end{bmatrix} \quad (4.2)$$

donde:

$$\mathbf{x}_c(\omega_j) = [1 \dots \cos(\Omega_j(M-1)\mu \sin\theta_i) \dots \cos((J-1)\Omega_j) \dots \cos(\Omega_j((M-1)\mu \sin\theta_i + (J-1)))]^T \quad (4.3)$$

$$\mathbf{x}_s(\omega_j) = [1 \dots \sin(\Omega_j(M-1)\mu \sin\theta_i) \dots \sin((J-1)\Omega_j) \dots \sin(\Omega_j((M-1)\mu \sin\theta_i + (J-1)))]^T \quad (4.4)$$

donde $\mu = d/(cT_s)$ y $j = 0, 1 \dots, \hat{r}_{\theta_i}$, quedando \hat{r}_{θ_i} definido como se explica en la subsección 2.4.3. La matriz \mathbf{C}_{θ_i} de restricciones, correspondiente a la dirección θ_i , quedará formada al disponer los vectores de restricción $\mathbf{x}_c(\omega_j)$ y $\mathbf{x}_s(\omega_j)$ obtenidos de cada una de las \hat{r}_{θ_i} frecuencias angulares seleccionadas $\{\omega_j \in BW_{\theta_i}, j = 0, 1 \dots, \hat{r}_{\theta_i}\}$ y antes de descomponerla en sus valores singulares (subsección 2.4.3) como se muestra en la siguiente ecuación:

$$\mathbf{C}_{\theta_i} = \left[\mathbf{x}_c(\omega_0) \dots \mathbf{x}_c(\omega_{\hat{r}_{\theta_i}}) \mid \mathbf{x}_s(\omega_0) \dots \mathbf{x}_s(\omega_{\hat{r}_{\theta_i}}) \right] \quad (4.5)$$

4.3. Estimador DOA

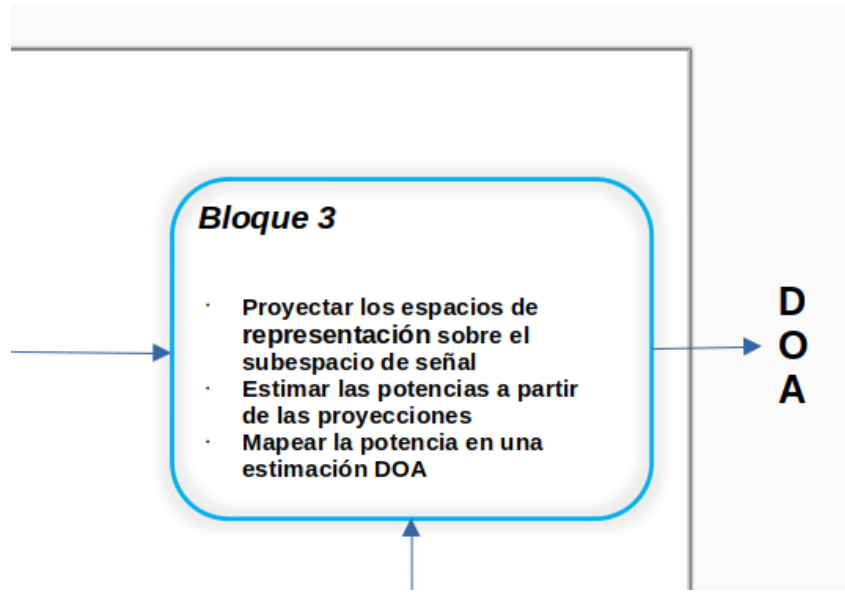


Figura 4.3: Bloque funcional 3, correspondiente al estimador DOA

La matriz de correlaciones obtenida a partir de la señal observada a la salida de la estructura Bf (la simulación de las muestras obtenidas se aborda en la sección 4.5) implementada en la sección 4.2 está dada por $\mathbf{R}_{\mathbf{xx}} = \mathbf{R}_s + \sigma_n^2 \mathbf{I}$, donde \mathbf{R}_s es la matriz de correlaciones espacio-temporal de señal solamente. Los eigenvalores de $\mathbf{R}_{\mathbf{xx}}$ son $[\lambda_i; i = 1, \dots, MJ; \lambda_i \geq \lambda_{i+1}]$ y los correspondientes eigenvectores son $[\mathbf{e}_i; i = 1, \dots, MJ]$. Definiendo el rango efectivo de la matriz \mathbf{R}_s como D_s , los subespacios de señal y solo-ruido quedan definidos como los espacios expandidos por los eigenvectores $[\mathbf{e}_i; i = 1, \dots, D_s]$ y $[\mathbf{e}_i; i = D_s + 1, \dots, MJ]$ respectivamente.

Para generar el contorno de localizaciones de fuentes se genera un estimado de la potencia en cada θ_i de interés. Para cada θ_i la base ortogonal $[\mathbf{v}_j; j = 1, \dots, r_\theta]$ (r_θ se define en la subsección 2.4.3) del espacio de representación que conforma la matriz de restricciones en la banda audible correspondiente a dicha dirección, \mathbf{C}_{θ_i} , se usa como modelo de la fuente para así proyectarlo sobre el subespacio de señal. Para la implementación, de entre los varios estimadores que se pueden usar, se empleó el siguiente:

$$\mathbf{P}_\theta = \frac{1}{r_\theta} \sum_{j=1}^{r_\theta} \left[1 - \sum_{i=1}^{D_s} |\mathbf{v}_{j,\theta}^T \mathbf{e}_i|^2 \right]^{-1} \quad (4.6)$$

Este estimador fue seleccionado debido a que obtiene valores relativamente altos si algún vector base de representación yace en el subespacio de señal. Es más robusto ante problemas de escala en la potencia estimada para varias señales que inciden desde diferentes direcciones, dado por el hecho de que el número de vectores base que yacen cerca del subespacio de señal para cada señal incidente es diferente. Además presenta muy buena resolución [10].

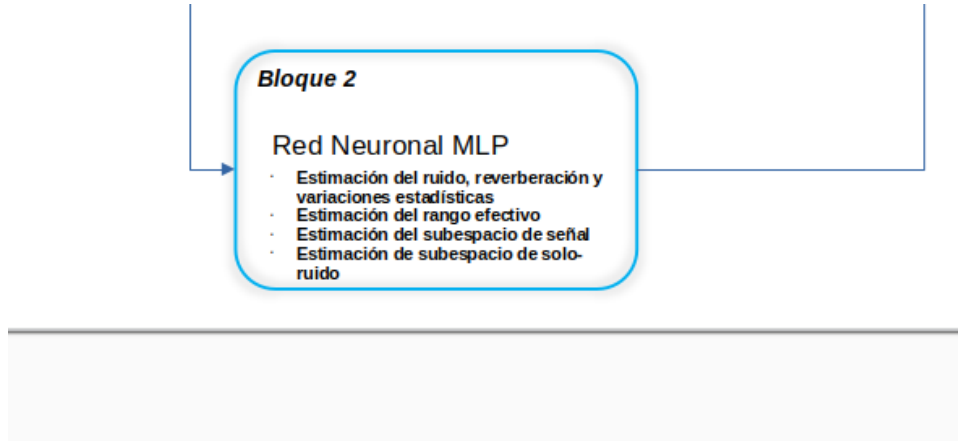


Figura 4.4: Bloque funcional 2, correspondiente a la red neuronal.

4.4. Red Neuronal

En la sección anterior se describió el estimador DOA empleado y quedó claro que depende de determinar correctamente el rango efectivo D_s de la matriz \mathbf{R}_{xx} para poder proyectar sobre el subespacio de señal. Para llevar a cabo esta tarea se decidió emplear una NN, que inspirado en el análisis del GSC y del Neuro-GSC descritos en las subsecciones 2.4.4 y 3.4 respectivamente, deberá ser capaz de estimar las variaciones estadísticas y el ruido a partir de la matriz de correlaciones espacio-temporal \mathbf{R}_{xx} de la señal y determinar el rango efectivo.

4.4.1. Modelo de Arquitectura Multilayer Perceptron

En este trabajo la NN debe ser capaz de estimar el rango efectivo a partir de entradas de $MJ \times N$ donde N está determinado por el preprocesamiento de los datos de entrada (subsección 4.4.2). El modelo podría haber sido pensado para realizar una tarea de regresión que obtuviera como salida un valor correspondiente al rango de la matriz, lo cual es viable debido a que es un número entero y por tanto tiene resolución finita. El problema es que habría que emplear una función de activación en la capa de salida capaz de mapear valores de salida del orden de la matriz de entrada, esto implica muy posiblemente emplear normalizaciones por lotes en las capas intermedias, lo que agrega procesamiento extra y mayor complejidad a la red. Se prefirió entonces que el modelo realizara una tarea de clasificación, donde cada clase representa un valor de rango efectivo y la resolución estaría determinada por el número de salidas, que sería igual a MJ , es decir, el orden de la matriz. De esta forma, la función de activación en la capa de salida otorgaría valores de probabilidad y sería la misma función para cada capa, lo que hace más simple su diseño e implementación.

Ambos modelos de arquitecturas, MLP y CNN, se consideraron para la implementación tomando en cuenta el análisis de varios artículos (ver subsección 3.5) y al ser estos dos los más empleados para la estimación DOA. Uno de los objetivos de la tesis es integrar una NN con la menor complejidad y requerimiento computacional posibles y como se abordó en el capítulo 3, el modelo MLP de redes profundas es el más sencillo comparado con el resto, de ahí que se eligiera para este trabajo. El número de nodos a la entrada, la cantidad de HLs y la estructura de la MLP estarán determinados por la naturaleza y la cantidad de los datos alimentados a la red. De esta forma, en la siguiente sección se discuten dos opciones consideradas para los datos de entrada y se muestran las estructuras y el resumen de las características de la MLP en cada caso. Se consideraron dos opciones para los datos de entrada: alimentar eigenvectores de la matriz de correlaciones \mathbf{R}_{xx} , o bien, alimentar eigenvalores de \mathbf{R}_{xx} .

En caso de ser alimentada con eigenvectores, cambiando el número de HLs se observó que el modelo no tuvo la capacidad suficiente para generalizar con una sola HL y que con tres y cuatro HLs no mejoró el rendimiento sobre los datos de prueba y solo se añaden más parámetros al modelo. La MLP implementada para este caso se muestra en la figura 4.5 y sus parámetros se resumen en la tabla 4.1.

Para el caso de alimentar la red con eigenvalores, con una sola HL generalizó exitosamente, mientras que aumentando el número de HLs no se logró una mejora significativa en el rendimiento y por tanto se incrementa la complejidad de la red innecesariamente. La MLP implementada para este caso se muestra

en la figura 4.6 y sus parámetros se resumen en la tabla 4.2.

En ambos casos, el número de coeficientes también fue cambiado a modo de comparación y no cambió demasiado la precisión o el error; los detalles de estas pruebas se presentan en la subsección 4.4.4. La función de activación empleada en todos los nodos de cada HL y la capa de salida fue la función sigmoide o logística, por ser muy utilizada, además de simple. El algoritmo de optimización de descenso más inclinado o descenso del gradiente fue empleado como optimizador por ser el más sencillo aplicado en el aprendizaje profundo, además [18] sugiere que ningún algoritmo de optimización, como bien pudiera ser el de momentos adaptivos (Adaptive Movement Estimation algorithm (Adam)), Root Mean Square Propagation (RMSProp) o Stochastic Gradient Descent (SGD), supera las demás opciones, de modo que la opción depende de la confianza del usuario.

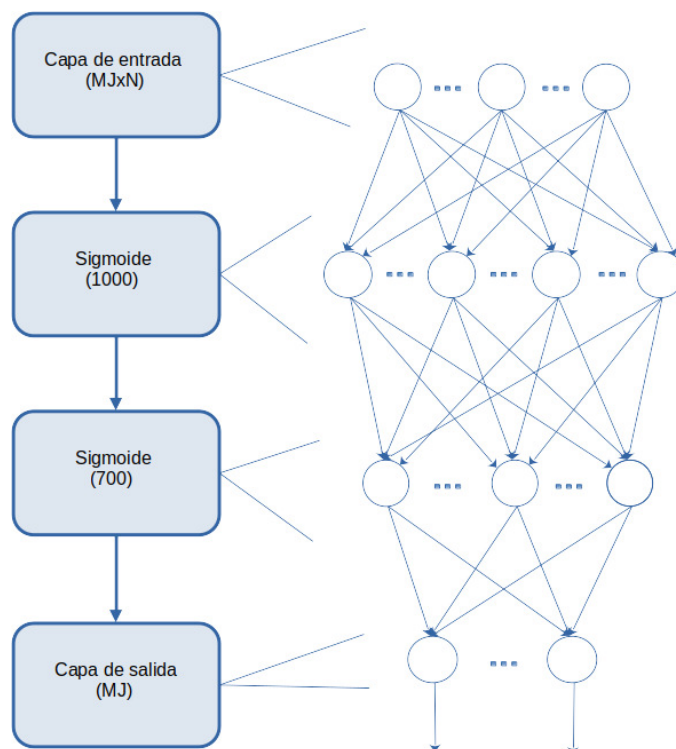


Figura 4.5: Gráfico de la MLP para el caso de alimentar eigenvectores de la matriz \mathbf{R}_{xx} .

Parámetro	Valor
Función de costo	Cuadrado de la diferencia
Optimizador	Gradient descent
Razón de aprendizaje	0.1
Épocas	5
Métrica	Precisión
Funciones activación	Sigmoide-Logística
Número de capas ocultas	2
Número de coeficientes de la MLP	58568800

Tabla 4.1: Parámetros de la MLP para el caso de alimentar eigenvectores de la matriz \mathbf{R}_{xx} .

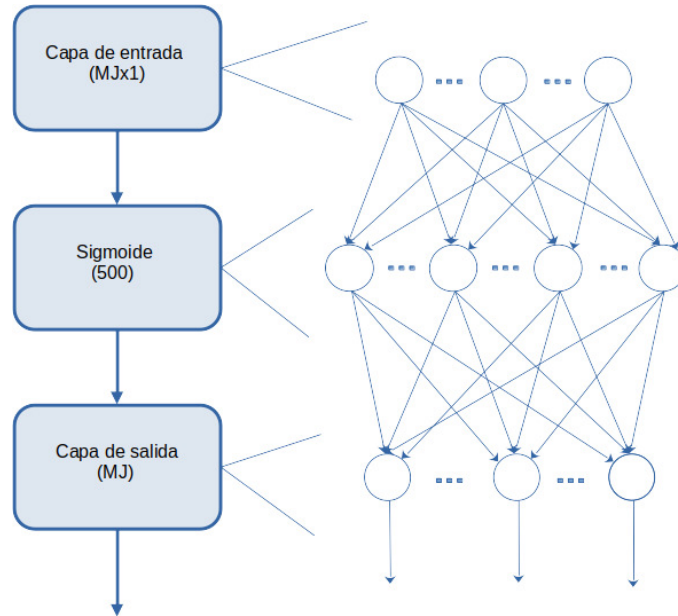


Figura 4.6: Gráfico de la MLP para el caso de alimentar eigenvalores de la matriz $\mathbf{R}_{\mathbf{xx}}$.

Parámetro	Valor
Función de costo	Cuadrado de la diferencia
Optimizador	Gradient descent
Razón de aprendizaje	0.3
Épocas	4
Métrica	Precisión
Funciones activación	Sigmoide-Logística
Número de capas ocultas	1
Número de coeficientes de la MLP	384000

Tabla 4.2: Parámetros de la MLP para el caso de alimentar eigenvalores de la matriz $\mathbf{R}_{\mathbf{xx}}$.

4.4.2. Extracción de características. PCA

Para este trabajo no es necesario procesar los datos a la salida de los canales del arreglo de micrófono en una forma que representen la información DOA, porque la NN no entrega directamente el DOA. En su lugar es alimentada la matriz de correlaciones espacio-temporal $\mathbf{R}_{\mathbf{xx}}$. Del análisis de la estructura Bf presentada en la sección 4.2 se desprende que el orden de $\mathbf{R}_{\mathbf{xx}}$ es $MJ \times MJ$ lo que para $M = 8$ y $J = 48$ constituye una capa de entrada en la MLP de 147456 nodos. Tomando como base el conocido algoritmo Principal Component Analysis (PCA) se procesó la matriz antes de ser alimentada a la red con el fin de reducir su dimensionalidad con la menor pérdida de información posible. El algoritmo PCA logra reducir la dimensión de una matriz de datos y logra una representación linealmente independiente de esta a partir de los eigenvectores correspondientes a los mayores eigenvalores de la matriz de varianzas y covarianzas asociada a los datos. De forma alternativa, y con más interés para nuestro caso, PCA se puede realizar también a partir de las variables estandarizadas y emplear las correlaciones en lugar de las covarianzas, de esta forma los componentes principales normados se obtienen calculando los valores y vectores propios de la matriz $\mathbf{R}_{\mathbf{xx}}$ [18],[40].

Así con el fin de reducir el volumen de cálculo computacional requerido y la complejidad de la MLP, esta es alimentada con los $N = 150$ primeros eigenvectores de $\mathbf{R}_{\mathbf{xx}}$ correspondientes a los 150 primeros valores propios. Los eigenvalores de $\mathbf{R}_{\mathbf{xx}}$ son de la forma $[\lambda_i ; i = 1, \dots, MJ ; \lambda_i \geq \lambda_{i+1}]$. De

esta manera en vez de emplear una MLP con 147456 nodos de entrada y en consecuencia, mayor número de nodos ocultos para poder extraer las características necesarias y mayor requerimiento computacional; la complejidad y el tamaño de la MLP se redujo al contar, en comparación, con 57600 nodos de entrada. El valor 150 fue seleccionado como resultado de los experimentos realizados. La MLP en este caso, debe ser capaz de, a partir de los eigenvectores alimentados, determinar las variaciones estadísticas y estimar el rango que corresponde a la dimensión del subespacio que genera la señal de voz muestreada y descrita por la matriz \mathbf{R}_{xx} .

Una mayor reducción en el número de nodos y por tanto en la complejidad de la MLP se logra si en vez de alimentar los 150 eigenvectores más significativos se alimentaran la totalidad de los eigenvalores de \mathbf{R}_{xx} . El proceso de cálculo de los eigenvalores no acarrea un proceso de cálculo adicional pues la eigendescomposición es de igual modo obligatoria para determinar los subespacios de señal y solo-ruido. Así, alimentando los eigenvalores, la MLP reduce el número de nodos de entrada a solo $MJ = 384$ y $N = 1$, y por tanto, la cantidad de nodos ocultos necesarios para extraer las características y estimar el ruido y las variaciones estadísticas es menor. En este caso la MLP debe ser capaz de determinar cuales eigenvalores son lo suficientemente pequeños o cercanos a ceros como para ser considerados como eigenvalores del subespacio de solo-ruido y de esta forma determinar el rango efectivo.

4.4.3. Entrenamiento y selección de parámetros

El modelo deberá ser entrenado bajo un algoritmo de aprendizaje supervisado donde a un conjunto de entradas de audio se les asignó etiquetas que representan la dirección de arriba en grados con respecto al costado del arreglo lineal. El aprendizaje supervisado fue elegido debido a que la MLP realiza una tarea de clasificación y no solo agrupa entradas. Un aspecto muy importante a notar es que la MLP no entrega directamente los valores DOA, sino que, como se ha explicado en la figura 4.1, es intermediaria en este proceso. De esta forma las etiquetas de los ejemplos de entrenamiento brindan el DOA, sin embargo la MLP determina el rango efectivo de la matriz de correlaciones a la salida del Bf (sección 4.3), por tanto para el entrenamiento y la retropropagación, el error que se tomó a la salida de los nodos de la capa de salida es determinado de la misma forma que se determinan los errores de las HLs en la MLP ya que las etiquetas de los ejemplos de entrenamiento no brindan el target o respuesta deseada a la salida de la red, es decir, el rango efectivo correcto. En este sentido, en el proceso de retropropagación, para el ajuste de los parámetros de la red, el error es dividido y asignado entre las interconexiones según su contribución; es decir, mientras mayor es el valor del coeficiente o peso asociado a la conexión, mayor el valor de error que acarrea hacia atrás.

Para el entrenamiento, los 3000 conjuntos de datos simulados fueron divididos en 2700 (90 %) datos de entrenamiento y 300 (10 %) datos de prueba. Cada conjunto de datos es preprocesado por el Bf para determinar la matriz de correlaciones espacio-temporal \mathbf{R}_{xx} que se alimenta a la MLP. Esta matriz a su vez es preprocesada para asegurar que sus valores no saturan la red, ya que debido a la función logística, valores muy grandes de entrada en los nodos que conforman las HLs y la capa de salida limitarían la habilidad de aprendizaje de la red, lo mismo sucedería si la mayoría o la totalidad de las entradas fuesen cero. Esto se explica con base al proceso de retropropagación abordado en la sección 3.1, ya que al ajustar los parámetros o pesos de la red, mediante el algoritmo de descenso del gradiente, los nuevos pesos son obtenidos empleando el gradiente de la función de activación logística y las salidas de los nodos de la capa siguiente; por tanto, alimentar la MLP con valores muy grandes o muy pequeños obliga a la función logística a acercarse demasiado a sus asíntotas, por ende, su gradiente será muy pequeño y la habilidad de aprendizaje será casi nula; lo mismo sucede si los valores de salida de las capas son muy cercanos a cero o iguales a cero. Así, tanto para los procesos de entrenamiento, evaluación de la red y empleo del Neuro-Detector es necesario acondicionar los eigenvectores o los eigenvalores de la matriz \mathbf{R}_{xx} , según el caso, mediante la normalización y escalamiento de sus valores antes de ser pasada a la MLP.

El modelo fue corrido para el mismo dataset de 300 ejemplos etiquetados, y se llevaron a cabo pruebas y comparaciones para determinar la razón de aprendizaje y el número de épocas en el entrenamiento que mejor rendimiento logran para el sistema. En la siguiente sección se muestran algunos resultados y la idea general del procedimiento de selección de dichos parámetros.

4.4.4. Análisis y selección de parámetros

Para la selección de los parámetros de la MLP se corrió el dataset de 300 ejemplos de entrenamientos sobre varios modelos de arquitectura para después obtener los desempeños del sistema Neuro-Detector correspondiente sobre el mismo conjunto de validación. De esta manera se seleccionó el factor de aprendizaje y las épocas para la MLP. Para medir el desempeño en estas pruebas se tomó el error absoluto entre la respuesta del Neuro-Detector y la etiqueta en cada ejemplo que se alimenta. Si este error es menor o igual a tres grados entonces se consideró como una estimación exitosa y se asigna el valor de uno; de otro modo, como errónea y se le asigna el valor de cero. Al conjunto de todas las respuestas obtenidas de correr el dataset se le halla la media y esta es el desempeño del sistema.

En la figura 4.7 se muestran los resultados del sistema para las dos opciones de los datos de entrada considerados en la sección 4.4.2 y varios valores de razón de aprendizaje.

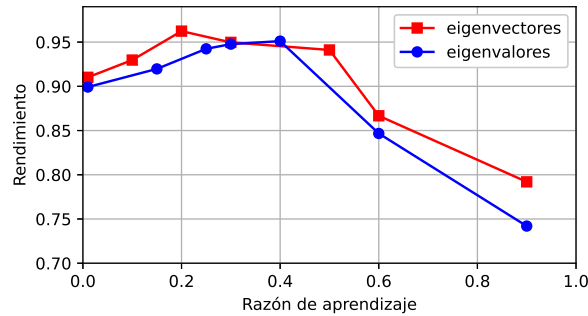


Figura 4.7: Desempeño y razón de aprendizaje para ambas opciones de datos de entrada

Como se aprecia para valores más pequeños, como 0.01, se obtienen peores valores de desempeño y esto pudiera ser debido a que se ralentiza la velocidad a la que se hace el descenso del gradiente y se limita la experiencia obtenida de cada ejemplo de entrenamiento; esto es perjudicial ya que como se observa la MLP no generaliza correctamente. Un aspecto a destacar es que este tipo de acercamiento se debería hacer muchas veces para una misma razón de aprendizaje o cualquier parámetro que se analice, ya que solo así se reducen los efectos de aleatoriedad y de malos caminos durante la selección de los pesos iniciales y el estancamientos en mínimos locales durante el descenso del gradiente. Sin embargo, para la implementación de este trabajo fue útil y suficiente para identificar el intervalo de razón de aprendizaje de mejor desempeño.

Del mismo modo se procedió para elegir el número de épocas. A pesar que seleccionar un número mayor de épocas significa un mayor tiempo de entrenamiento, esto ayuda a corregir los pesos “rezagados” en el descenso del gradiente y así se logra un mejor mínimo y por tanto mejor desempeño en la mayoría de casos. En este proceso también se corroboró que un número elevado de épocas resulta en un desempeño menor debido a que la red se sobreajusta a los datos de entrenamiento y luego no obtiene buenos resultados para nuevos datos que no ha “visto” antes. Este sobreajuste es algo a tener en cuenta en NN y ML en general [18].

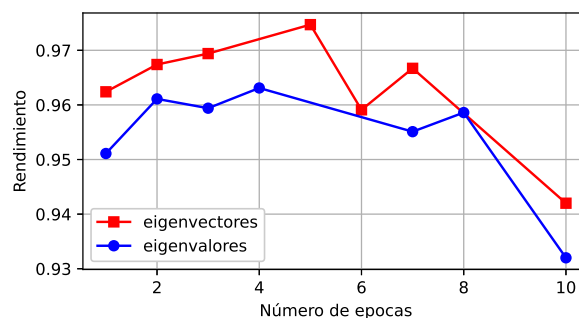


Figura 4.8: Desempeño y número de épocas para ambas opciones de datos de entrada

Se puede ver que los resultados para el caso de selección de las épocas no es muy predecible. Además, para 6 épocas en el caso de alimentar la red con eigenvectores hay una caída del desempeño y esto puede

ser debido a una mala corrida, en la que la NN quedó atascada en un mínimo local durante el descenso. Sin embargo, debido al hecho que no se realizaron muchos experimentos para cada elemento de selección estas variaciones son esperadas debido al carácter aleatorio intrínseco de las NNs.

Siguiendo los objetivos de implementar una NN con el menor requerimiento computacional posible y corroborando de las gráficas 4.7 y 4.8 que la diferencia del rendimiento obtenido entre las dos opciones de MLP consideradas no es muy distante; se decidió seleccionar la MLP resultante de alimentar los eigenvalores de la matriz de correlaciones. Por último, se destaca que los experimentos para la selección de las épocas fueron realizados considerando la razón de aprendizaje obtenida de los experimentos de la figura 4.7, es decir: **0.2-eigenvalores** y **0.4-eigenvalores**. No obstante, debido a que el aumento del número de épocas significa explorar en mayor medida el descenso del gradiente, entonces se puede tomar pasos más pequeños, es decir, menor razón de aprendizaje y por ende encontrar un mejor camino a un mejor mínimo local o al mínimo global. Así, el valor final de la razón de aprendizaje para ambos casos de la MLP es menor a los mostrados en la figura 4.7 y son los que se recogen en las tablas 4.1 y 4.2. En la figura 4.9 se muestran los resultados para el caso de alimentar los eigenvalores y se observa la veracidad de esta idea; de modo que se disminuye la razón de aprendizaje a **0.3** dado que se realizan **4** épocas en el entrenamiento.

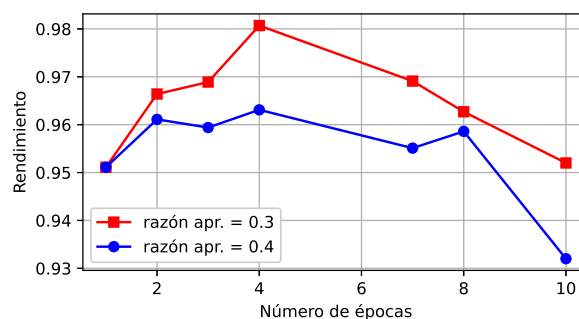


Figura 4.9: Desempeño para un mismo número de épocas y diferentes razones de aprendizaje.

4.5. Generación de los datos

En aras de entrenar la red MLP para determinar el rango efectivo de la matriz \mathbf{R}_{xx} es requerida una cantidad considerable de datos de entrenamiento. El trabajo presentado en [18](p.20-21) sugiere que el aprendizaje supervisado requiere aproximadamente 5000 datos de entrenamiento por clase y que resultados excepcionales se logran para 10 millones. Las tres vías posibles para tener acceso a esta cantidad de datos es mediante la simulación de estos, grabando datos reales o encontrando un dataset existente. Grabar u obtener los datos reales mediante el hardware idóneo para la implementación tiene como beneficios que el modelo será entrenado con datos similares a los que deberá clasificar, lo que dotará al modelo con el mejor rendimiento, pero tiene dos limitantes fundamentales. Primeramente, grabar los datos de voz requiere mucho tiempo, en particular cuando el número de clases, que en este caso corresponde a la resolución, aumenta. En segundo lugar, una disminución del desempeño es esperada si el arreglo de micrófonos es colocado en una habitación diferente, o incluso en una posición diferente en la misma habitación donde los datos de voz de entrenamiento fueron grabados. Esto pudiera ocurrir debido a que la respuesta al impulso de la habitación sería diferente y el modelo pudiera no generalizar correctamente esta nueva entrada. Otra opción sería encontrar un dataset existente con datos de voz pregrabados y que además contara con el número suficiente de datos para el entrenamiento. Las limitaciones de este acercamiento es que las grabaciones deberían cumplir con los requerimientos de esta tesis, dígame, hasta el momento un arreglo lineal uniforme con $M = 8$ canales y un espaciado intermicrófonos de $d = 0.04287$, además el rendimiento pudiera ser malo dependiendo de las condiciones de la habitación y el escenario en que fueron grabadas. Una solución intermedia es simular los datos de entrenamiento empleando un software, lo que brinda la facilidad de crear un conjunto de datos que sean lo más compatible posible a las necesidades de la tesis, pues da la libertad de cambiar parámetros como la cantidad de reverberación, la distancia entre las fuentes de voz y el arreglo de micrófonos, el tipo de arreglo empleado, la SNR recibida, entre otros. Sin embargo, las simulaciones también requieren cierta cantidad de tiempo y pueden no tener el mejor desempeño una vez que se pruebe el dispositivo en un escenario real.

4.5.1. Detección de actividad de voz

Una cuestión de importancia considerada en la generación de datos y el entrenamiento de la MLP fue el hecho que la matriz \mathbf{R}_{xx} obtenida a partir de solo ruido ambiente y sin presencia de voz no podría ser usada para delimitar entre subespacios de señal y solo-ruido pues solo habría contribución del ruido y solo un subespacio estaría presente, por tanto es necesario garantizar que para el entrenamiento haya presencia en todo momento de información correspondiente a la voz. Además, una característica deseable para este trabajo fuera lograr su aplicación en tiempo real. Tomando en cuenta lo anterior es necesario una precisa detección de actividad de voz (VAD) para alimentar solo segmentos de voz al algoritmo desarrollado y preparar el dataset de entrenamiento. La interfaz de programación de aplicaciones (API) WebRTCvad Python [35] fue seleccionada para esta tarea, esta fue desarrollada por Google para el proyecto WebRTC, el cual ayuda a desarrolladores a crear servicios de comunicación web en tiempo real. El WebRTC VAD solo acepta audio PCM mono a 16-bit muestreado a 8000, 16000, 32000 ó 48000 Hz . Un frame o segmento debe tener una duración de 10, 20 ó 30ms. Mediante inspección del código fuente [36], se aprecia que se emplean modelos de mezcla Gaussiana para determinar la probabilidad y clasificar si el segmento es voz o ruido de ambiente. Para esta tesis se creó una función en Python que toma una grabación de audio y retorna los segmentos de voz para la generación de datos.

4.5.2. Simulación de los datos

Para generar audio capturado por un arreglo lineal uniforme de 8 micrófonos un dataset de pronunciations o voces es necesario, así como un software que pueda estimar las respuestas al impulso de la habitación (Room Impulse Response (RIR)) para los 8 micrófonos. Entonces las muestras de audio de la fuente son convolucionadas con la RIR apropiada para obtener la entrada de audio de cada micrófono. Cualquier dataset con voces grabadas pudiera funcionar pero en esta implementación se empleó el dataset AMI Corpus [37], el cual es brindado por la Universidad de Edinburgh y tiene audios grabados de reuniones en condiciones de oficina. Una clase que implementa código Python fue creada con este objetivo y mediante esta se aplica VAD a los archivos WAV con voces grabadas que se le administren para descomponerlos en segmentos de 0.5 segundos conteniendo solo voz, la longitud de los segmentos que devuelve esta clase es ajustable. No es necesario introducirle segmentos de voz completos o coherentes a esta clase, ya que cualquier audio con contenido de frecuencia similar es aceptado.

Para encontrar la RIR de cada micrófono el método de fuentes imágenes (Image Source Method (ISM)) es empleado. El ISM modela las ondas de presión emitidas por las fuentes de sonido en una habitación rectangular y crea una imagen de cada onda con energía atenuada después de cada reflexión en alguna pared. Una función de transferencia de la sumatoria de todas las ondas de presión en cualquier coordenada espacial puede ser hallada y su transformada inversa de Fourier es la RIR [38]. El método ISM puede estimar con bastante exactitud las RIR sobre la banda de frecuencia de audio para habitaciones rectangulares donde las fuentes y el arreglo de micrófono no estén muy cercanos a las paredes, lo cual es una condición muy aceptable para la implementación de este trabajo. Para simular la RIR que se muestra en la figura 4.10 se empleó la librería de Python pyroomacoustics [39], esta permite cambiar de muchas formas la habitación, introducir un número arbitrario de fuentes de voz y micrófonos, así como de localizaciones, emplear métodos complementarios a ISM, entre otras funcionalidades. Pyroomacoustics convolucionan las muestras de voz de las fuentes para determinar las RIR mediante una función que permite además agregar ruido aditivo blanco Gaussiano (AWGN) para simular las señales a las salidas de los micrófonos con la SNR seleccionada. De esta forma se creó el dataset requerido para crear la cantidad de datos de entrenamiento necesaria, variando las condiciones de ruido, reverberación, distancia y SNR para dotar a la red de una capacidad mejor para generalizar. En la figura 4.11 se muestra un ejemplo de una habitación y un escenario con reverberación, SNR, una fuente de voz y un arreglo de micrófonos lineal uniforme simulados con la clase creada en esta tesis mediante el empleo de la librería pyroomacoustics. En la figura 4.11b se muestran como puntos el número de las reflexiones consideradas por el modelo ISM para la simulación del escenario. Las posiciones de las fuentes imágenes representan las direcciones y la distancia es una medida de la atenuación con las que inciden en el arreglo estas reflexiones. En la figura 4.11c se muestra la señal de voz a la salida del cuarto canal del arreglo de micrófonos lineal uniforme simulado.

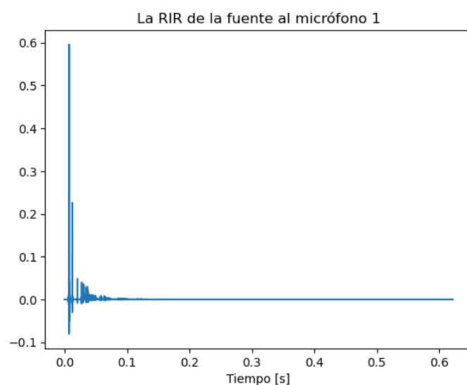
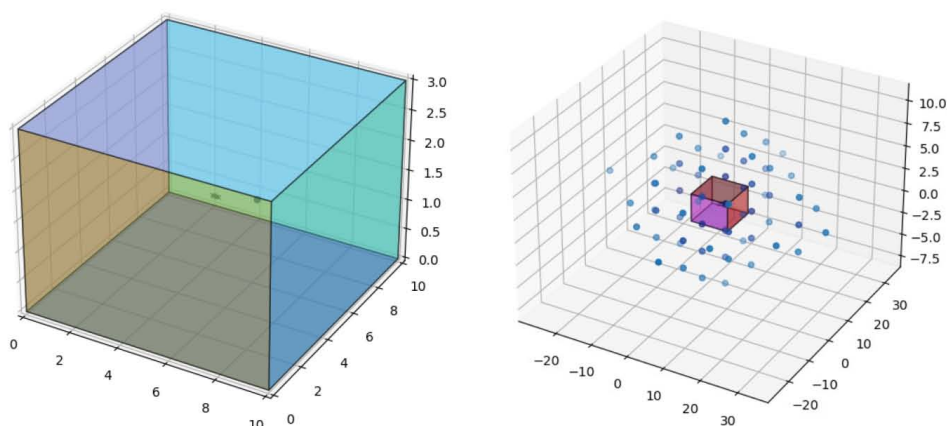
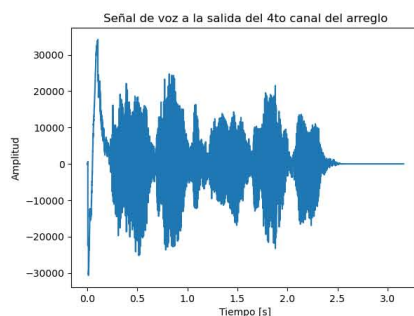


Figura 4.10: Respuesta al impulso de la habitación para una habitación simulada.



(a) Arreglo lineal en el centro y fuente de voz a 68° (b) Número de reflexiones consideradas por ISM



(c) Salida de voz del cuarto micrófono del arreglo en un escenario simulado.

Figura 4.11: Ejemplo de escenario simulado para la creación del dataset de audio.

Resumen

En este capítulo se expuso el diseño e implementación del algoritmo Neuro-Detector DOA que combina características y el funcionamiento de dos métodos convencionales para la estimación DOA: Beamforming y Subespacios. El algoritmo Neuro-Detector DOA implementado realiza un barrido de direcciones de arriba potenciales y por cada una crea un matriz de restricción (del mismo modo que se procediera en el caso de un LCMV en dicha dirección), a partir de la cual crea el espacio de representación de la señal en dicha dirección. Luego, mediante la matriz de correlaciones a la salida del Bf embebido en el sistema, utiliza una red neuronal para estimar el ruido y las variaciones estadísticas para determinar los subespacios de señal y solo-ruido sobre los que se proyecta cada espacio de representación. El valor

máximo del contorno de potencias resultante es la salida del sistema y por tanto la estimación DOA de la señal de voz. La NN implementada es del tipo MLP y fue probada con dos diseños diferentes, ambos satisfactorios pero con ventajas el uno sobre el otro. Uno de los diseños involucra alimentar la NN con los 150 eigenvectores de mayor información de la matriz \mathbf{R}_{xx} y devolver su rango efectivo a la salida, empleando dos HLs en la arquitectura. El otro diseño alimenta a la red los eigenvalores de la matriz \mathbf{R}_{xx} y recibe como salida el rango efectivo de la matriz, empleando para esto una sola HL. En el primer caso la NN es capaz de obtener más información de la estadística y el ruido de las señales, a expensas de un mayor número de capas y coeficientes en la red. El segundo caso solo emplea una capa oculta y una cantidad significativamente menor de coeficientes, sin embargo los resultados son menos satisfactorios que el primero. Para dar cumplimiento a los objetivos propuestos se empleó el diseño con menor complejidad y requerimiento computacional debido a que, de igual forma, funciona correctamente dentro del algoritmo.

Capítulo 5

Pruebas y Resultados

En este capítulo se presentan los resultados obtenidos del sistema Neuro-Detector para estimar los DOAs de señales de voz capturadas por dos geometrías de arreglos de micrófonos diferentes: una lineal uniforme (ULA) y otra circular uniforme (UCA). Se describen las condiciones bajo las cuales fueron simulados los datasets de validación para cada geometría y las características de estos. Se presenta una comparativa del desempeño del Neuro-Detector para las dos configuraciones de arreglos de micrófonos bajo el mismo escenario, dígase: iguales dimensiones para la habitación simulada, iguales intervalos de reverberación, SNR y distancias para la generación de los datos. Se abordan la resolución de estimación lograda, la robustés ante el aumento de las reverberaciones, SNR y las distancias entre el arreglo de micrófonos y las fuentes de voz; los tiempos de respuesta del sistema y la capacidad de estimación DOA para varias fuentes de voz incidentes.

La frecuencia de muestreo en el total de las pruebas realizadas para este capítulo fue $f_s = 16000 \text{ Hz}$. Convenientemente, para presentar los resultados del Neuro-Detector DOA se llevaron a cabo dos tipos de pruebas:

1. Pruebas de medio segundo de voz

- Tiene como propósito evaluar el rendimiento ante segmentos de medio segundo de voz incidentes, con DOAs, reverberación, SNR y distancias generados aleatoriamente (no contiguos) y así medir la robustés del sistema para el seguimiento o rastreo de fuentes en entornos altamente variables.
- Medir la aptitud del sistema para implementaciones en tiempo real.

2. Pruebas de segmentos completos de voz

- Tiene como propósito evaluar el rendimiento ante segmentos de duración mayor a un segundo y que representen palabras o frases completas de voz en situaciones de fuentes estáticas en escenarios variables.
- El sistema estará haciendo estimaciones cada medio segundo de voz y es de interés compararlas en el caso que pertenezcan a una misma frase o declaración de voz y la fuente sea estática.

5.1. Neuro-Detector DOA para ULA

La validación del Neuro-Detector para la estimación DOA de señales de voz capturadas por un arreglo de micrófonos lineal uniforme se llevó a cabo generando varios datasets (subsección 4.5.2) de ejemplos etiquetados, que representan las secuencias de muestras obtenidas de cada canal del arreglo y las etiquetas indican el valor DOA con el cual inciden las señales simuladas sobre el arreglo y que debe estimar el sistema. De esta forma, se midió el desempeño del sistema obteniendo el error absoluto, $E_{||}$, entre la respuesta del Neuro-Detector y la etiqueta en cada ejemplo que se alimenta. Si este error es menor o igual a cinco grados entonces se consideró como una estimación exitosa, de otro modo como errónea. Así las respuestas del sistema se dividieron en exitosas, con el valor de uno y en erróneas, con el valor de cero. Al conjunto de todas las respuestas dadas por el sistema ante el dataset de validación en cuestión se le halló la media y esta fue el índice de desempeño empleado para la validación. Además se presentan otras medidas de rendimiento como son la media, varianza, mediana, moda y los valores mínimos y máximo del error de estimación.

Es importante hacer notar que el índice de reverberación empleado se denomina $rt60$ y representa el tiempo que toman las reverberaciones en atenuarse hasta 60 dB . Los valores de SNR e índice de reverberación son calculados como parte de las funciones pertenecientes a la librería `pyroomacoustics` [39] empleada para las simulaciones. El ULA considerado para las generaciones de datasets está compuesto por $M = 8$ micrófonos omnidireccionales separados a una distancia $d = 0.04287 \leq \lambda_{min}/2$. Además, la distancia D de la fuente de voz al arreglo fue tal que el modelo de onda incidente sobre el arreglo se considera de campo lejano. Para el caso del ULA las posibles direcciones de arribo van desde $[-90^\circ, 90^\circ]$, lo que representa un número infinito de posibles DOAs, sin embargo, con fines prácticos solo se consideraron arribos DOAs que representan números enteros y por tanto, para la implementación y pruebas se hace un barrido de 181 direcciones potenciales. De este modo, el sistema en cada prueba, estará estimando la más probable de entre las 181 direcciones posibles.

5.1.1. Pruebas de medio segundo de voz

Se generó un primer dataset con 2707 ejemplos etiquetados para llevar a cabo esta prueba. Las características de este dataset se muestran en la tabla 5.1. Para la generación de este dataset se utilizó un archivo de audio tomado de AMI corpus [37] con una duración de 32 minutos y 25 segundos y además dos archivos de audio con mi voz, de 12 segundos cada uno. Por supuesto, ninguna de estas señales fue utilizada en los ejemplos etiquetados del dataset de entrenamiento del sistema. Los archivos fueron concatenados y preprocesados mediante WebRTCvad (subsección 4.5.1) y se dispusieron como segmentos de voz de medio segundo de duración, estos fueron organizados de forma aleatoria para simular y generar las muestras obtenidas por el ULA con las características mostradas en la tabla 5.1.

Parámetro	Valor
Dimensiones de la habitación	(10 <i>largo</i> , 10 <i>ancho</i> , 3 <i>altura</i>) [m]
Intervalo de direcciones de arribo	$[-90^\circ, 90^\circ]$
Intervalo del índice de reverberación $rt60$	[0.18, 0.35] [s]
SNR	[0.0, 20.0] [dB]
Intervalo de distancia de la fuente de voz al arreglo (D)	[1.0, 2.0] [m]
Posición del arreglo	(5 <i>largo</i> , 5 <i>ancho</i> , 1.5 <i>altura</i>) [m]
Posición de la fuente de voz	(5 <i>largo</i> , 5 <i>ancho</i> , 1.5 <i>altura</i>) + $D \cdot \overrightarrow{DOA}$ [m]
Cantidad de ejemplos etiquetados	2707

Tabla 5.1: Características del dataset generado para las pruebas de medio segundo ULA

En la figura 5.1 se muestra una gráfica escalada donde se despliega en dos dimensiones la posición (distancia y dirección), en azul, de cada fuente de voz con respecto al costado del arreglo de micrófonos, en rojo, en cada ejemplo etiquetado del dataset. Mientras que en la tabla 5.2 se muestran el desempeño y otros valores de rendimiento del sistema obtenidos a partir del error absoluto generado en cada respuesta. Como se aprecia el desempeño fue de **98.96%**, lo que significa que solo cerca del 2% de los 2707 ejemplos fue estimado con un error absoluto mayor a 5° y en estos casos, el mayor error cometido fue de **9.9447°** como muestra el error absoluto máximo. Además, en la figura 5.2 se muestra la etiqueta con el DOA correcto de cada medio segmento generado y la respuesta correspondiente del Neuro-Detector DOA para algunos ejemplos del dataset. En el Apéndice A se muestra una mayor parcialidad de los resultados.

Con los resultados obtenidos de este tipo de pruebas para varios datasets se puede considerar que el sistema es funcional y capaz de realizar estimaciones DOAs de señales de voz provenientes de fuentes en movimiento en un escenario real y altamente variable. Se verifica que no necesita grandes cantidades de realizaciones o vectores de muestras espacio-temporales de la señal para obtener la matriz de correlaciones \mathbf{R}_{xx} necesaria para las estimaciones, haciendo al sistema más versátil y rápido.

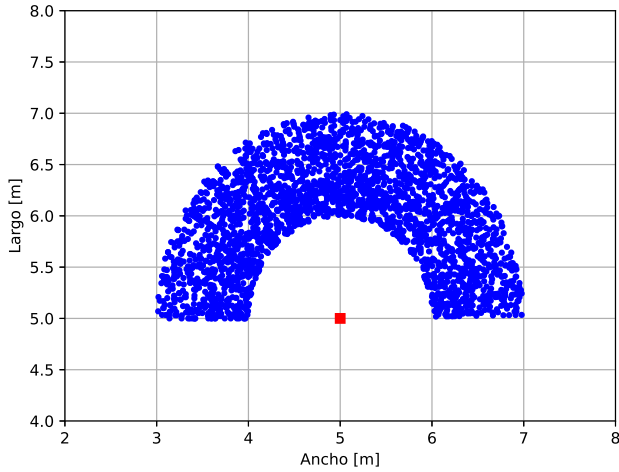


Figura 5.1: Posiciones en 2D del dataset generado para la validación ULA

Desempeño	98.96 %
Media del error absoluto	1.1798°
Varianza del error absoluto	1.0941°
Mediana del error absoluto	0.8342°
Moda del error absoluto	0.5082°
Máximo del error absoluto	9.9447°
Mínimo del error absoluto	0.0°

Tabla 5.2: Desempeño y valores de rendimiento del sistema. Dataset 2707 ejemplos. ULA

etiqueta DOA: -->	-29.000000	etiqueta DOA: -->	52.000000	etiqueta DOA: -->	74.000000
estimación DOA:-->	-28.342541	estimación DOA:-->	51.215470	estimación DOA:-->	73.093923
etiqueta DOA: -->	-10.000000	etiqueta DOA: -->	6.000000	etiqueta DOA: -->	-69.000000
estimación DOA:-->	-10.441989	estimación DOA:-->	4.475138	estimación DOA:-->	-69.116022
etiqueta DOA: -->	21.000000	etiqueta DOA: -->	-5.000000	etiqueta DOA: -->	-20.000000
estimación DOA:-->	19.392265	estimación DOA:-->	-5.469613	estimación DOA:-->	-19.392265
etiqueta DOA: -->	-50.000000	etiqueta DOA: -->	88.000000	etiqueta DOA: -->	-34.000000
estimación DOA:-->	-50.220994	estimación DOA:-->	85.027624	estimación DOA:-->	-34.309392
etiqueta DOA: -->	-54.000000	etiqueta DOA: -->	5.000000	etiqueta DOA: -->	-27.000000
estimación DOA:-->	-54.198895	estimación DOA:-->	3.480663	estimación DOA:-->	-26.353591
etiqueta DOA: -->	50.000000	etiqueta DOA: -->	-34.000000	etiqueta DOA: -->	-53.000000
estimación DOA:-->	49.226519	estimación DOA:-->	-33.314917	estimación DOA:-->	-53.204420
etiqueta DOA: -->	-28.000000	etiqueta DOA: -->	-87.000000	etiqueta DOA: -->	17.000000
estimación DOA:-->	-27.348066	estimación DOA:-->	-81.049724	estimación DOA:-->	15.414365
etiqueta DOA: -->	75.000000	etiqueta DOA: -->	-90.000000	etiqueta DOA: -->	26.000000
estimación DOA:-->	73.093923	estimación DOA:-->	-84.033149	estimación DOA:-->	24.364641
etiqueta DOA: -->	40.000000	etiqueta DOA: -->	-25.000000	etiqueta DOA: -->	-59.000000
estimación DOA:-->	39.281768	estimación DOA:-->	-24.364641	estimación DOA:-->	-59.171271
etiqueta DOA: -->	72.000000	etiqueta DOA: -->	45.000000	etiqueta DOA: -->	31.000000
estimación DOA:-->	70.110497	estimación DOA:-->	44.254144	estimación DOA:-->	28.342541
etiqueta DOA: -->	-32.000000	etiqueta DOA: -->	-20.000000	etiqueta DOA: -->	69.000000
estimación DOA:-->	-31.325967	estimación DOA:-->	-19.392265	estimación DOA:-->	67.127072

Figura 5.2: Algunos resultados producidos por el sistema para el dataset de 2707 ejemplos etiquetados ULA

Para mostrar más claramente la salida y el desempeño del sistema en este capítulo se generó un segundo dataset empleando los mismos archivos de audio del dataset anterior y las mismas dimensiones de la habitación (tabla 5.1), generando una vez más, el conjunto de condiciones DOA, reverberación, SNR y distancias de cada medio segundo de voz de forma aleatoria. El objetivo principal en esta ocasión es acotar el dataset a solo 200 ejemplos etiquetados para visualizar mejor los resultados. Con relación a este dataset se muestra en la figura 5.3 las estimaciones obtenidas, en azul, superpuestas a las posiciones, en magenta, simuladas en el mismo. En la figura 5.3a se grafica la posición (distancia y dirección), en magenta, de cada fuente de voz con respecto al costado del arreglo de micrófonos, en rojo, en cada ejemplo etiquetado del dataset. En la figura 5.3b se muestra el error absoluto en radianes (en verde) asociado a cada resultado obtenido por el Neuro-Detector y en la figura 5.3c se muestran los errores absolutos en grados vs el número de estimaciones realizadas. El desempeño del sistema fue del **100 %** como se muestra en la tabla 5.3 y la media del error absoluto de solo **1.1450°**. El valor máximo del error, que justifica

el desempeño, fue de 4.9668° , y como se muestra en la figura 5.3c se concentran principalmente en las direcciones límites a los bordes del ULA.

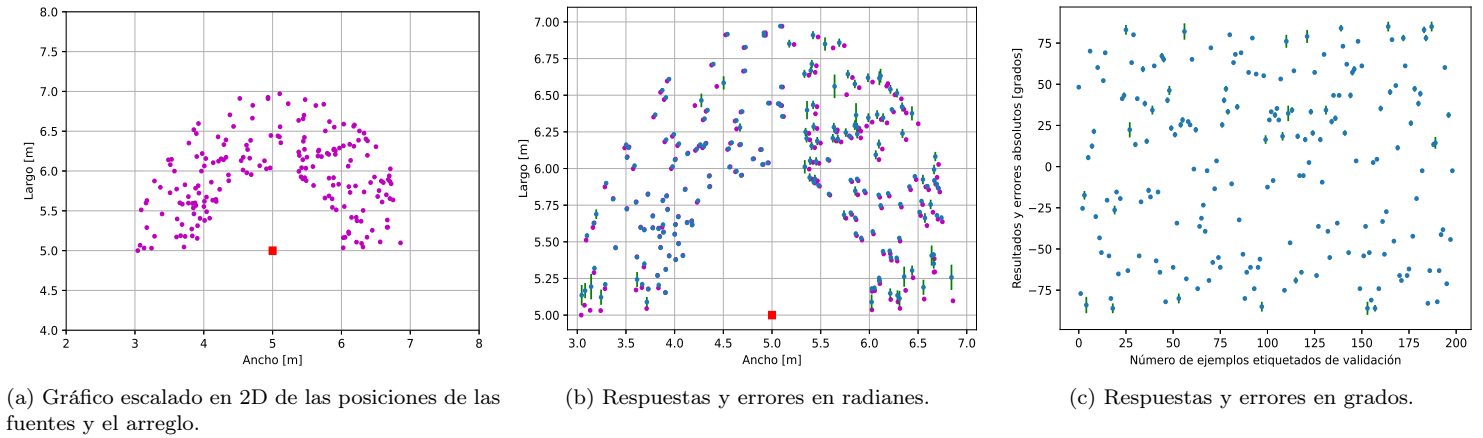


Figura 5.3: Dataset alimentado y el desempeño del sistema ULA.

Desempeño	1.0
Media del error absoluto	1.1450°
Varianza del error absoluto	0.9571°
Mediana del error absoluto	0.8342°
Moda del error absoluto	0.8397°
Máximo del error absoluto	4.9668°
Mínimo del error absoluto	0.0441°

Tabla 5.3: Desempeño y valores de rendimiento del sistema. Dataset 200 ejemplos. ULA

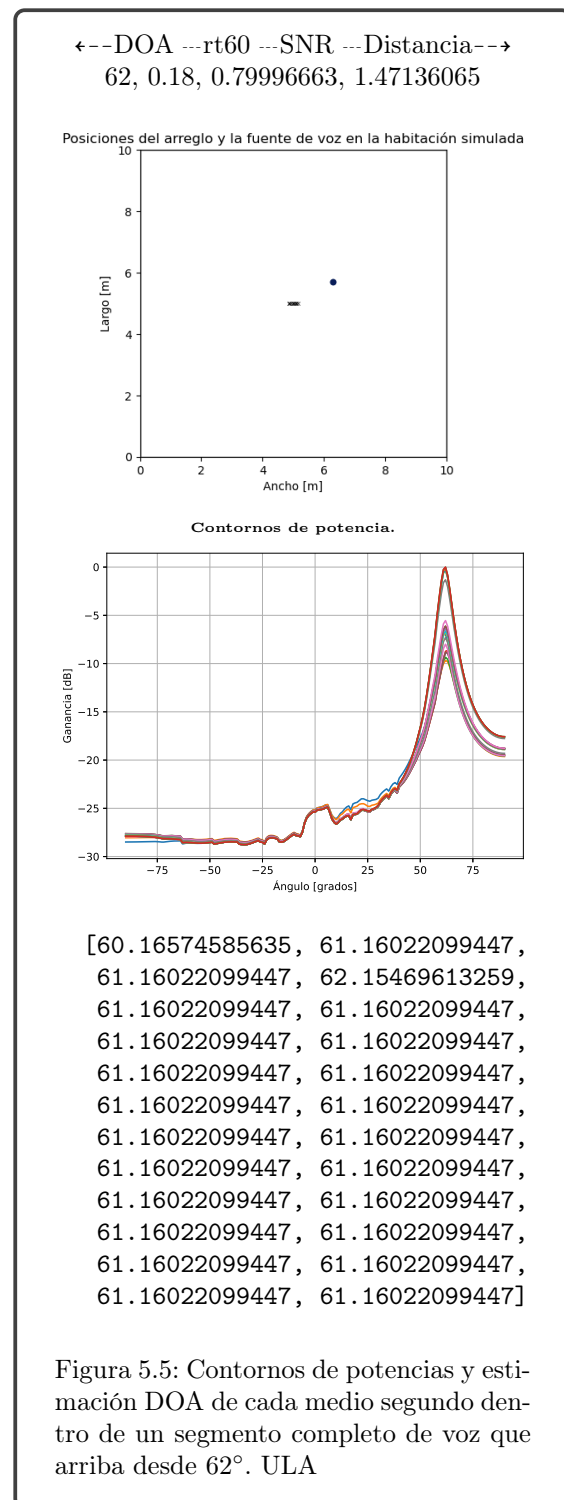
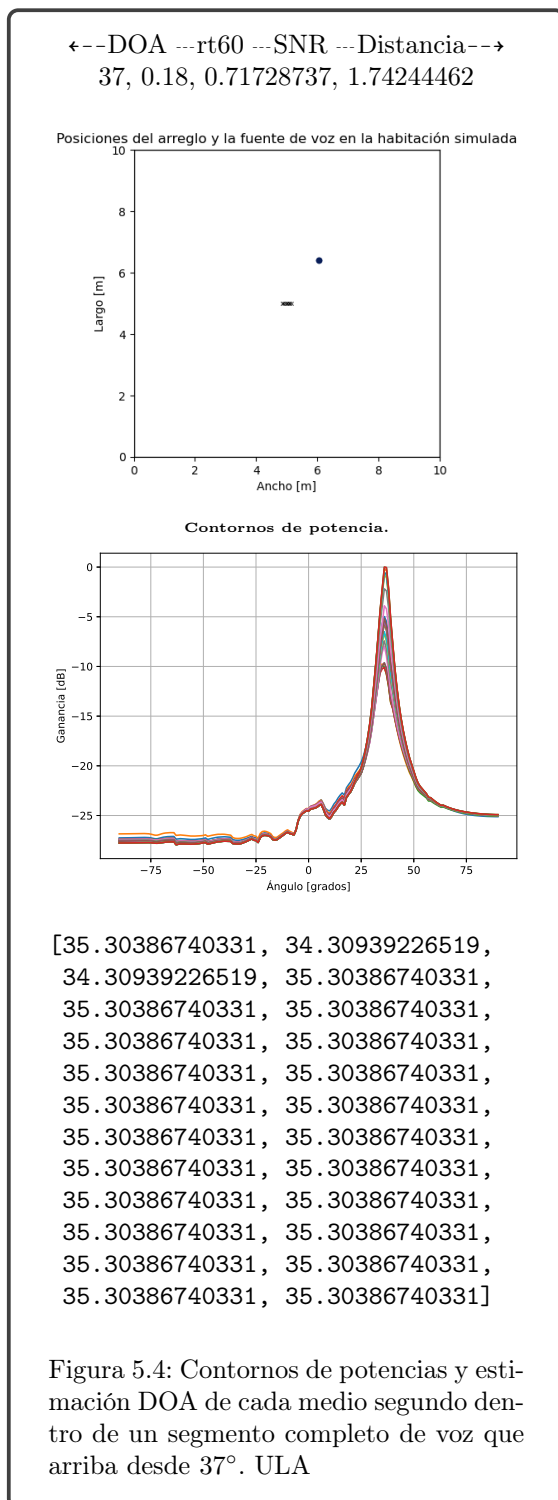
5.1.2. Pruebas de segmentos completos de voz

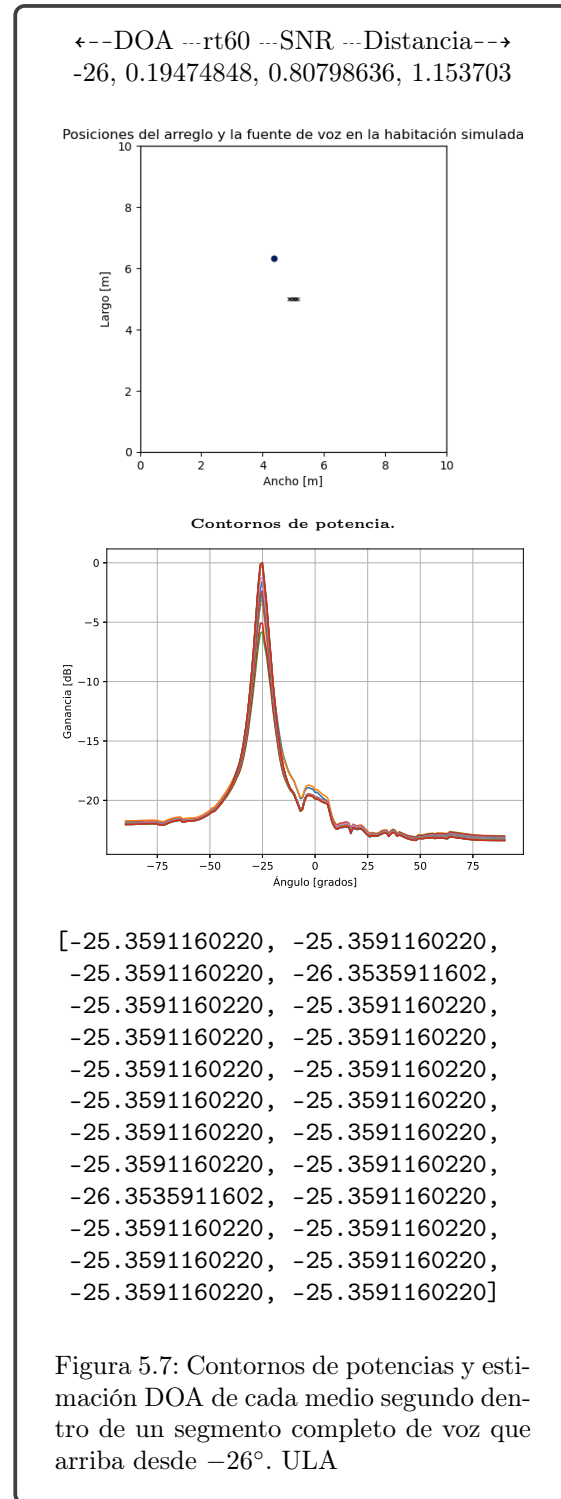
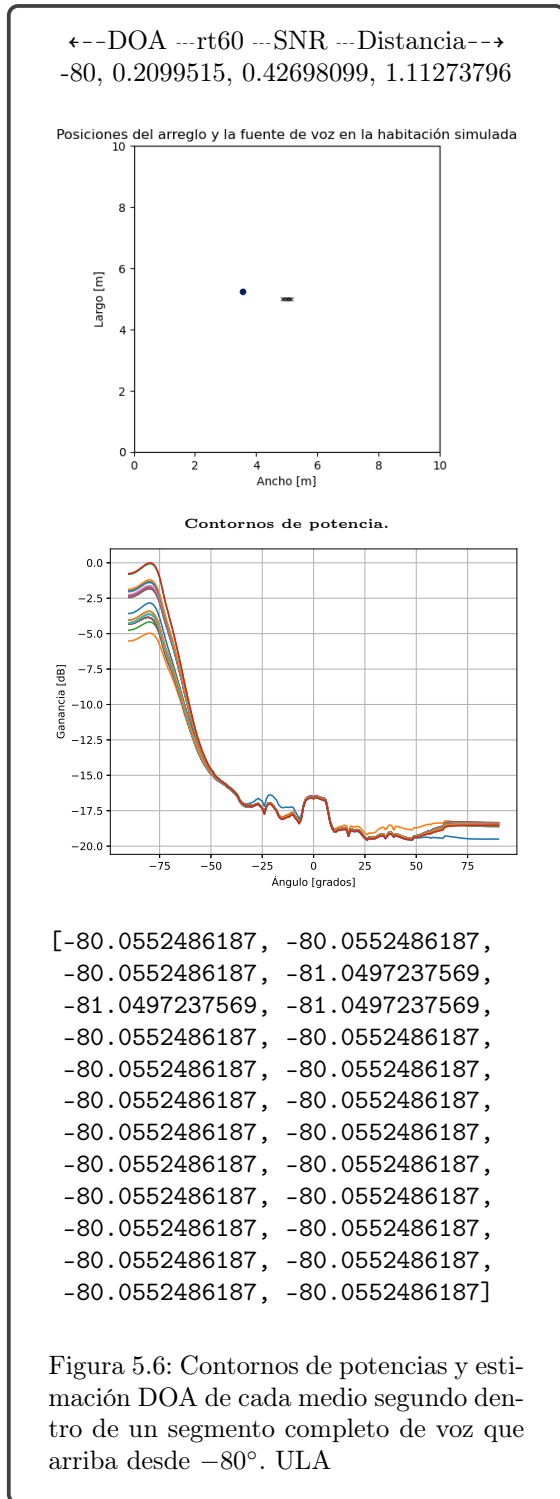
Para estas pruebas no se generaron datasets sino que se concatenaron varios archivos de voz con locutores diferentes y en diferentes idiomas. Estos fueron obtenidos de AMI corpus y de grabaciones de audio hechas para este propósito, por supuesto ninguna de estas señales fue utilizada en los ejemplos etiquetados del dataset de entrenamiento. Una vez concatenadas las señales, se emplearon los módulos desarrollados en esta tesis y, mediante la librería pyroomacoustics, se generaron los ocho canales de voz a la salida del ULA, para seis combinaciones aleatorias de DOA, reverberación, SNR y distancia entre las fuentes y el ULA en una habitación simulada de diez metros de largo y ancho y tres metros de alto. En la tabla 5.4 se resumen estas características.

El objetivo de estas pruebas fue corroborar el correcto funcionamiento del Neuro-Detector para estimar DOA de señales de voz correspondientes a palabras completas o frases coherentes provenientes de fuentes estáticas. Como ya se explicó, cada medio segundo se hace una estimación y para estas pruebas, cada estimación de medio segundo debe corresponder a una misma dirección de arriba o al menos tener una variación muy pequeña. Desde la figura 5.4 hasta la figura 5.9 se pueden apreciar, en cada caso, las posiciones de las fuentes relativas al ULA; los contornos de potencias obtenidos de las proyecciones sobre el subespacio de señal, por cada dirección considerada de cada medio segundo en que son divididos los segmentos completos de voz, y por último se despliegan los valores máximos de cada contorno que corresponden a la estimación de medio segundo, las cuales no deben ser muy diferentes ya que corresponden a una misma señal de voz desde una misma fuente estática.

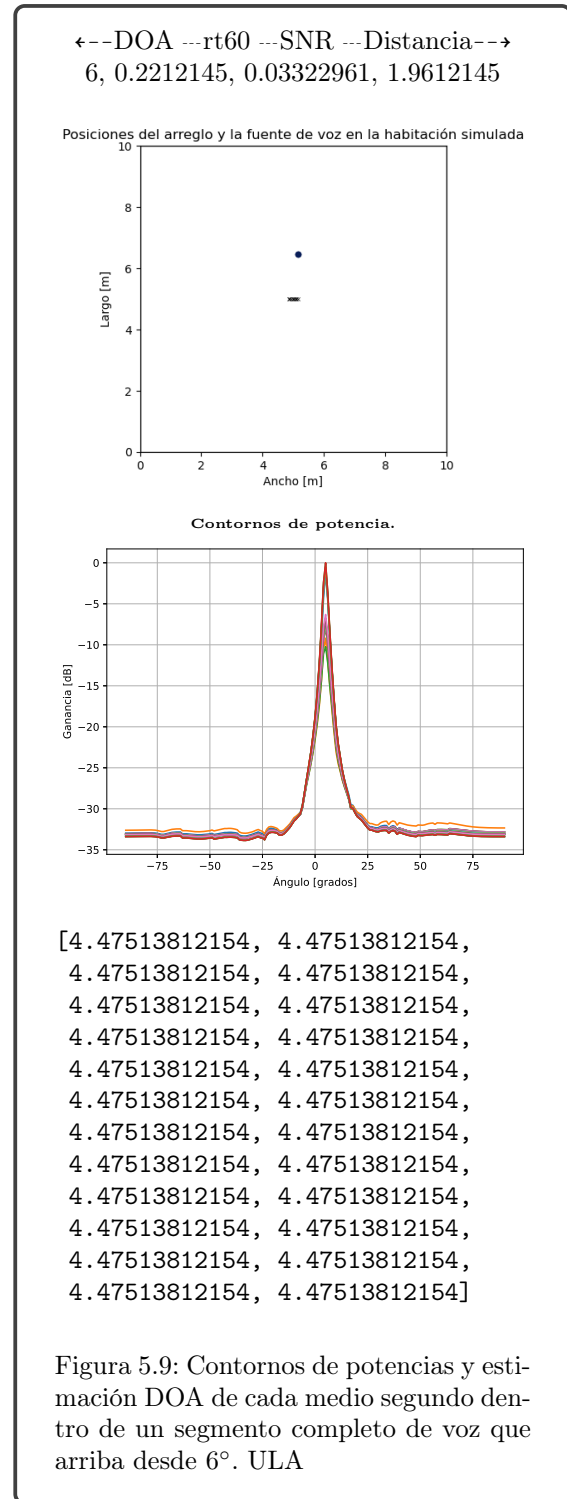
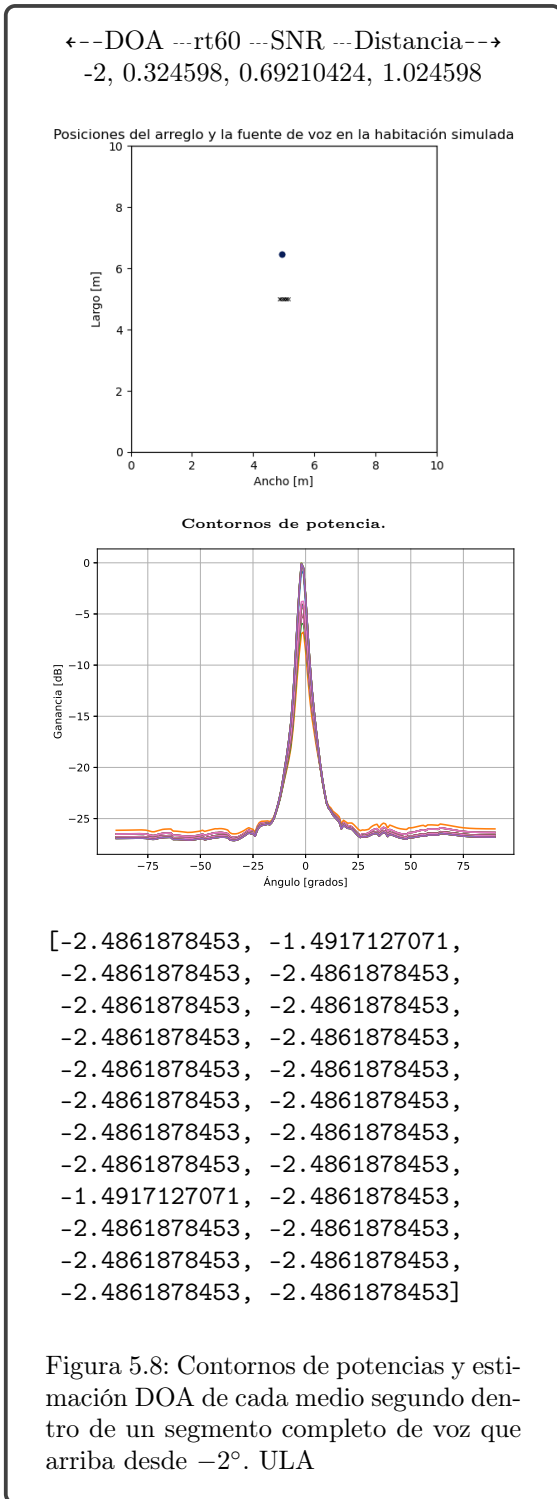
Parámetro	Valor
DOAs generados	37, 62, -80, -26, -2, 6
rt60 generados	0.18, 0.18, 0.2099515, 0.19474848, 0.324598, 0.2212145
SNR generados	0.71728737, 0.79996663, 0.42698099, 0.80798636, 0.80798636, 0.03322961
Distancias generadas	1.74244462, 1.47136065, 1.11273796, 1.153703, 1.024598, 1.9612145

Tabla 5.4: Intervalos considerados para las pruebas de segmentos completos ULA





Como se observa en las respuestas del sistema recogidas desde la figura 5.4 hasta la figura 5.9 , el algoritmo logra en los seis casos un desempeño del **100 %**, de modo que todas las estimaciones tienen un error absoluto menor a 5° . Además, como es de esperar, las estimaciones de medio segundo correspondientes a cada prueba no presentan grandes variaciones, lo que significa que se estimó correctamente el segmento completo de voz en cada caso.



5.1.3. Efectos del aumento de las reverberaciones y la distancia

El desempeño y la resolución en la estimación del Neuro-Detector estarán sujetos a las cualidades del ULA con el que se adquieren las muestras de voz, así como de la configuración del Bf. En las pruebas de subsecciones anteriores se consideraron valores de $rt60$ que no eran altos y distancias que no excedían los $2m$. En esta sección se muestran los efectos sobre las estimaciones del sistema cuando se somete a escenarios con altos valores de $rt60$ y distancias mayores entre la fuente y el ULA. En la figura 5.10 se muestran diferentes escenarios simulados con valores de reverberación altos, $rt60 \in [0.4, 1.0]$ [s], y la mínima distancia de campo lejano ($D = 1m$) para pruebas de segmentos completos de voz; y se muestran cómo son afectadas negativamente la resolución, los contornos de potencia y por tanto la estimación DOA.

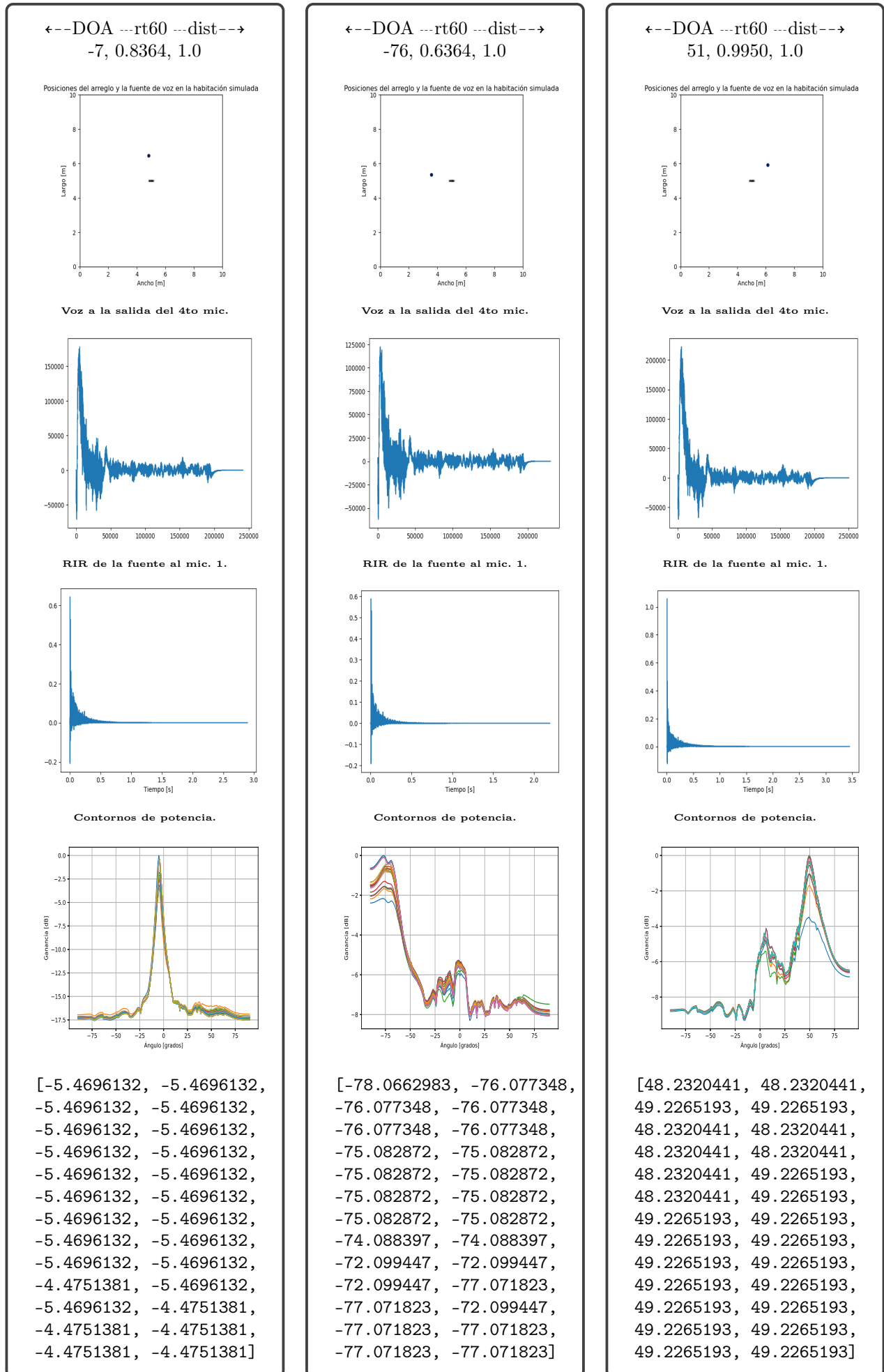


Figura 5.10: Efecto del aumento en rt60 para una distancia de un metro en la estimación DOA. ULA

Para las pruebas de segmentos completos de voz de la figura 5.10 se utilizaron los mismos archivos de audio que en la subsección anterior y las dimensiones de la habitación simulada fueron las mismas (tabla 5.1). En esta figura se muestran los efectos del aumento de las reverberaciones en las muestras de voz y en la RIR a la salida de un canal del ULA. A modo de comparación se puede referir a la figura 5.11 donde se muestran la salida de un canal del ULA y la RIR con el menor $rt60$ permitido para estas dimensiones de la habitación. Como se muestra, cuando el $rt60$ aumenta considerablemente, a pesar de que la distancia sea la mínima para campo lejano, se afecta apreciablemente la precisión y la varianza en las estimaciones de medio segundo para los segmentos completos. La resolución de los contornos de potencia se afecta de forma notable y la diferencia en los decibeles de ganancia entre los máximos y mínimos se reduce drásticamente. Sin embargo, las estimaciones siguen estando dentro del intervalo de 5° y el desempeño para el total de estimaciones de medio segundo para estas tres pruebas fue del **100%**, con un valor máximo del error absoluto de **3.900°** y una media de **1.7466°** .

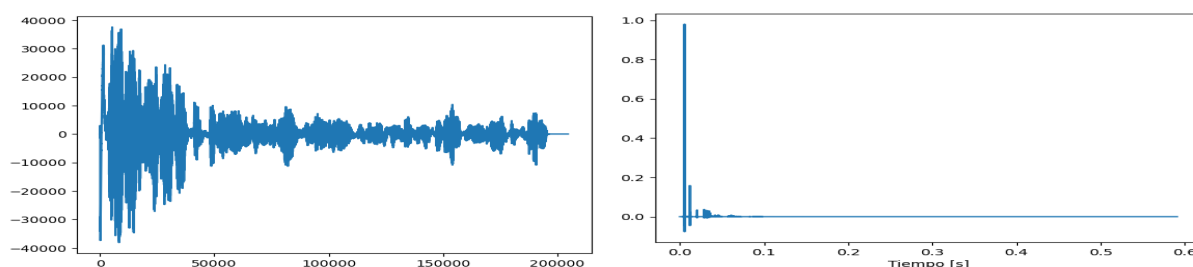


Figura 5.11: Señal de voz y RIR a la salida de un canal del ULA con el menor efecto de reverberaciones y distancia.

En la figura 5.12 se muestran otras tres pruebas de segmentos completos, empleando las mismas dimensiones en la habitación simulada y los mismos archivos de audio que en las pruebas anteriores. En este caso no solo varía el índice $rt60$ sino también la distancia entre las fuentes y el ULA. En esta ocasión se lleva al límite las condiciones en esta habitación, bajo las que puede operar el Neuro-Detector sujeto a un ULA. Como se observa en este caso las distancias son cercanas a $4m$ y los valores de $rt60$ son muy altos. Imagine una habitación donde al producirse un sonido, los ecos y las reverberaciones en general demoren cerca de un segundo para desvanecerse y donde las distancias sean considerables hasta para una persona. En este contexto extremo hemos probados el sistema. Como muestran los resultados recogidos en la figura 5.12, ninguna de las estimaciones de medio segundo tiene una precisión de al menos 5° y por tanto el desempeño para este caso fue del **0%**, la media del error absoluto para estas pruebas fue de **20.7721°** y el valor máximo fue de **51.6850°** . Sin embargo, el sistema aún es capaz de estimar correctamente el sentido desde donde incide la señal, es decir, si proviene de la izquierda (valores negativos de DOA), o la derecha (valores positivos de DOA). En los casos en que la fuente esté localizada en direcciones cercanas a la perpendicular al ULA (0°), como en la figura 5.12c, a pesar de las condiciones extremas logra estimaciones de cerca de 10° de precisión, lo cual, bajo estas condiciones, puede ser aceptable.

5.2. Neuro-Detector DOA para UCA

Como se mencionó en la sección 4.1 el algoritmo desarrollado no es restrictivo a una configuración específica del arreglo de micrófonos, como es ULA. En esta sección se muestran resultados satisfactorios para el caso del Neuro-Detector sujeto a un arreglo circular uniforme.

Para validar el desempeño del Neuro-Detector en la estimación DOA de señales de voz capturadas por un UCA se procedió de igual forma que en la sección anterior, generándose varios datasets de ejemplos etiquetados, que representan las secuencias de muestras obtenidas de cada canal del arreglo y las etiquetas indican el valor DOA con el cual inciden las señales. De esta forma, se re-entrena la MLP y se obtienen los coeficientes correspondientes al caso del UCA.

Para el caso del Neuro-Detector asociado a un UCA, no es correcto medir el desempeño a partir del error absoluto, es decir, como el módulo de la diferencia de la etiqueta DOA y la respuesta del sistema, pues para casos en que la fuente se encuentre en una dirección cercana a 0° , digamos, 359° , la respuesta del sistema igual 1° acarrea un error absoluto de 358° cuando en realidad es de 2° . En la tabla 5.5 se muestran ejemplos y se deja claro que para este caso el desempeño debe ser calculado como la distancia o

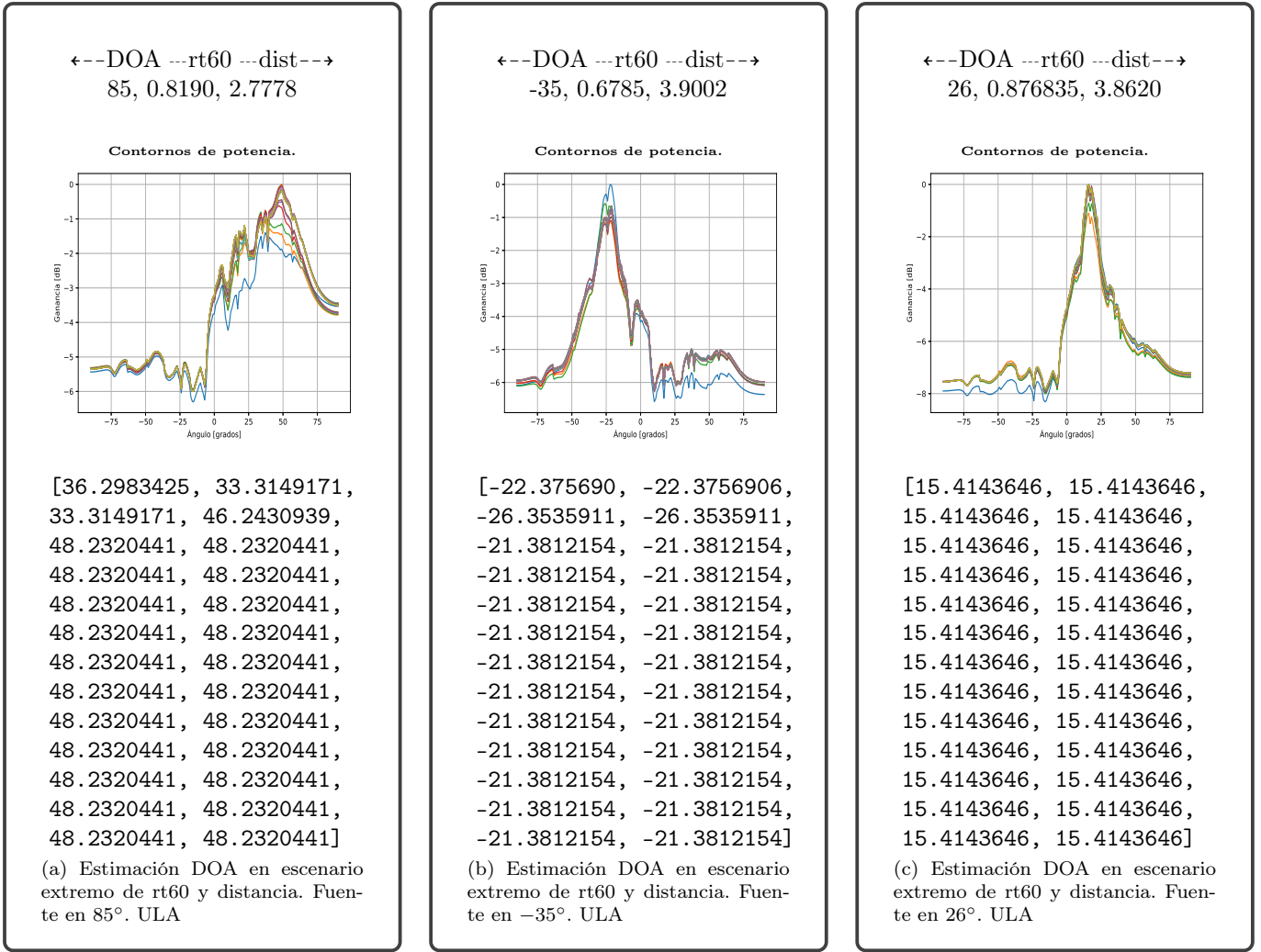


Figura 5.12: Efecto del aumento en rt60 para una diatancia de un metro en la estimación DOA. ULA

norma euclidiana. De esta forma, si la distancia fue menor o igual a cinco grados entonces se consideró como una estimación exitosa, de otro modo como errónea. Así, las respuestas del sistema se dividieron en exitosas con el valor de uno, y en erróneas con el valor de cero. Al conjunto de todas las respuestas dadas por el sistema se le halló la media y esta fue el índice de desempeño empleado para la validación. Además se presentan otras medidas de rendimiento como son la media, varianza, mediana, moda y los valores mínimos y máximo del error de estimación.

Etiqueta DOA	Estimación DOA	Error absoluto $E_{ }$	Distancia euclidiana $E_{ }$
		$ etqDOA - estDOA $	$\ e^{j\alpha} - e^{j\beta}\ $
359°	0°	359°	0.9999°
2°	359°	357°	2.9999°
142°	140.6879°	1.3121	1.3120

Tabla 5.5: Forma de medir el desempeño del Nuero-Detector sujeto a UCA

El UCA considerado para las generaciones de datasets está compuesto por $M = 8$ micrófonos omnidireccionales, colocados cada 45° y a una distancia $d = 0.04287 \leq \lambda_{min}/2$. Además, la distancia D de la

fuente de voz al arreglo fue tal que el modelo de onda incidente sobre el arreglo se considera de campo lejano. Para el caso del UCA las posibles direcciones de arribo van desde $[0^\circ, 359^\circ]$, lo que representa un número infinito de posibles DOAs. Al igual que en el caso del ULA solo se consideraron, en el barrido, 181 direcciones que representan DOAs enteros, sin embargo, para este caso del UCA esto representa una resolución de barrido de 2° . De este modo el sistema, en cada prueba, estará estimando la más probable de entre las 181 direcciones consideradas.

Para las pruebas de esta sección se utilizaron las mismas dimensiones de la habitación simulada en cada dataset y también los mismos archivos de audio que en pruebas anteriores. La metodología para la generación aleatoria de las condiciones DOA, rt60, SNR y distancias en los datasets para las pruebas de medio segundo de voz y las pruebas de segmentos completos de voz fue la misma que se describe en la sección anterior.

5.2.1. Pruebas de medio segundo de voz

Las características del dataset generado para estas pruebas se muestran en la tabla 5.6. Se aprecia que en este caso el intervalo de los valores de rt60 y las distancias entre las fuentes y el UCA comprenden valores mucho mayores que en los casos del dataset ULA para esta misma prueba. Sin embargo, como se muestra en la tabla 5.7 el desempeño logrado por el sistema sujeto a acrs bajo estas condiciones es satisfactorio y superior al logrado en el caso del ULA, aún cuando las condiciones en el escenario de este último eran más favorables. En la figura 5.14 se muestran las posiciones (distancia y dirección), en azul, de cada fuente de voz con respecto al centro del UCA, en rojo, en cada ejemplo etiquetado del dataset. Además, en la figura 5.14 se muestra la etiqueta con el DOA correcto de cada medio segmento generado y la respuesta correspondiente del Neuro-Detector DOA para algunos ejemplos del dataset. En el Apéndice A se muestra una mayor parcialidad de estos resultados.

Parámetro	Valor
Dimensiones de la habitación	(10 largo, 10 ancho, 3 altura) [m]
Intervalo de direcciones de arribo	$[0^\circ, 359^\circ]$
Intervalo del índice de reverberación rt60	$[0.18, 1.0]$ [s]
SNR	$[0.0, 20.0]$ [dB]
Intervalo de distancia de la fuente de voz al arreglo (D)	$[1.5, 3.0]$ [m]
Posición del arreglo	(5 largo, 5 ancho, 1.5 altura) [m]
Posición de la fuente de voz	$(5 \text{ largo}, 5 \text{ ancho}, 1.5 \text{ altura}) + D \cdot \overrightarrow{DOA}$ [m]
Cantidad de ejemplos etiquetados	2707

Tabla 5.6: Características del dataset generado para las pruebas de medio segundo UCA

Analizando el resumen de los resultados para esta prueba en la tabla 5.7 se observa que el **99%** de los ejemplos de medio segundo del dataset fue estimado con una precisión menor a 5° , aún cuando los valores de rt60 y las distancias eran elevados. Estas condiciones desfavorables son las causantes de que se tenga un valor máximo del error de 30.58° , correspondiente a los valores encerrados entre llaves de la segunda columna de la figura 5.14. Sin embargo, la media, que es una medición que se ve afectada por los valores extremos, es de solo **1.0819°** , la varianza, la medida de las variaciones del error con respecto a su media, es de solo **1.5323°** , el valor que más se repite del error es de **0.1988°** y el valor justo en el centro del total de errores cometidos por el sistema es de solo **0.7458°** . Por tanto, se desprende que es muy probable que este valor máximo del error ocurrió solo una vez y es la razón por la que no se logró el 100% del desempeño.

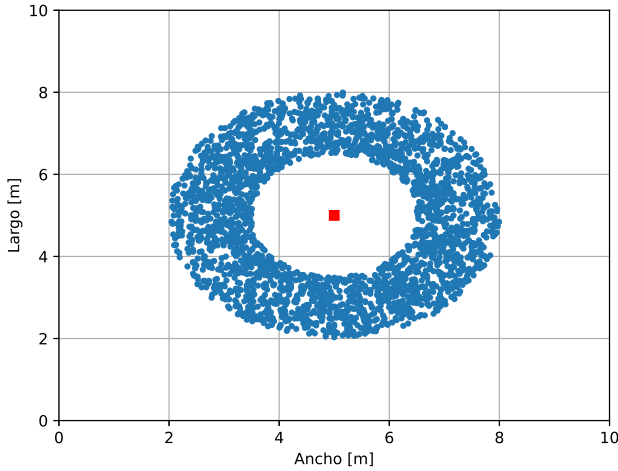


Figura (5.14) Posiciones en 2D del dataset generado para la validación UCA

Desempeño	0.9922
Media norma euclidiana	1.0819°
Varianza norma euclidiana	1.5323°
Mediana norma euclidiana	0.7458°
Moda norma euclidiana	0.1988°
Máximo norma euclidiana	30.5856°
Mínimo norma euclidiana	0.0°

Tabla (5.7) Desempeño y valores de rendimiento del sistema. Dataset 2707 ejemplos. UCA

etiqueta DOA: -->	324.000000	etiqueta DOA: -->	31.000000	etiqueta DOA: -->	344.000000
estimación DOA:-->	324.198895	estimación DOA:-->	27.845304	estimación DOA:-->	344.088398
etiqueta DOA: -->	126.000000	etiqueta DOA: -->	{136.000000}	etiqueta DOA: -->	240.000000
estimación DOA:-->	125.303867	estimación DOA:-->	{105.414365}	estimación DOA:-->	238.674033
etiqueta DOA: -->	228.000000	etiqueta DOA: -->	187.000000	etiqueta DOA: -->	185.000000
estimación DOA:-->	228.729282	estimación DOA:-->	186.961326	estimación DOA:-->	184.972376
etiqueta DOA: -->	2.000000	etiqueta DOA: -->	346.000000	etiqueta DOA: -->	85.000000
estimación DOA:-->	1.988950	estimación DOA:-->	346.077348	estimación DOA:-->	85.524862
etiqueta DOA: -->	206.000000	etiqueta DOA: -->	319.000000	etiqueta DOA: -->	335.000000
estimación DOA:-->	204.861878	estimación DOA:-->	316.243094	estimación DOA:-->	336.132597
etiqueta DOA: -->	227.000000	etiqueta DOA: -->	330.000000	etiqueta DOA: -->	169.000000
estimación DOA:-->	224.751381	estimación DOA:-->	332.154696	estimación DOA:-->	169.060773
etiqueta DOA: -->	6.000000	etiqueta DOA: -->	49.000000	etiqueta DOA: -->	176.000000
estimación DOA:-->	5.966851	estimación DOA:-->	47.734807	estimación DOA:-->	175.027624

Figura 5.14: Algunos resultados producidos por el sistema para el dataset de 2707 ejemplos etiquetados UCA

Para visualizar más claramente la salida y el desempeño del sistema para el UCA, se generó un segundo dataset empleando los mismos archivos de audio del dataset anterior, las mismas dimensiones de la habitación y, una vez más, el conjunto de condiciones DOA, reverberación, SNR y distancias de cada medio segundo de voz se generó de forma aleatoria. El objetivo fue acotar el dataset a solo 200 ejemplos etiquetados para visualizar mejor los resultados. Con relación a este dataset se muestra en la figura 5.15 las estimaciones obtenidas, en azul, superpuestas a las posiciones, en amarillo, simuladas en el mismo. En la figura 5.15a se grafica la posición (distancia y dirección), en amarillo, de cada fuente de voz con respecto al centro del UCA, en rojo, en cada ejemplo etiquetado del dataset. En la figura 5.15b se muestra el error absoluto en radianes (en verde) asociado a cada resultado obtenido por el Neuro-Detector y en la figura 5.15c se muestran los errores absolutos en grados vs el número de estimaciones realizadas. El desempeño del sistema fue del **100%** como se muestra en la tabla 5.8, logrando una media del error absoluto de solo **0.6627°** y un valor máximo de **2.7569°**, lo que justifica el desempeño logrado.

5.2.2. Pruebas de segmentos completos de voz

Para llevar a cabo estas pruebas se siguen los mismos pasos descritos en la subsección 5.1.2 y se generan las muestras de voz a la salida de los ocho canales del UCA, para seis combinaciones aleatorias de DOA, reverberación, SNR y distancia entre las fuentes y el arreglo; en una habitación simulada de diez metros de largo y ancho, con tres metros de alto. En la tabla 5.9 se resumen estas características.

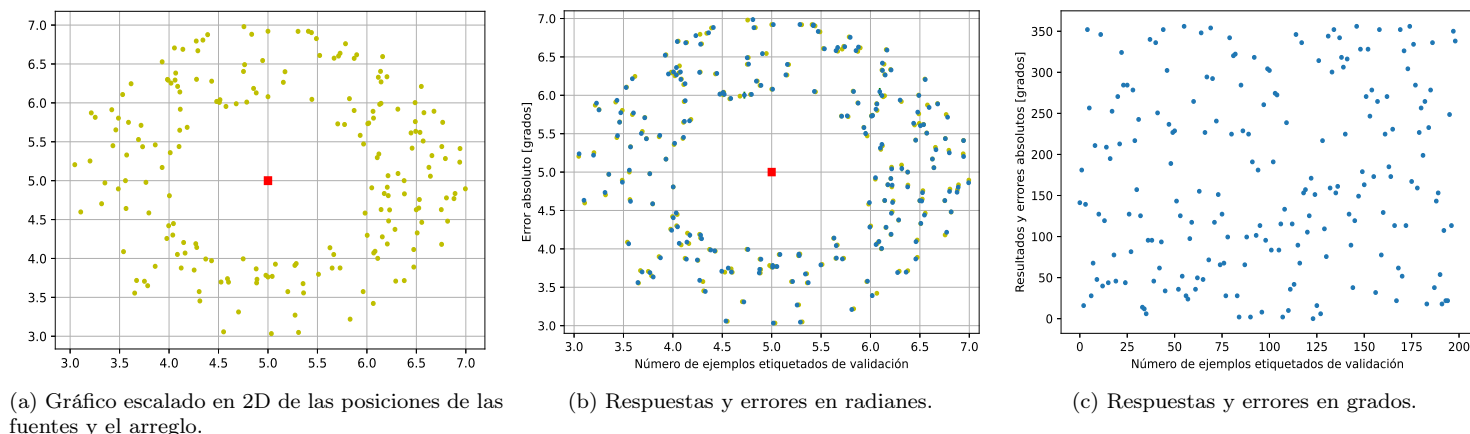


Figura 5.15: Dataset alimentado y el desempeño del sistema. UCA.

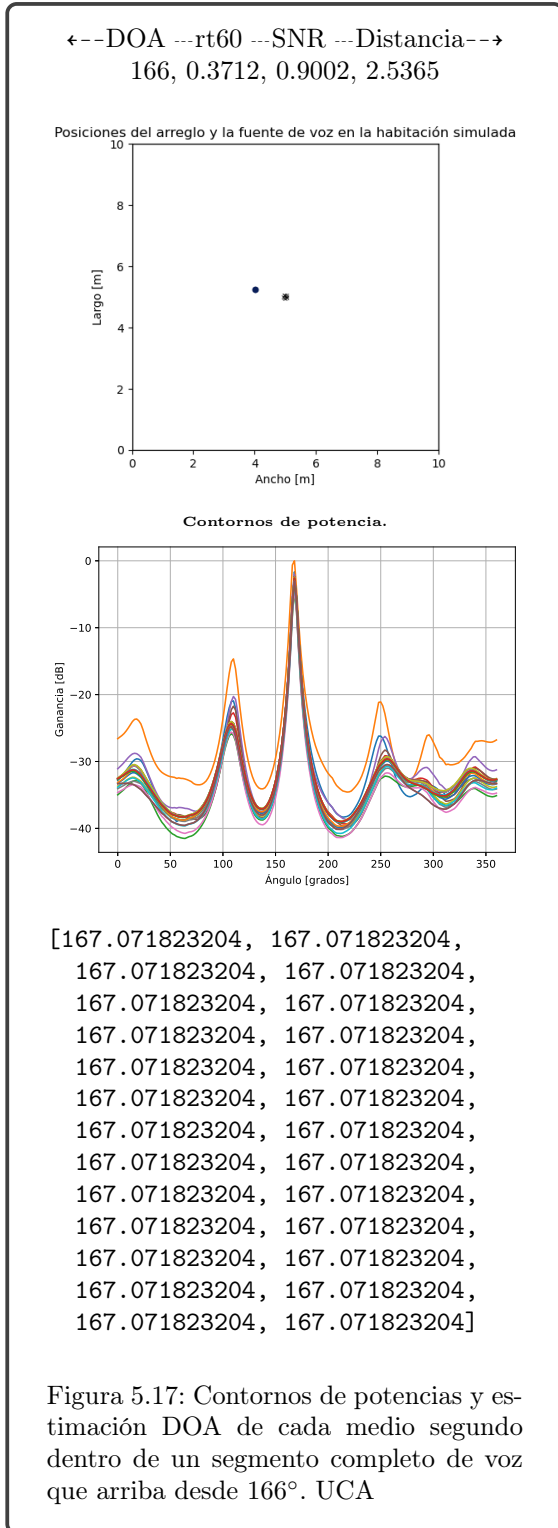
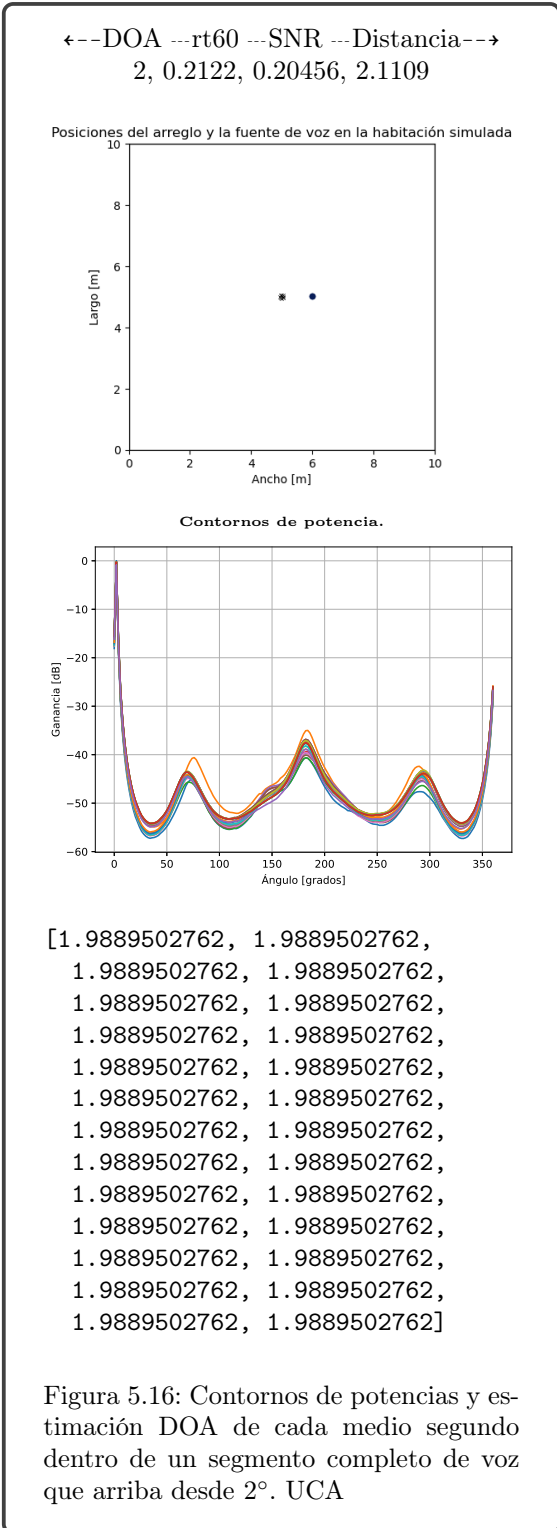
Desempeño	1.0
Media norma euclidiana	0.6627
Varianza norma euclidiana	0.2063
Mediana norma euclidiana	0.6298
Moda norma euclidiana	0.7071
Máximo norma euclidiana	2.7569
Mínimo norma euclidiana	0.0

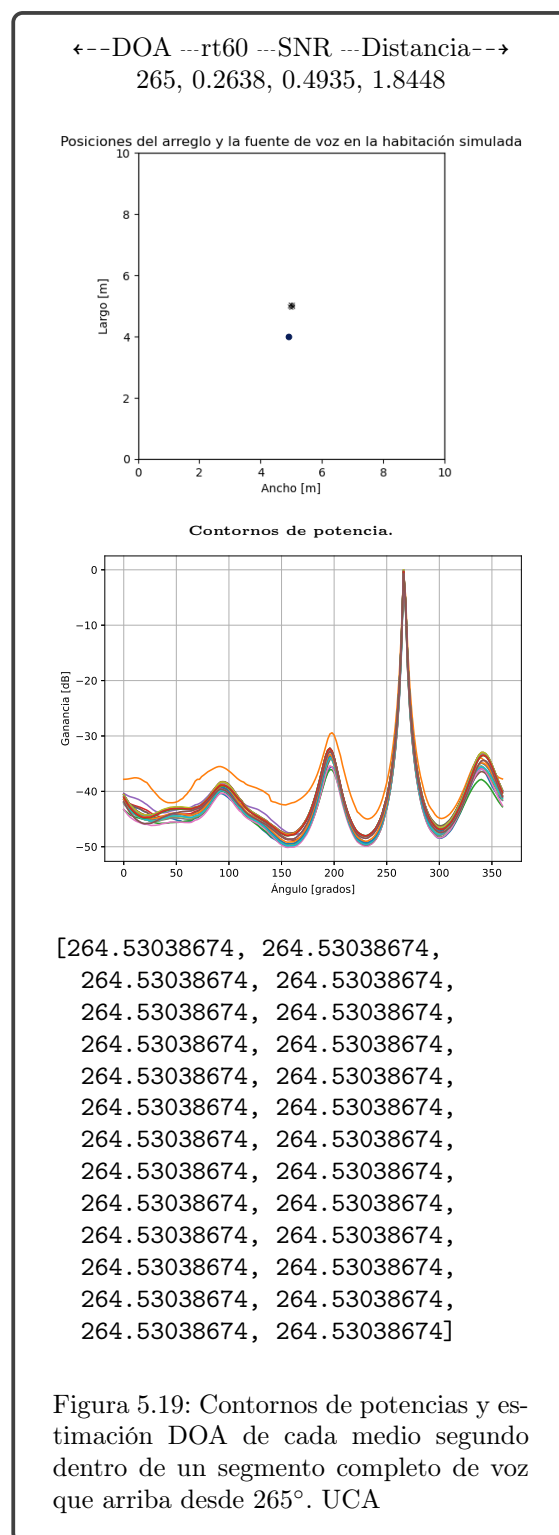
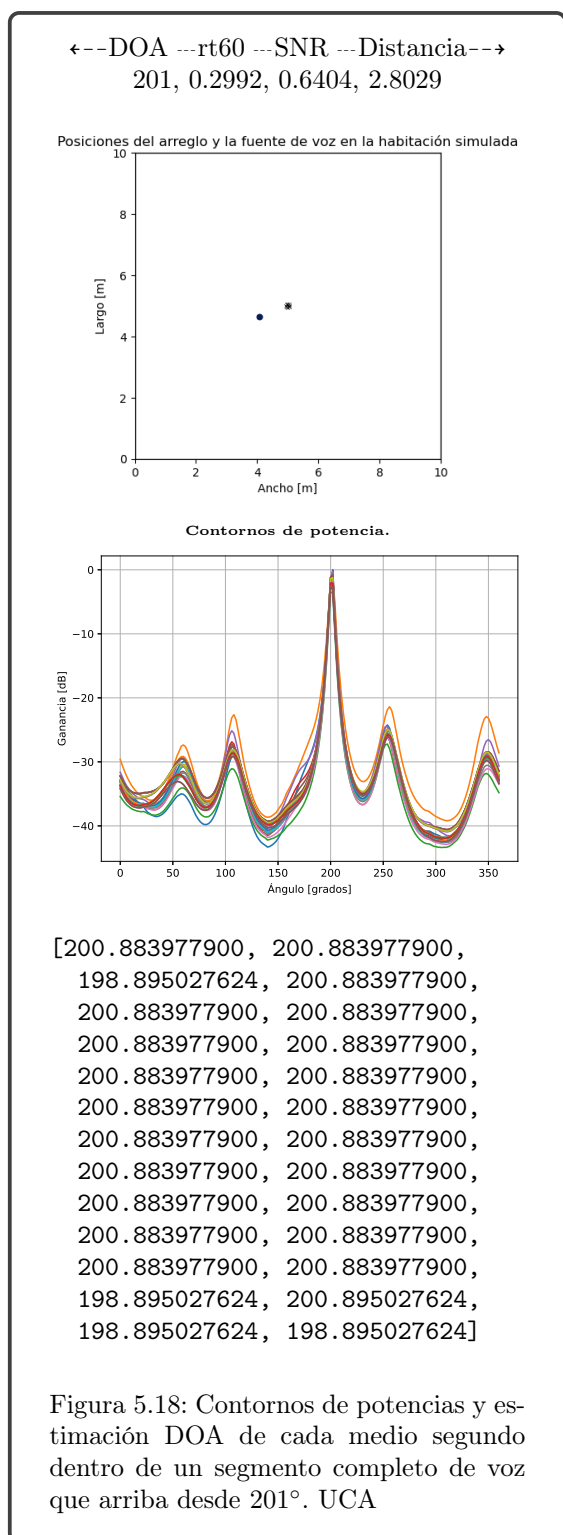
Tabla 5.8: Desempeño y valores de rendimiento del sistema. Dataset 200 ejemplos. UCA

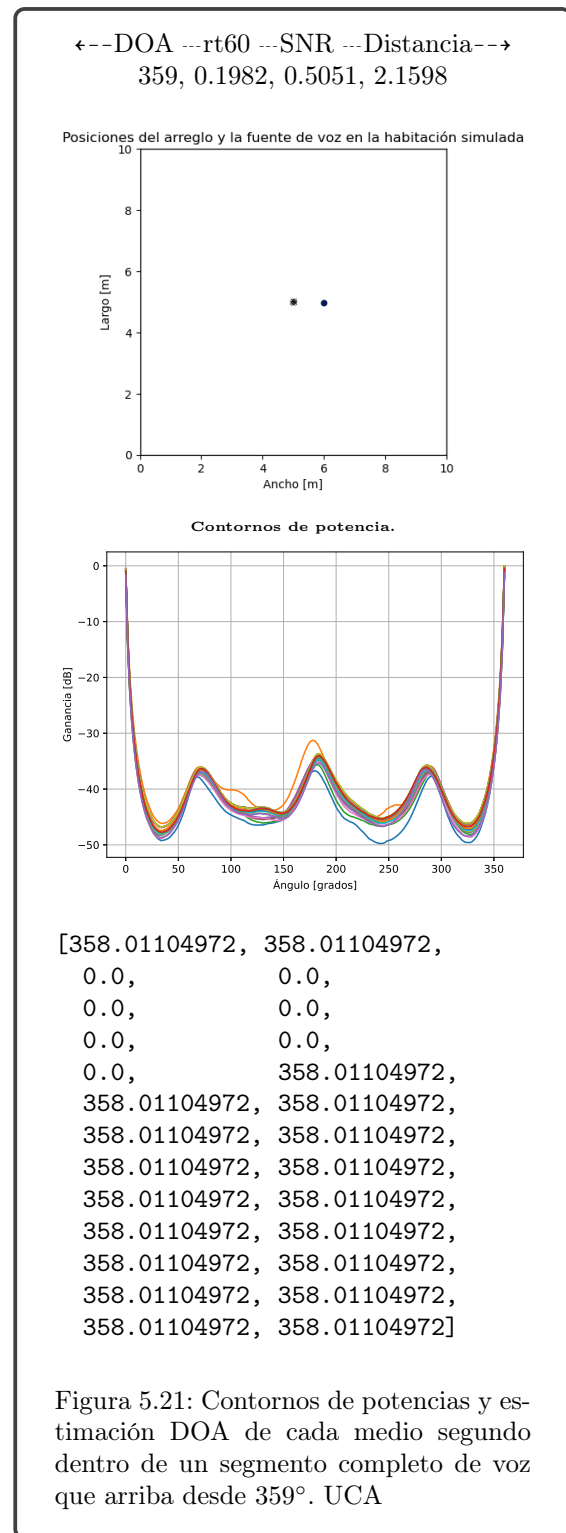
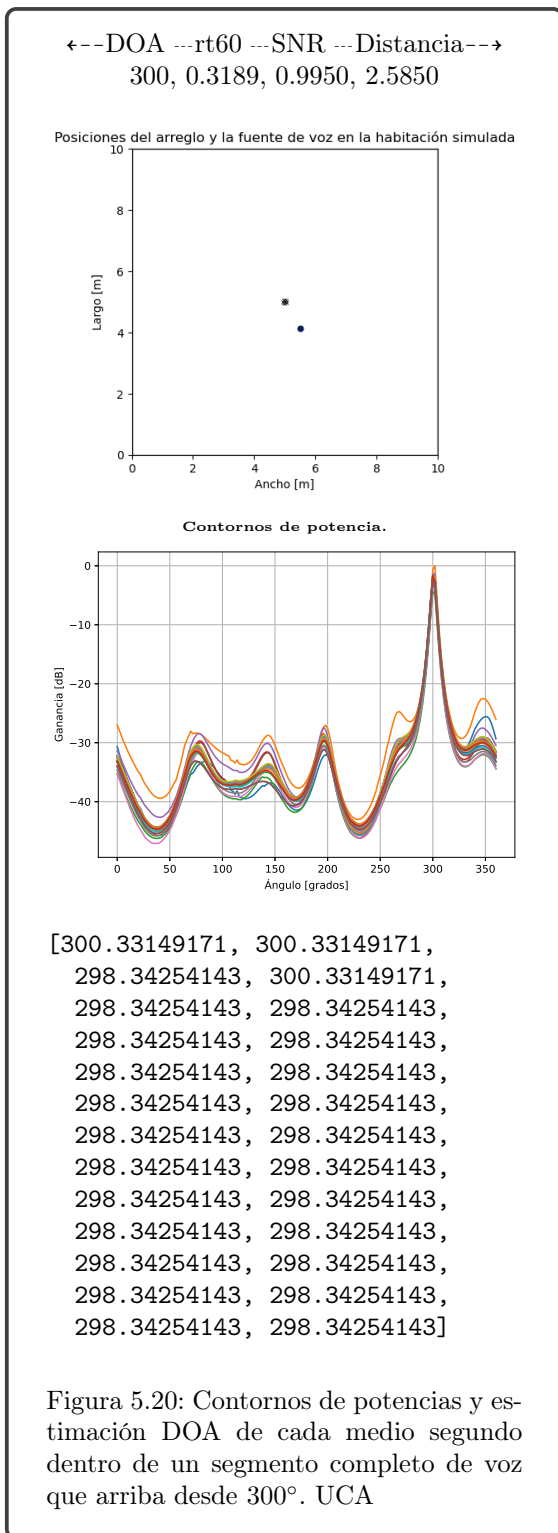
Parámetro	Valor
DOAs generados	2, 166, 201, 265, 300, 359
rt60 generados	0.2122, 0.3712, 0.2992, 0.2638, 0.3189, 0.1982
SNR generados	0.20456799, 0.90022008, 0.6404383, 0.49357913, 0.99502937, 0.5051867
Distancias generadas	2.1109143, 2.53654173, 2.80299607, 1.84484673, 2.58507624, 2.15980522

Tabla 5.9: Intervalos considerados para las pruebas de segmentos completos UCA

En las figuras 5.16 hasta 5.21 se pueden apreciar las posiciones relativas al UCA de las fuentes en cada caso; los contornos de potencias de cada medio segundo en que son divididos los segmentos completos de voz y por último se despliegan los valores máximos de cada contorno que corresponden a las estimaciones de medio segundo. De estos resultados se desprende que las estimaciones de medio segundo pertenecientes a una misma prueba de segmento completo de voz presentan una varianza pequeña y que logran un error menor a 5° . Por tanto el sistema es capaz de estimar exitosamente el DOA de señales de voz con sentencias completas que provienen de una fuente estática alejada considerablemente del UCA y en escenarios con valores altos de reverberaciones.







5.2.3. Efectos del aumento de las reverberaciones y la distancia

Las pruebas mostradas hasta este momento para el Neuro-Detector sujeto a un UCA se han realizado en condiciones exigentes de reverberación y distancias de las fuentes al arreglo. Muestra de esto se observa en los valores de $rt60$ y distancias contemplados para las pruebas de medio segundo de voz que se recogen en la tabla 5.6. Las pruebas de segmentos completos de voz también se realizaron en escenarios considerados exigentes. Sin embargo, en esta sección se muestran resultados para escenarios extremos de $rt60$ y las distancias para comprobar el comportamiento del sistema y cómo se afecta ante aumentos considerables de estos valores. Las pruebas que se muestran son de segmentos completos de voz y para ello se utilizan

los mismos archivos de audio y una habitación de iguales dimensiones que en pruebas similares anteriores.

En las figuras 5.22a hasta 5.22c se muestran los contornos de potencia de medio segundo en que se divide cada segmento completo de voz de una prueba y los valores máximos de estos contornos que representan las estimaciones DOA de medio segundo obtenidas. En las pruebas que se muestran en estas figuras, la distancia entre las fuentes y el UCA se mantuvo fija e igual a $d = 1.5m$, que es un valor pequeño; mientras se varía el $rt60$ para comprobar en qué medida afecta la precisión de las estimaciones. Como se observa, para un valor de distancia a lo sumo mediano, un aumento elevado de $rt60$ trae consigo la aparición de numerosos picos además del máximo, pero la diferencia de ganancia entre estos sigue siendo grande; por tanto el sistema en este tipo de escenario, mientras las fuentes estén cercanas no se verá afectado y la precisión se mantiene dentro de 5° . En este contexto, algunos valores de desempeño para el sistema fueron una media del error de 1.4908° , una varianza de 0.5768° y un valor máximo del error de 2.4419° , logrando un desempeño del 100 % para el total de estimaciones hechas en las tres pruebas.

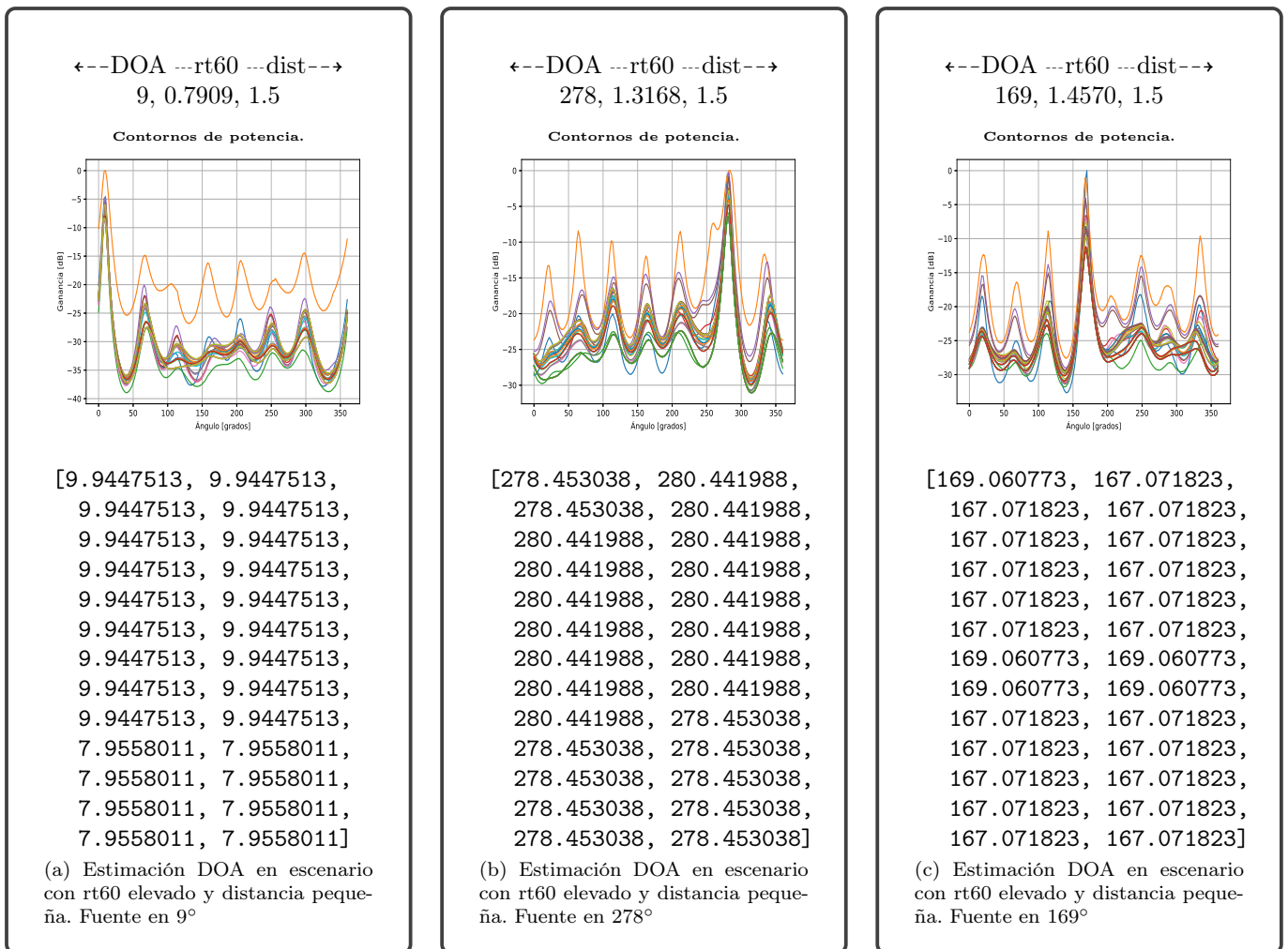


Figura 5.22: Efecto del aumento en $rt60$ para una distancia fija de $1.5m$ en la estimación DOA. UCA

Para el caso en que varía la distancia entre las fuentes y el UCA, manteniendo fijo el índice de reverberación, en un valor no muy elevado, $rt60 = 0.4s$; el sistema evidencia afectaciones notables que se muestran en el número de picos de los contornos de potencia y en la diferencia en la ganancia entre estos. El aumento en la anchura de los picos, más visible en la figura 5.23c, también es un efecto negativo que junto al número de estos puede llevar a errores de estimación del sistema. Las figuras 5.23a hasta 5.23c recogen el comportamiento de estos resultados. Para el total de estimaciones de medio segundo de estas pruebas el valor de la media en los errores cometidos fue de 2.2277° , con una varianza de 1.6712° y un valor máximo de 9.1767° que justifica un desempeño del 97.53 %. De este modo, para el Neuro-Detector

($moda = 194.9171^\circ$, $mediana = 194.9171^\circ$). El objetivo de estas pruebas, a pesar de estar alejadas de la realidad para las que son pensadas este tipo de aplicaciones, fue demostrar la robustez del sistema aún cuando está en ambientes completamente hostiles.

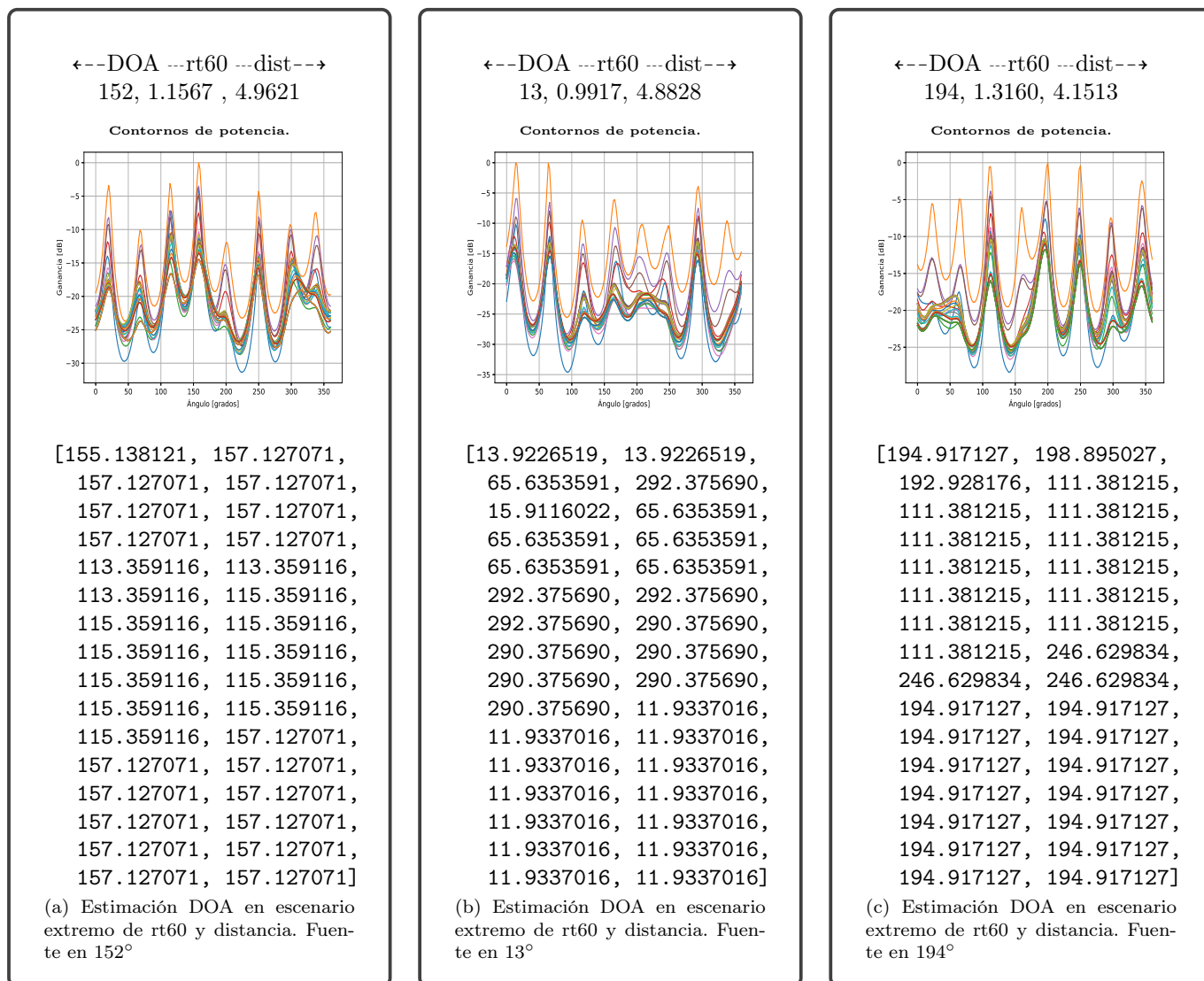


Figura 5.24: Respuestas del sistema sujeto a UCA para escenario extremo de rt60 y distancia.

5.3. Efectos de las variaciones en la SNR

A lo largo de este capítulo se han presentado pruebas y resultados para escenarios donde se varían las condiciones de reverberación, las direcciones desde donde inciden las señales de voz y las distancias desde donde se producen hasta el arreglo de micrófonos. Sin embargo no se ha destacado nada sobre los efectos de las variaciones en la relación señal-ruido. Observando los valores de SNR contemplados en todas las pruebas anteriores, notamos que estos siempre han sido considerados cercanos a $0dB$, a excepción de los dataset de entrenamiento y los dataset de validación, donde es importante tomar en cuenta un mayor rango de variación ya sea para dotar a la NN de una mejor capacidad para generalizar, o en el caso de la validación, para abarcar un mayor espacio muestral. Por tanto, se pudiera asegurar que el algoritmo desarrollado presenta robustez ante las variaciones en la SNR, la razón de esto radica en el empleo de la NN MLP para estimar el rango efectivo de la matriz de correlaciones \mathbf{R}_{xx} a la salida del Bf. Entonces, la MLP está distinguiendo correctamente en cada caso el subespacio de señal del de solo-ruido. De esta forma, aún cuando los niveles de ruido son equiparables a los de la señal de voz de interés, la NN y por tanto el sistema es capaz de discriminarlos exitosamente.

Para tomar en cuenta las variaciones de SNR en los dataset y escenarios generados en general, durante la tesis se calculó la varianza del ruido empleando la siguiente expresión [6]:

$$\sigma = \frac{10^{(-SNR/10)}}{(4.0\pi D)^2} \quad (5.1)$$

donde SNR es el valor que se desea en el escenario a simular y D es la distancia de la fuente de voz al arreglo en cuestión. El valor de varianza del ruido, calculado a partir de la ecuación 5.1, es utilizado por la librería `pyroomacoustics` [39] para generar ruido blanco gaussiano, el cual es adicionado a las señales de voz dentro de la habitación generada.

5.4. Comparación de los resultados

En aras de establecer comparaciones entre el desempeño del algoritmo Neuro-Detector sujeto a ULA y sujeto a UCA es importante hacer notar que para todas las pruebas realizadas en este capítulo, se consideraron, para ambas geometrías del arreglo, el mismo número de matrices de restricción en el barrido de las posibles direcciones de arriba; debido a esto, el Neuro-Detector sujeto a UCA ha tenido una desventaja con relación a la precisión de estimación debido al número de direcciones de arriba consideradas. Es decir, para el caso del sistema sujeto a UCA el barrido abarca un número menor de posibles direcciones, una dirección cada 2° , lo que fija este valor como cota superior de estimación. En contraste, en el caso del ULA, el barrido de direcciones es cada 1° . Esto se refleja en la tabla 5.10.

	Neuro-Detector ULA	Neuro-Detector UCA
Intervalo de posibles direcciones de arriba	$[-90^\circ, 90^\circ]$	$[0^\circ, 359^\circ]$
Número de direcciones consideradas en el barrido (Cantidad de matrices de restricción)	181	181
Resolución del barrido de direcciones	1°	2°
Cota superior en la precisión de estimación que establece la relación entre el Intervalo de posibles direcciones y el Número de matrices de restricción	1°	2°

Tabla 5.10: Sesgo de incertidumbre en la estimación DOA del sistema para los dos casos de arreglo considerados

Aclarado esto, en la tabla 5.11 se recogen los valores de desempeño y rendimiento para la mayoría de las pruebas realizadas en este capítulo. Además, en la figura 5.25 se muestra una última prueba de medio segundo para cada caso, en donde se generaron dos datasets de 500 ejemplos etiquetados cada uno, bajo un mismo escenario de reverberaciones, distancias y de dimensiones de la habitación. Se generaron empleando un archivo de audio de AMI corpus distinto al del resto de las pruebas. El objetivo de esta prueba fue comparar el desempeño de ambos sistemas ante un dataset que incluya los mismos intervalos para `rt60` y las distancias entre las fuentes y el arreglo. En las figuras 5.25b y 5.25a se muestran las similitudes de los datasets; se grafican las posiciones (color magenta/amarillo) en dos dimensiones de las fuentes entorno al arreglo en cada caso y se superponen las posiciones (color azul) y los errores (color verde) obtenidos a partir de las respuestas DOAs de los sistemas. Por último se muestran los valores de rendimiento.

Como se observa, a pesar que para el caso del UCA el sistema debe estimar el DOA con un sesgo de incertidumbre mayor, el resumen de resultados de la tabla 5.11 y las pruebas de la figura 5.25 corroboran que el sistema Neuro-Detector en este caso es más robusto ante variaciones en las condiciones de operación y en cuanto a la precisión de estimación. Como se desprende de esta sección, la selección de la geometría determina en gran medida el desempeño de un algoritmo; para el Neuro-Detector desarrollado en esta tesis, con la geometría UCA se logran mejores resultados. Esto, en cierto modo era esperado porque ha sido demostrado su buen rendimiento para tareas de estimación DOA que integran AI y aprendizaje profundo en general [20],[21],[33],[41], además, en [42] se abordan situaciones donde es superior respecto al resto de geometrías en términos de resolución, robustés en ambientes ruidosos y directividad

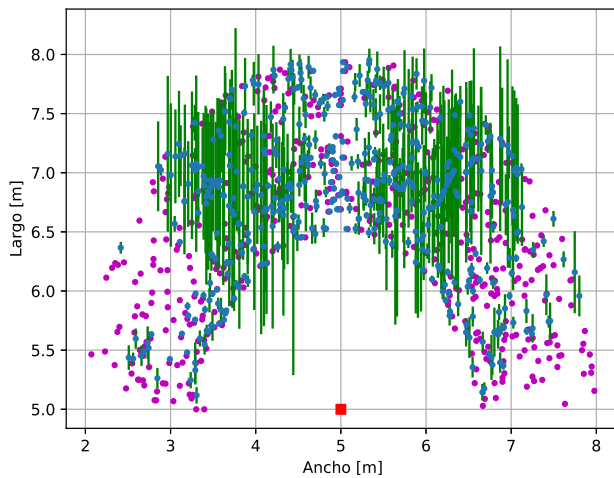
Pruebas de medio segundo de voz		
	Neuro-Detector ULA	Neuro-Detector UCA
<ul style="list-style-type: none"> ■ Dataset 2707 ejemplos ■ Dataset 200 ejemplos 	$rt60 \in [0.18, 0.35] [s]$ $D \in [1.0, 2.0] [m]$ <ul style="list-style-type: none"> ■ Desempeño: 98 % Media $E_{ }$: 1.1798 Varianza $E_{ }$: 1.0941 Mediana $E_{ }$: 0.8342 Moda $E_{ }$: 0.5082 Máximo $E_{ }$: 9.9447 Mínimo $E_{ }$: 0.0000 	$rt60 \in [0.18, 1.0] [s]$ $D \in [1.5, 3.0] [m]$ <ul style="list-style-type: none"> ■ Desempeño: 99 % Media $E_{ }$: 1.0819 Varianza $E_{ }$: 1.5323 Mediana $E_{ }$: 0.7458 Moda $E_{ }$: 0.1988 Máximo $E_{ }$: 30.585 Mínimo $E_{ }$: 0.0000
	$rt60 \in [0.18, 0.35] [s]$ $D \in [1.0, 2.0] [m]$ <ul style="list-style-type: none"> ■ Desempeño 100 % Media $E_{ }$: 1.1450 Varianza $E_{ }$: 0.9571 Mediana $E_{ }$: 0.8342 Moda $E_{ }$: 0.8397 Máximo $E_{ }$: 4.9668 Mínimo $E_{ }$: 0.0441 	$rt60 \in [0.18, 0.35] [s]$ $D \in [1.0, 2.0] [m]$ <ul style="list-style-type: none"> ■ Desempeño 100 % Media $E_{ }$: 0.6627 Varianza $E_{ }$: 0.2063 Mediana $E_{ }$: 0.6298 Moda $E_{ }$: 0.7071 Máximo $E_{ }$: 2.7569 Mínimo $E_{ }$: 0.0000
Pruebas de segmentos completos de voz		
	Neuro-Detector ULA	Neuro-Detector UCA
<ul style="list-style-type: none"> ■ Dos archivos de audio concatenados. Cerca de 50 [s]. ■ Dos archivos de audio concatenados. Cerca de 50 [s]. 	$rt60 = [0.8364, 0.6364, 0.9950] [s]$ $D = [1.0, 1.0, 1.0] [m]$ <ul style="list-style-type: none"> ■ Desempeño: 100 % Media $E_{ }$: 1.7466 Varianza $E_{ }$: 0.6456 Mediana $E_{ }$: 1.7734 Moda $E_{ }$: 1.7734 Máximo $E_{ }$: 3.9005 Mínimo $E_{ }$: 0.0773 	$rt60 = [0.7909, 1.3168, 1.4570] [s]$ $D = [1.5, 1.5, 1.5] [m]$ <ul style="list-style-type: none"> ■ Desempeño: 100 % Media $E_{ }$: 1.4908 Varianza $E_{ }$: 0.5768 Mediana $E_{ }$: 1.9281 Moda $E_{ }$: 1.9281 Máximo $E_{ }$: 2.4419 Mínimo $E_{ }$: 0.0607
	$rt60 = [0.4, 0.4, 0.4] [s]$ $D = [1.8324, 2.4467, 3.2511] [m]$ <ul style="list-style-type: none"> ■ Desempeño 28.20 % Media $E_{ }$: 8.7603 Varianza $E_{ }$: 17.4453 Mediana $E_{ }$: 8.7624 Moda $E_{ }$: 3.6795 Máximo $E_{ }$: 16.7458 Mínimo $E_{ }$: 2.6850 	$rt60 = [0.4, 0.4, 0.4] [s]$ $D = [3.6896, 4.5774, 4.6934] [m]$ <ul style="list-style-type: none"> ■ Desempeño 97.53 % Media $E_{ }$: 2.2277 Varianza $E_{ }$: 1.6712 Mediana $E_{ }$: 1.5414 Moda $E_{ }$: 2.6022 Máximo $E_{ }$: 9.1767 Mínimo $E_{ }$: 0.4475

Tabla 5.11: Resumen de resultados del sistema para ambas geometrías de arreglo.

←--NN-Detector--ULA--→

Dataset. Número de ejemplos	500
Dimensiones de la habitación	[10, 10, 3][m]
Intervalo de direcciones de arribo	$[-90^\circ, 90^\circ]$
Intervalo del índice de reverberación $rt60$	[0.18, 1.0][s]
SNR	[0.0, 20][dB]
Intervalo de distancia de la fuente de voz al arreglo (D)	[1.5, 3.0] [m]
Posición del arreglo	(5, 5, 1.5) [m]
Posición de la fuente de voz	(5, 5, 1.5) + $D \cdot \overrightarrow{DOA}$ [m]

Superposición de etiquetas DOAs y respuestas del sistema.



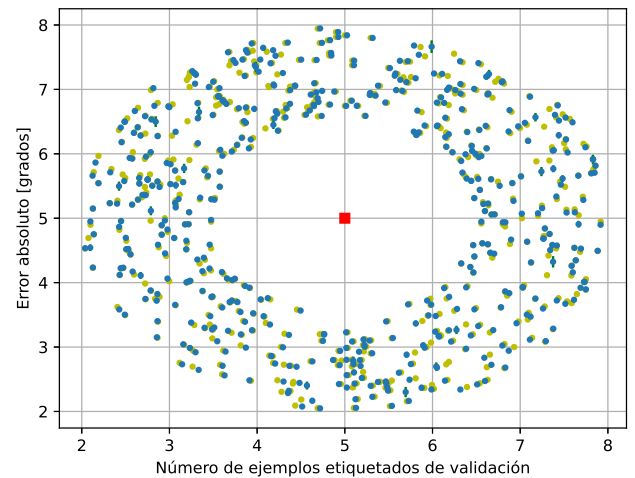
Desempeño	0.4609
Media error absoluto	12.0282
Varianza error absoluto	189.7156
Mediana error absoluto	5.7071
Moda error absoluto	0.4972
Máximo error absoluto	69.6132
Mínimo error absoluto	0.1491

(a) Desempeño en escenario altos $rt60$ y distancia. ULA

←--NN-Detector--UCA--→

Dataset. Número de ejemplos	500
Dimensiones de la habitación	[10, 10, 3][m]
Intervalo de direcciones de arribo	$[0^\circ, 359^\circ]$
Intervalo del índice de reverberación $rt60$	[0.18, 1.0][s]
SNR	[0.0, 20][dB]
Intervalo de distancia de la fuente de voz al arreglo (D)	[1.5, 3.0] [m]
Posición del arreglo	(5, 5, 1.5) [m]
Posición de la fuente de voz	(5, 5, 1.5) + $D \cdot \overrightarrow{DOA}$ [m]

Superposición de etiquetas DOAs y respuestas del sistema.



Desempeño	0.9959
Media norma euclidiana	0.9705
Varianza norma euclidiana	0.8800
Mediana norma euclidiana	0.6298
Moda norma euclidiana	0.6795
Máximo norma euclidiana	5.6132
Mínimo norma euclidiana	0.0055

(b) Desempeño en escenario altos $rt60$ y distancia. UCAFigura 5.25: Desempeño del sistema sujeto a ULA (a) y a UCA (b) bajo las mismas condiciones: altos valores de $rt60$ y distancia.

5.5. Tiempos de respuesta

El algoritmo Neuro-Detector desarrollado en esta tesis desde su entrenamiento hasta las pruebas de validación han involucrado etapas y consideraciones para su implementación en tiempo real. Estas abarcan desde detección de actividad de voz, rastreo de fuentes de voz y la optimización del código Python con librerías y funciones lo más sencillapus y cercanas a sus paralelas en lenguaje C/C++. Por tanto para este trabajo es muy importante los tiempos de procesamiento y respuesta logrados.

Para conocer los tiempos de ejecución y respuesta del sistema se utilizó un ambiente computacional interactivo llamado Jupyter Notebook o formalmente IPython Notebook, el cual presenta los comandos intrínsecos `%%timeit` y `%timeit` para medir el tiempo de ejecución de una celda o una línea de código respectivamente. Las características del sistema de cómputo empleado para la implementación y pruebas se recogen en el apéndice A, con base a estas, el tiempo de respuesta del sistema para procesar medio segundo de voz y estimar el DOA se muestra en la tabla 5.12 con respaldo en la figura 5.26.

Operación	Tiempo
Procesamiento total - Estimación DOA	240 ms \pm 18.7 ms per loop (mean \pm std. dev. of 7 runs, 1 loop each)
Procesamiento del Beamformer (Cálculo de $\mathbf{R}_{xx} \in \mathbb{R}^{384}$)	83.7 ms \pm 7.49 ms per loop (mean \pm std. dev. of 7 runs, 10 loops each)

Tabla 5.12: Tiempo de respuesta del sistema para estimar el DOA de medio segundo de voz.

```
In [591]: 1 fs,data_test = wavfile.read("voz_lab.wav")
          2 test = dsa.dataSetAudio(111,0.27,0,1.8)
          3 M_Signals = test.SimuData(_,data_test,fs,M,array_mic="UCA",dataset=False)
          4 M_S=M_Signals[:, :8000]
          5
          6 %timeit an,ind = ndd.estDoa(M_S, M, J, NumAng, Cc_list, nnk2,array_mic="UCA")
          7
          8 an

240 ms  $\pm$  18.7 ms per loop (mean  $\pm$  std. dev. of 7 runs, 1 loop each)

Out[591]: [111.38121546961327]
```

Figura 5.26: Tiempo consumido para estimar el DOA de medio segundo de voz.

Con los datos de la tabla 5.12 se obtiene que en **156.3 [ms]** la MLP calcula el rango efectivo de la matriz de correlaciones, se definen los subespacios de señal y solo-ruido y se proyectan los espacios de representación sobre el subespacio de señal. En la figura 5.27 se muestra el tiempo de respuesta para el caso de un segundo de voz y en el apéndice A se muestra para el caso de un archivo completo de voz de 12 s de duración. Dado que los tiempos de respuesta representan cerca de la mitad del segmento de voz a procesar, de forma secuencial: a la vez que se realiza el procesamiento del segmento actual, se puede ir muestreando y almacenando el siguiente segmento de voz; de modo que el sistema podría ser empleado en línea si se mantiene en esta versión desarrollada en Python. Es seguro que una vez migrado el sistema a código C/C++ pueda trabajar en tiempo real.

```
In [599]: 1 fs,data_test = wavfile.read("voz_lab.wav")
          2 test = dsa.dataSetAudio(349,0.5,0,2.2)
          3 M_Signals = test.SimuData(_,data_test,fs,M,array_mic="UCA",dataset=False)
          4 M_S=M_Signals[:, :24000:40000]
          5
          6 %timeit an,ind = ndd.estDoa(M_S, M, J, NumAng, Cc_list, nnk2,array_mic="UCA")
          7
          8 an

524 ms  $\pm$  1.69 ms per loop (mean  $\pm$  std. dev. of 7 runs, 1 loop each)

Out[599]: [348.0662983425414, 348.0662983425414]
```

Figura 5.27: Tiempo consumido para estimar el DOA de un segundo de voz.

5.6. Estimación DOA para varias fuentes incidentes

Como se abordó en la sección 3.5, el número de fuentes a las que se puede estimar su DOA es menor que el número de micrófonos en el arreglo [31] y en dicho caso, el requerimiento y recursos computacionales necesarios para el entrenamiento del sistema es elevado. No obstante, el algoritmo Neuro-Detector desarrollado, sin haberse considerado múltiples fuentes para su entrenamiento, logra realizar esta tarea de estimación múltiple. La razón de esta capacidad radica una vez más en la correcta separación de los subespacios de señal y de solo-ruido, lo cual es consecuencia directa del rango efectivo de la matriz de correlaciones \mathbf{R}_{xx} , el cual es determinado por la MLP a partir de la estimación del ruido y las variaciones estadísticas de la matriz. Una segunda razón estriba en el estimador empleado para determinar los contornos de potencia a partir de las proyecciones de los espacios de representación de la señal de voz en cada dirección del arribo sobre el subespacio de señal. Este estimador (ecuación 4.6) presenta gran robustés ante problemas de escala en la potencia estimada, lo que es una ventaja para la estimación DOA múltiple, ya que minimiza el problema que representa el hecho de que para más de una fuente el número de vectores base en el espacio de representación de cada una, que yace cerca del subespacio de señal, es diferente y esto acarrea problemas de escala y por ende de estimación [10].

En las figuras 5.28 y 5.29 se muestran los resultados para la estimación DOA de dos y tres fuentes de voz respectivamente, que inciden a la vez sobre el arreglo. Las dimensiones de la habitación para la generación de las pruebas es la misma que en pruebas anteriores, los valores de $rt60$ y las distancias consideradas se muestran en cada figura. Los archivos de audio empleados fueron generados a partir de grabaciones de mi voz en un ambiente no controlado.

Como se observa, los resultados en la estimación son satisfactorios, sobre todo cuando el conjunto de valores de $rt60$ y de las distancias no son demasiado elevados. Para el caso de tres fuentes de voz, donde las reverberaciones son altas y las distancias moderadas, figura 5.29b, existen estimaciones de medio segundo en que solo se detecta el DOA de dos fuentes. Esto, como muestran los contornos de potencia de medio segundo es debido a la disminución de la diferencia en los decibeles de ganancia de los picos máximos, lo cual es el resultado del valor elevado de $rt60$ principalmente.

De forma general, como muestran estas figuras, los valores DOA devueltos por el sistema son de una precisión de menos de 5° , por lo que el sistema tiene un desempeño aceptable para la estimación de múltiples fuentes.

Resumen

En este capítulo se implementó el algoritmo Neuro-Detector DOA desarrollado para dos geometrías diferentes de arreglos: lineal uniforme y circular uniforme. Se mostraron numerosas pruebas y sus resultados para medir el desempeño de los sistemas en diversos escenarios generados mediante software y bajo variadas condiciones acústicas simuladas. Los resultados presentados evidencian que el algoritmo para ambas geometrías estima el DOA de una fuente de voz con una precisión menor a 5° en escenarios bajo condiciones acústicas normales a rigurosas, mientras que para condiciones extremas de reverberación y lejanía de las fuentes el sistema en ambos casos presenta una respuesta, como mínimo, capaz de determinar el sentido del arribo (derecha, izquierda) en el caso del sistema con geometría ULA y en el caso del sistema con geometría UCA es capaz de dar un estimado aceptable basado en descriptores estadísticos de los resultados de medio segundo de voz, como es la moda. Se establece una comparación entre el rendimiento obtenido por el sistema para ambas geometrías, siendo el Neuro-Detector para UCA más robusto ante variaciones de reverberación, distancias y número de fuentes a estimar su DOA. Los tiempos de respuesta para ambos casos es similar, siendo de 240 ms para procesar y estimar el DOA de medio segundo de voz y de 524 ms en el caso de un segundo de voz. Por lo que el sistema puede ser utilizado en línea. Por último, se mostraron resultados satisfactorios en el caso de la estimación DOA de dos fuentes y hasta tres fuentes estáticas incidentes. Se llevaron a cabo pruebas de rendimiento de medio segundo de voz cuyas direcciones de incidencia y condiciones acústicas varían en cada caso para entrenar y validar el sistema para el restreo de fuentes sonoras móviles; por tanto, se espera que el sistema sea capaz de seguir el desplazamiento de fuentes en movimiento.

Capítulo 6

Conclusiones

En este trabajo se diseñó e implementó un algoritmo para la localización de señales de voz que no es restrictivo a una configuración particular del arreglo de micrófonos. El algoritmo Neuro-Detector DOA desarrollado y probado para dos geometrías diferentes de arreglos: lineal uniforme y circular uniforme; fue sometido a numerosas pruebas y sus resultados fueron analizados para medir el desempeño de los sistemas en diversos escenarios generados mediante software y bajo variadas condiciones acústicas simuladas. La librería [39] que sirve de núcleo para todas las pruebas generadas tiene como abal publicaciones en IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) y en la Universidad Cornell por lo que, a pesar de no ser pruebas llevadas a cabo con arreglos de micrófonos reales y en un escenario acústico real, estas se consideran generadas con una elevada fidelidad y calidad en la reproducción de las características de escenarios acústicos cotidianos de la vida real y por tanto afianzan el análisis y los resultados obtenidos.

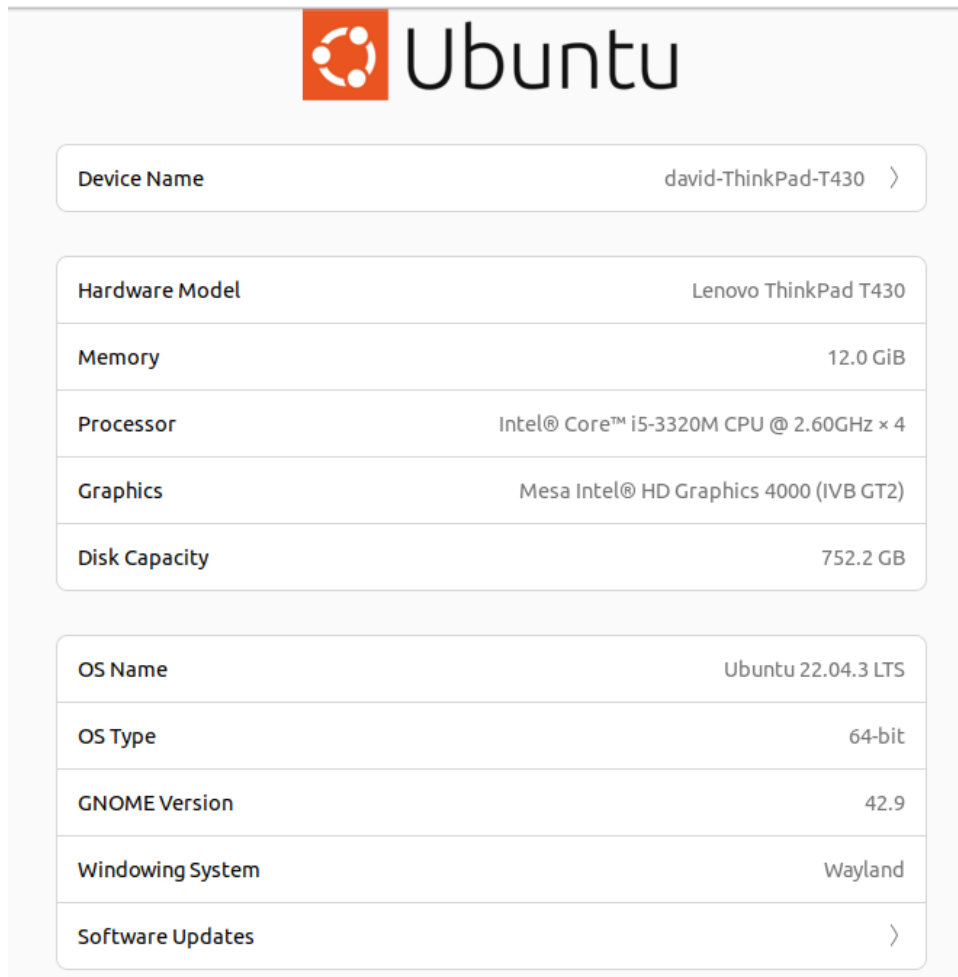
Los resultados presentados evidencian que el algoritmo para ambas geometrías estima el DOA de una fuente de voz con una precisión menor a 5° en escenarios bajo condiciones acústicas normales a rigurosas, mientras que para condiciones extremas de reverberación y lejanía de las fuentes el sistema en ambos casos presenta una respuesta, como mínimo, capaz de determinar el sentido del arribo (derecha, izquierda) en el caso del sistema con geometría ULA y en el caso del sistema con geometría UCA es capaz de dar un estimado aceptable basado en descriptores estadísticos de los resultados de medio segundo de voz, como es la moda. Se estableció una comparación entre el rendimiento obtenido por el sistema para ambas geometrías, siendo el Neuro-Detector para UCA más robusto ante variaciones de reverberación, distancias y número de fuentes a estimar su DOA. Los tiempos de respuesta para ambos casos es similar, siendo de 240 ms para procesar y estimar el DOA de medio segundo de voz y de 524 ms en el caso de un segundo de voz. Por lo que el sistema puede ser utilizado en línea. Por último, se mostraron resultados satisfactorios en el caso de la estimación DOA de dos fuentes y hasta tres fuentes estáticas incidentes. Se llevaron a cabo pruebas de rendimiento de medio segundo de voz cuyas direcciones de incidencia y condiciones acústicas varían en cada caso para entrenar y validar el sistema para el restreo de fuentes sonoras móviles; por tanto, se espera que el sistema sea capaz de seguir el desplazamiento de fuentes en movimiento.

Se concluye además que los objetivos propuestos fueron cumplidos, logrando integrar exitosamente una red neuronal y la combinación de dos de los métodos más convencionales de estimación DOA para implementar un algoritmo robusto ante variaciones estadísticas y el ruido, capaz de detectar satisfactoriamente la dirección de arribo de una o varias fuentes de voz en condiciones acústicas variadas.

Apéndice A

Anexos

A.1. Características del hardware del sistema de cómputo



The image shows a screenshot of the Ubuntu system information page. At the top, there is the Ubuntu logo and the word "Ubuntu". Below this, there are two main sections of system information, each presented as a table with rounded corners and a light gray background.

Hardware	
Device Name	david-ThinkPad-T430 >
Hardware Model	Lenovo ThinkPad T430
Memory	12.0 GiB
Processor	Intel® Core™ i5-3320M CPU @ 2.60GHz × 4
Graphics	Mesa Intel® HD Graphics 4000 (IVB GT2)
Disk Capacity	752.2 GB

Software	
OS Name	Ubuntu 22.04.3 LTS
OS Type	64-bit
GNOME Version	42.9
Windowing System	Wayland
Software Updates	>

Figura A.1: Sistema de cómputo empleado en la implementación y pruebas del algoritmo.

A.2. Tiempo de respuesta para 12 segundos de voz.

```
In [609]: 1 fs,data_test = wavfile.read("voz_lab.wav")
          2 test = dsa.dataSetAudio(349,0.5,0,2.2)
          3 M_Signals = test.SimuData(_,data_test,fs,M,array_mic="UCA",dataset=False)
          4 # M_S=M_Signals[:,24000:40000]
          5
          6 %timeit an,ind = ndd.estDoa(M_Signals, M, J, NumAng, Cc_list, nnk2,array_mic="UCA")
          7
          8
          4.74 s ± 555 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

Figura A.2: Tiempo de respuesta del sistema para un archivo de audio de doce segundos de duración.

A.3. Comparación etiquetas DOAs y resultados del sistema.

Los datos presentados a continuación se muestran a modo de comparación. Están dispuestos en pares de datos de comparación, estando siempre el valor deseado (etiqueta DOA) encima y la respuesta resultante del sistema debajo (estimación). Cada columna está compuesta por varios pares de datos de comparación.

etiqueta DDA: -->	-29.000000	etiqueta DDA: -->	52.000000	etiqueta DDA: -->	74.000000	etiqueta DDA: -->	-10.000000	etiqueta DDA: -->	6.000000	etiqueta DDA: -->	-69.000000
estimación DDA:-->	-28.342541	estimación DDA:-->	51.215470	estimación DDA:-->	73.093923	estimación DDA:-->	-10.441989	estimación DDA:-->	4.475138	estimación DDA:-->	-69.116022
etiqueta DDA: -->	21.000000	etiqueta DDA: -->	-5.000000	etiqueta DDA: -->	-20.000000	etiqueta DDA: -->	-50.000000	etiqueta DDA: -->	88.000000	etiqueta DDA: -->	-34.000000
estimación DDA:-->	19.392265	estimación DDA:-->	-5.469613	estimación DDA:-->	-19.392265	estimación DDA:-->	-50.220994	estimación DDA:-->	85.027624	estimación DDA:-->	-34.309392
etiqueta DDA: -->	-54.000000	etiqueta DDA: -->	5.000000	etiqueta DDA: -->	-27.000000	etiqueta DDA: -->	50.000000	etiqueta DDA: -->	-34.000000	etiqueta DDA: -->	-53.000000
estimación DDA:-->	-54.198895	estimación DDA:-->	3.480663	estimación DDA:-->	-26.353591	estimación DDA:-->	49.226519	estimación DDA:-->	-33.314917	estimación DDA:-->	-53.204420
etiqueta DDA: -->	-28.000000	etiqueta DDA: -->	-87.000000	etiqueta DDA: -->	17.000000	etiqueta DDA: -->	75.000000	etiqueta DDA: -->	-90.000000	etiqueta DDA: -->	26.000000
estimación DDA:-->	-27.348066	estimación DDA:-->	-81.049724	estimación DDA:-->	15.414365	estimación DDA:-->	73.093923	estimación DDA:-->	-84.033149	estimación DDA:-->	24.364641
etiqueta DDA: -->	40.000000	etiqueta DDA: -->	-25.000000	etiqueta DDA: -->	-59.000000	etiqueta DDA: -->	72.000000	etiqueta DDA: -->	45.000000	etiqueta DDA: -->	31.000000
estimación DDA:-->	39.281768	estimación DDA:-->	-24.364641	estimación DDA:-->	-59.171271	estimación DDA:-->	70.110497	estimación DDA:-->	44.254144	estimación DDA:-->	28.342541
etiqueta DDA: -->	-32.000000	etiqueta DDA: -->	-20.000000	etiqueta DDA: -->	69.000000	etiqueta DDA: -->	72.000000	etiqueta DDA: -->	72.000000	etiqueta DDA: -->	-71.000000
estimación DDA:-->	-31.325967	estimación DDA:-->	-19.392265	estimación DDA:-->	67.127072	estimación DDA:-->	70.110497	estimación DDA:-->	70.110497	estimación DDA:-->	-70.110497
etiqueta DDA: -->	-42.000000	etiqueta DDA: -->	-82.000000	etiqueta DDA: -->	15.000000	etiqueta DDA: -->	-27.000000	etiqueta DDA: -->	24.000000	etiqueta DDA: -->	57.000000
estimación DDA:-->	-41.270718	estimación DDA:-->	-81.049724	estimación DDA:-->	13.425414	estimación DDA:-->	-26.353591	estimación DDA:-->	22.375691	estimación DDA:-->	55.193370
etiqueta DDA: -->	-78.000000	etiqueta DDA: -->	-20.000000	etiqueta DDA: -->	44.000000	etiqueta DDA: -->	-86.000000	etiqueta DDA: -->	-3.000000	etiqueta DDA: -->	-25.000000
estimación DDA:-->	-78.066298	estimación DDA:-->	-19.392265	estimación DDA:-->	43.259669	estimación DDA:-->	-84.033149	estimación DDA:-->	-3.480663	estimación DDA:-->	-25.359116
etiqueta DDA: -->	-85.000000	etiqueta DDA: -->	64.000000	etiqueta DDA: -->	-16.000000	etiqueta DDA: -->	-57.000000	etiqueta DDA: -->	29.000000	etiqueta DDA: -->	30.000000
estimación DDA:-->	-84.033149	estimación DDA:-->	63.149171	estimación DDA:-->	-15.414365	estimación DDA:-->	-57.182230	estimación DDA:-->	27.348066	estimación DDA:-->	28.342541
etiqueta DDA: -->	89.000000	etiqueta DDA: -->	30.000000	etiqueta DDA: -->	-29.000000	etiqueta DDA: -->	-38.000000	etiqueta DDA: -->	-67.000000	etiqueta DDA: -->	52.000000
estimación DDA:-->	86.022099	estimación DDA:-->	28.342541	estimación DDA:-->	-29.337017	estimación DDA:-->	-38.287293	estimación DDA:-->	-66.132597	estimación DDA:-->	51.215470
etiqueta DDA: -->	49.000000	etiqueta DDA: -->	-79.000000	etiqueta DDA: -->	20.000000	etiqueta DDA: -->	14.000000	etiqueta DDA: -->	-56.000000	etiqueta DDA: -->	-11.000000
estimación DDA:-->	48.232044	estimación DDA:-->	-79.060773	estimación DDA:-->	16.408840	estimación DDA:-->	12.430939	estimación DDA:-->	-55.193370	estimación DDA:-->	-9.447514
etiqueta DDA: -->	14.000000	etiqueta DDA: -->	-64.000000	etiqueta DDA: -->	-21.000000	etiqueta DDA: -->	-60.000000	etiqueta DDA: -->	22.000000	etiqueta DDA: -->	-11.000000
estimación DDA:-->	12.430939	estimación DDA:-->	-64.143646	estimación DDA:-->	-19.392265	estimación DDA:-->	-60.165746	estimación DDA:-->	20.386740	estimación DDA:-->	-10.441989
etiqueta DDA: -->	85.000000	etiqueta DDA: -->	-58.000000	etiqueta DDA: -->	20.000000	etiqueta DDA: -->	35.000000	etiqueta DDA: -->	-49.000000	etiqueta DDA: -->	16.000000
estimación DDA:-->	82.044199	estimación DDA:-->	-58.176796	estimación DDA:-->	18.397790	estimación DDA:-->	33.314917	estimación DDA:-->	-49.226519	estimación DDA:-->	14.198990
etiqueta DDA: -->	22.000000	etiqueta DDA: -->	-45.000000	etiqueta DDA: -->	-8.000000	etiqueta DDA: -->	76.000000	etiqueta DDA: -->	-1.000000	etiqueta DDA: -->	41.000000
estimación DDA:-->	20.386740	estimación DDA:-->	-45.248619	estimación DDA:-->	-8.453039	estimación DDA:-->	70.110497	estimación DDA:-->	-1.491713	estimación DDA:-->	39.281768
etiqueta DDA: -->	37.000000	etiqueta DDA: -->	61.000000	etiqueta DDA: -->	75.000000	etiqueta DDA: -->	73.000000	etiqueta DDA: -->	-44.000000	etiqueta DDA: -->	-48.000000
estimación DDA:-->	35.303867	estimación DDA:-->	60.165746	estimación DDA:-->	73.093923	estimación DDA:-->	71.104972	estimación DDA:-->	-43.259669	estimación DDA:-->	-45.248619
etiqueta DDA: -->	-74.000000	etiqueta DDA: -->	-30.000000	etiqueta DDA: -->	56.000000	etiqueta DDA: -->	70.000000	etiqueta DDA: -->	-48.000000	etiqueta DDA: -->	60.000000
estimación DDA:-->	-73.093923	estimación DDA:-->	-29.337017	estimación DDA:-->	55.193370	estimación DDA:-->	69.116022	estimación DDA:-->	-47.237569	estimación DDA:-->	59.171271
etiqueta DDA: -->	9.000000	etiqueta DDA: -->	73.000000	etiqueta DDA: -->	38.000000	etiqueta DDA: -->	20.000000	etiqueta DDA: -->	-71.000000	etiqueta DDA: -->	40.000000
estimación DDA:-->	7.458564	estimación DDA:-->	71.104972	estimación DDA:-->	36.298343	estimación DDA:-->	18.397790	estimación DDA:-->	-71.104972	estimación DDA:-->	37.292818
etiqueta DDA: -->	37.000000	etiqueta DDA: -->	67.000000	etiqueta DDA: -->	-57.000000	etiqueta DDA: -->	-43.000000	etiqueta DDA: -->	45.000000	etiqueta DDA: -->	89.000000
estimación DDA:-->	34.309392	estimación DDA:-->	65.138122	estimación DDA:-->	-57.182230	estimación DDA:-->	-43.259669	estimación DDA:-->	42.265193	estimación DDA:-->	90.000000
etiqueta DDA: -->	6.000000	etiqueta DDA: -->	35.000000	etiqueta DDA: -->	7.000000	etiqueta DDA: -->	47.000000	etiqueta DDA: -->	44.000000	etiqueta DDA: -->	87.000000
estimación DDA:-->	4.475138	estimación DDA:-->	33.314917	estimación DDA:-->	6.464088	estimación DDA:-->	45.248619	estimación DDA:-->	42.265193	estimación DDA:-->	82.044199
etiqueta DDA: -->	-28.000000	etiqueta DDA: -->	63.000000	etiqueta DDA: -->	47.000000	etiqueta DDA: -->	70.000000	etiqueta DDA: -->	-30.000000	etiqueta DDA: -->	78.000000
estimación DDA:-->	-27.348066	estimación DDA:-->	61.160221	estimación DDA:-->	45.248619	estimación DDA:-->	69.116022	estimación DDA:-->	-28.342541	estimación DDA:-->	76.077348
etiqueta DDA: -->	29.000000	etiqueta DDA: -->	-88.000000	etiqueta DDA: -->	35.000000	etiqueta DDA: -->	-46.000000	etiqueta DDA: -->	75.000000	etiqueta DDA: -->	-37.000000
estimación DDA:-->	25.359116	estimación DDA:-->	-88.011050	estimación DDA:-->	33.314917	estimación DDA:-->	-45.248619	estimación DDA:-->	72.099448	estimación DDA:-->	-37.292818
etiqueta DDA: -->	4.000000	etiqueta DDA: -->	-63.000000	etiqueta DDA: -->	70.000000	etiqueta DDA: -->	23.000000	etiqueta DDA: -->	-68.000000	etiqueta DDA: -->	33.000000
estimación DDA:-->	3.480663	estimación DDA:-->	-63.149171	estimación DDA:-->	69.116022	estimación DDA:-->	21.381215	estimación DDA:-->	-68.121547	estimación DDA:-->	31.325967
etiqueta DDA: -->	10.000000	etiqueta DDA: -->	70.000000	etiqueta DDA: -->	59.000000	etiqueta DDA: -->	74.000000	etiqueta DDA: -->	-52.000000	etiqueta DDA: -->	43.000000
estimación DDA:-->	8.453039	estimación DDA:-->	69.116022	estimación DDA:-->	57.182230	estimación DDA:-->	72.099448	estimación DDA:-->	-51.215470	estimación DDA:-->	41.270718
etiqueta DDA: -->	2.000000	etiqueta DDA: -->	-18.000000	etiqueta DDA: -->	17.000000	etiqueta DDA: -->	72.000000	etiqueta DDA: -->	-61.000000	etiqueta DDA: -->	3.000000
estimación DDA:-->	1.491713	estimación DDA:-->	-17.403315	estimación DDA:-->	15.414365	estimación DDA:-->	71.104972	estimación DDA:-->	-79.060773	estimación DDA:-->	2.486188
etiqueta DDA: -->	64.000000	etiqueta DDA: -->	-8.000000	etiqueta DDA: -->	-17.000000	etiqueta DDA: -->	43.000000	etiqueta DDA: -->	-9.000000	etiqueta DDA: -->	-20.000000
estimación DDA:-->	63.149171	estimación DDA:-->	-6.464088	estimación DDA:-->	-13.425414	estimación DDA:-->	41.270718	estimación DDA:-->	-8.453039	estimación DDA:-->	-19.392265
etiqueta DDA: -->	10.000000	etiqueta DDA: -->	61.000000	etiqueta DDA: -->	-4.000000	etiqueta DDA: -->	42.000000	etiqueta DDA: -->	-77.000000	etiqueta DDA: -->	65.000000
estimación DDA:-->	7.458564	estimación DDA:-->	60.165746	estimación DDA:-->	-4.475138	estimación DDA:-->	41.270718	estimación DDA:-->	-77.071823	estimación DDA:-->	63.149171
etiqueta DDA: -->	0.000000	etiqueta DDA: -->	39.000000	etiqueta DDA: -->	81.000000	etiqueta DDA: -->	-49.000000	etiqueta DDA: -->	11.000000	etiqueta DDA: -->	51.000000
estimación DDA:-->	-0.497238	estimación DDA:-->	38.287293	estimación DDA:-->	79.060773	estimación DDA:-->	-48.232044	estimación DDA:-->	10.441989	estimación DDA:-->	49.226519
etiqueta DDA: -->	6.000000	etiqueta DDA: -->	-24.000000	etiqueta DDA: -->	79.000000	etiqueta DDA: -->	-21.000000	etiqueta DDA: -->	-71.000000	etiqueta DDA: -->	-41.000000
estimación DDA:-->	3.480663	estimación DDA:-->	-23.370166	estimación DDA:-->	77.071823	estimación DDA:-->	-19.392265	estimación DDA:-->	-70.110497	estimación DDA:-->	-41.270718
etiqueta DDA: -->	-36.000000	etiqueta DDA: -->	-14.000000	etiqueta DDA: -->	44.000000	etiqueta DDA: -->	51.000000	etiqueta DDA: -->	-37.000000	etiqueta DDA: -->	-83.000000
estimación DDA:-->	-35.303867	estimación DDA:-->	-13.425414	estimación DDA:-->	43.259669	estimación DDA:-->	49.226519	estimación DDA:-->	-36.298343	estimación DDA:-->	-82.044199
etiqueta DDA: -->	-86.000000	etiqueta DDA: -->	13.000000	etiqueta DDA: -->	19.000000	etiqueta DDA: -->	-73.000000	etiqueta DDA: -->	-2.000000	etiqueta DDA: -->	70.000000
estimación DDA:-->	-82.044199	estimación DDA:-->	11.436464	estimación DDA:-->	17.403315	estimación DDA:-->	-72.099448	estimación DDA:-->	-2.486188	estimación DDA:-->	69.116022
etiqueta DDA: -->	20.000000	etiqueta DDA: -->	55.000000	etiqueta DDA: -->	28.000000	etiqueta DDA: -->	82.000000	etiqueta DDA: -->	74.000000	etiqueta DDA: -->	58.000000
estimación DDA:-->	15.414365	estimación DDA:-->	54.198895	estimación DDA:-->	24.364641	estimación DDA:-->	81.049724	estimación DDA:-->	72.099448	estimación DDA:-->	57.182230
etiqueta DDA: -->	86.000000	etiqueta DDA: -->	49.000000	etiqueta DDA: -->	50.000000	etiqueta DDA: -->	73.000000	etiqueta DDA: -->	79.000000	etiqueta DDA: -->	-14.000000
estimación DDA:-->	85.027624	estimación DDA:-->	47.237569	estimación DDA:-->	48.232044	estimación DDA:-->	72.099448	estimación DDA:-->	77.071823	estimación DDA:-->	-13.425414
etiqueta DDA: -->	26.000000	etiqueta DDA: -->	-60.000000	etiqueta DDA: -->	10.000000	etiqueta DDA: -->	-68.000000	etiqueta DDA: -->	-50.000000	etiqueta DDA: -->	-43.000000
estimación DDA:-->	24.364641	estimación DDA:-->	-60.165746	estimación DDA:-->	8.453039	estimación DDA:-->	-68.121547	estimación DDA:-->	-50.220994	estimación DDA:-->	-43.259669
etiqueta DDA: -->	-30.000000	etiqueta DDA: -->	-73.000000	etiqueta DDA: -->	-26.000000	etiqueta DDA: -->	33.000000	etiqueta DDA: -->	32.000000	etiqueta DDA: -->	70.000000
estimación DDA:-->	-29.337017	estimación DDA:-->	-73.093923	estimación DDA:-->	-25.359116	estimación DDA:-->	27.348066	estimación DDA:-->	31.325967	estimación DDA:-->	69.116022
etiqueta DDA: -->	31.000000	etiqueta DDA: -->	-23.000000	etiqueta DDA: -->	-40.000000	etiqueta DDA: -->	57.000000	etiqueta DDA: -->	37.000000	etiqueta DDA: -->	58.000000
estimación DDA:-->	25.359116	estimación DDA:-->	-22.375691	estimación DDA:-->	-39.281768	estimación DDA:-->	56.187845	estimación DDA:-->	35.303867	estimación DDA:-->	57.182230
etiqueta DDA: -->	60.000000	etiqueta DDA: -->	54.000000	etiqueta DDA: -->	-64.000000	etiqueta DDA: -->	5.000000	etiqueta DDA: -->	-80.000000	etiqueta DDA: -->	-38.000000
estimación DDA:-->	58.176796	estimación DDA:-->	53.204420	estimación DDA:-->	-64.143646	estimación DDA:-->	3.480663	estimación DDA:-->	-79.060773	estimación DDA:-->	-37.292818
etiqueta DDA: -->	40.000000	etiqueta DDA: -->	60.000000	etiqueta DDA: -->	84.000000	etiqueta DDA: -->	-78.000000	etiqueta DDA: -->	9.000000	etiqueta DDA: -->	-78.000000
estimación DDA:-->	39.281768	estimación DDA:-->	58.176796	estimación DDA:-->	83.038674	estimación DDA:-->	-77.				

estimación DDA:-->	322.209945	estimación DDA:-->	294.364641	estimación DDA:-->	344.088398	estimación DDA:-->	212.817680	estimación DDA:-->	15.911602	estimación DDA:-->	3.977901
etiqueta DDA: -->	6.000000	etiqueta DDA: -->	173.000000	etiqueta DDA: -->	84.000000	etiqueta DDA: -->	253.000000	etiqueta DDA: -->	332.000000	etiqueta DDA: -->	341.000000
estimación DDA:-->	5.966851	estimación DDA:-->	173.038674	estimación DDA:-->	83.535912	estimación DDA:-->	254.585635	estimación DDA:-->	332.154696	estimación DDA:-->	344.088398
etiqueta DDA: -->	270.000000	etiqueta DDA: -->	71.000000	etiqueta DDA: -->	21.000000	etiqueta DDA: -->	180.000000	etiqueta DDA: -->	4.000000	etiqueta DDA: -->	20.000000
estimación DDA:-->	270.497238	estimación DDA:-->	73.591160	estimación DDA:-->	21.878453	estimación DDA:-->	179.005525	estimación DDA:-->	3.977901	estimación DDA:-->	17.900552
etiqueta DDA: -->	303.000000	etiqueta DDA: -->	105.000000	etiqueta DDA: -->	121.000000	etiqueta DDA: -->	173.000000	etiqueta DDA: -->	71.000000	etiqueta DDA: -->	88.000000
estimación DDA:-->	302.320442	estimación DDA:-->	105.414365	estimación DDA:-->	121.325967	estimación DDA:-->	173.038674	estimación DDA:-->	71.602210	estimación DDA:-->	87.513812
etiqueta DDA: -->	261.000000	etiqueta DDA: -->	126.000000	etiqueta DDA: -->	151.000000	etiqueta DDA: -->	85.000000	etiqueta DDA: -->	97.000000	etiqueta DDA: -->	335.000000
estimación DDA:-->	260.552486	estimación DDA:-->	125.303867	estimación DDA:-->	151.160221	estimación DDA:-->	85.524862	estimación DDA:-->	97.458564	estimación DDA:-->	336.132597
etiqueta DDA: -->	130.000000	etiqueta DDA: -->	136.000000	etiqueta DDA: -->	194.000000	etiqueta DDA: -->	71.000000	etiqueta DDA: -->	309.000000	etiqueta DDA: -->	74.000000
estimación DDA:-->	129.281768	estimación DDA:-->	135.248619	estimación DDA:-->	192.375691	estimación DDA:-->	73.591160	estimación DDA:-->	306.298343	estimación DDA:-->	73.591160
etiqueta DDA: -->	199.000000	etiqueta DDA: -->	99.000000	etiqueta DDA: -->	282.000000	etiqueta DDA: -->	232.000000	etiqueta DDA: -->	289.000000	etiqueta DDA: -->	237.000000
estimación DDA:-->	196.906077	estimación DDA:-->	99.447514	estimación DDA:-->	282.430939	estimación DDA:-->	232.707182	estimación DDA:-->	288.397790	estimación DDA:-->	236.685083
etiqueta DDA: -->	181.000000	etiqueta DDA: -->	169.000000	etiqueta DDA: -->	25.000000	etiqueta DDA: -->	197.000000	etiqueta DDA: -->	261.000000	etiqueta DDA: -->	196.000000
estimación DDA:-->	179.005525	estimación DDA:-->	169.060773	estimación DDA:-->	23.867403	estimación DDA:-->	198.895028	estimación DDA:-->	260.552486	estimación DDA:-->	194.917127
etiqueta DDA: -->	335.000000	etiqueta DDA: -->	276.000000	etiqueta DDA: -->	153.000000	etiqueta DDA: -->	116.000000	etiqueta DDA: -->	271.000000	etiqueta DDA: -->	314.000000
estimación DDA:-->	336.132597	estimación DDA:-->	276.464088	estimación DDA:-->	155.138122	estimación DDA:-->	113.370166	estimación DDA:-->	270.497238	estimación DDA:-->	314.254144
etiqueta DDA: -->	207.000000	etiqueta DDA: -->	186.000000	etiqueta DDA: -->	72.000000	etiqueta DDA: -->	66.000000	etiqueta DDA: -->	209.000000	etiqueta DDA: -->	348.000000
estimación DDA:-->	204.861878	estimación DDA:-->	184.972376	estimación DDA:-->	73.591160	estimación DDA:-->	67.624309	estimación DDA:-->	208.839779	estimación DDA:-->	348.066298
etiqueta DDA: -->	5.000000	etiqueta DDA: -->	189.000000	etiqueta DDA: -->	103.000000	etiqueta DDA: -->	294.000000	etiqueta DDA: -->	19.000000	etiqueta DDA: -->	313.000000
estimación DDA:-->	3.977901	estimación DDA:-->	188.950276	estimación DDA:-->	101.436464	estimación DDA:-->	290.386740	estimación DDA:-->	17.900552	estimación DDA:-->	312.265193
etiqueta DDA: -->	123.000000	etiqueta DDA: -->	23.000000	etiqueta DDA: -->	322.000000	etiqueta DDA: -->	209.000000	etiqueta DDA: -->	24.000000	etiqueta DDA: -->	275.000000
estimación DDA:-->	123.314917	estimación DDA:-->	27.845304	estimación DDA:-->	322.209945	estimación DDA:-->	208.839779	estimación DDA:-->	21.878453	estimación DDA:-->	272.486188
etiqueta DDA: -->	184.000000	etiqueta DDA: -->	69.613260	etiqueta DDA: -->	116.000000	etiqueta DDA: -->	45.000000	etiqueta DDA: -->	131.000000	etiqueta DDA: -->	178.000000
estimación DDA:-->	182.983425	estimación DDA:-->	69.613260	estimación DDA:-->	113.370166	estimación DDA:-->	45.745856	estimación DDA:-->	131.270718	estimación DDA:-->	171.049724
etiqueta DDA: -->	310.000000	etiqueta DDA: -->	313.000000	etiqueta DDA: -->	73.000000	etiqueta DDA: -->	141.000000	etiqueta DDA: -->	52.000000	etiqueta DDA: -->	196.000000
estimación DDA:-->	306.298343	estimación DDA:-->	310.276243	estimación DDA:-->	73.591160	estimación DDA:-->	141.215470	estimación DDA:-->	53.701657	estimación DDA:-->	194.917127
etiqueta DDA: -->	321.000000	etiqueta DDA: -->	183.000000	etiqueta DDA: -->	189.000000	etiqueta DDA: -->	232.000000	etiqueta DDA: -->	346.000000	etiqueta DDA: -->	274.475138
estimación DDA:-->	322.209945	estimación DDA:-->	182.983425	estimación DDA:-->	188.950276	estimación DDA:-->	232.707182	estimación DDA:-->	348.066298	estimación DDA:-->	274.475138
etiqueta DDA: -->	71.000000	etiqueta DDA: -->	322.000000	etiqueta DDA: -->	9.000000	etiqueta DDA: -->	192.000000	etiqueta DDA: -->	199.000000	etiqueta DDA: -->	33.000000
estimación DDA:-->	69.613260	estimación DDA:-->	322.209945	estimación DDA:-->	7.955801	estimación DDA:-->	190.939227	estimación DDA:-->	200.883978	estimación DDA:-->	35.801105
etiqueta DDA: -->	259.000000	etiqueta DDA: -->	135.000000	etiqueta DDA: -->	88.000000	etiqueta DDA: -->	333.000000	etiqueta DDA: -->	359.000000	etiqueta DDA: -->	256.000000
estimación DDA:-->	262.541436	estimación DDA:-->	135.248619	estimación DDA:-->	89.502762	estimación DDA:-->	332.154696	estimación DDA:-->	358.011050	estimación DDA:-->	258.563536
etiqueta DDA: -->	146.000000	etiqueta DDA: -->	139.000000	etiqueta DDA: -->	138.000000	etiqueta DDA: -->	228.000000	etiqueta DDA: -->	289.000000	etiqueta DDA: -->	346.000000
estimación DDA:-->	145.193370	estimación DDA:-->	139.226519	estimación DDA:-->	137.237569	estimación DDA:-->	228.729282	estimación DDA:-->	288.397790	estimación DDA:-->	346.077348
etiqueta DDA: -->	257.000000	etiqueta DDA: -->	265.000000	etiqueta DDA: -->	301.000000	etiqueta DDA: -->	236.000000	etiqueta DDA: -->	6.000000	etiqueta DDA: -->	156.000000
estimación DDA:-->	256.574586	estimación DDA:-->	264.530387	estimación DDA:-->	300.331492	estimación DDA:-->	234.696133	estimación DDA:-->	7.955801	estimación DDA:-->	157.127072
etiqueta DDA: -->	108.000000	etiqueta DDA: -->	263.000000	etiqueta DDA: -->	245.000000	etiqueta DDA: -->	76.000000	etiqueta DDA: -->	264.000000	etiqueta DDA: -->	295.000000
estimación DDA:-->	107.403315	estimación DDA:-->	264.530387	estimación DDA:-->	244.640864	estimación DDA:-->	77.569061	estimación DDA:-->	264.530387	estimación DDA:-->	294.364641
etiqueta DDA: -->	309.000000	etiqueta DDA: -->	327.000000	etiqueta DDA: -->	137.000000	etiqueta DDA: -->	304.000000	etiqueta DDA: -->	316.000000	etiqueta DDA: -->	44.000000
estimación DDA:-->	308.287293	estimación DDA:-->	324.198895	estimación DDA:-->	135.248619	estimación DDA:-->	304.309392	estimación DDA:-->	316.243094	estimación DDA:-->	43.756906
etiqueta DDA: -->	290.000000	etiqueta DDA: -->	129.000000	etiqueta DDA: -->	198.000000	etiqueta DDA: -->	227.000000	etiqueta DDA: -->	255.000000	etiqueta DDA: -->	104.000000
estimación DDA:-->	288.397790	estimación DDA:-->	127.292818	estimación DDA:-->	196.906077	estimación DDA:-->	226.740331	estimación DDA:-->	254.585635	estimación DDA:-->	101.436464
etiqueta DDA: -->	86.000000	etiqueta DDA: -->	334.000000	etiqueta DDA: -->	317.000000	etiqueta DDA: -->	43.000000	etiqueta DDA: -->	317.000000	etiqueta DDA: -->	334.000000
estimación DDA:-->	85.524862	estimación DDA:-->	334.143646	estimación DDA:-->	316.243094	estimación DDA:-->	43.756906	estimación DDA:-->	316.243094	estimación DDA:-->	334.143646
etiqueta DDA: -->	21.000000	etiqueta DDA: -->	20.000000	etiqueta DDA: -->	257.000000	etiqueta DDA: -->	84.000000	etiqueta DDA: -->	345.000000	etiqueta DDA: -->	284.000000
estimación DDA:-->	19.889503	estimación DDA:-->	17.900552	estimación DDA:-->	256.574586	estimación DDA:-->	83.535912	estimación DDA:-->	346.077348	estimación DDA:-->	284.419890
etiqueta DDA: -->	97.000000	etiqueta DDA: -->	235.000000	etiqueta DDA: -->	300.000000	etiqueta DDA: -->	352.000000	etiqueta DDA: -->	233.000000	etiqueta DDA: -->	29.000000
estimación DDA:-->	95.469613	estimación DDA:-->	232.707182	estimación DDA:-->	300.331492	estimación DDA:-->	352.044199	estimación DDA:-->	232.707182	estimación DDA:-->	27.845304
etiqueta DDA: -->	143.000000	etiqueta DDA: -->	75.000000	etiqueta DDA: -->	227.000000	etiqueta DDA: -->	240.000000	etiqueta DDA: -->	287.000000	etiqueta DDA: -->	229.000000
estimación DDA:-->	143.204420	estimación DDA:-->	75.580110	estimación DDA:-->	226.740331	estimación DDA:-->	246.629834	estimación DDA:-->	284.419890	estimación DDA:-->	228.729282
etiqueta DDA: -->	271.000000	etiqueta DDA: -->	234.000000	etiqueta DDA: -->	179.000000	etiqueta DDA: -->	123.000000	etiqueta DDA: -->	5.000000	etiqueta DDA: -->	142.000000
estimación DDA:-->	270.497238	estimación DDA:-->	234.696133	estimación DDA:-->	179.005525	estimación DDA:-->	123.303867	estimación DDA:-->	3.977901	estimación DDA:-->	141.215470
etiqueta DDA: -->	344.000000	etiqueta DDA: -->	336.000000	etiqueta DDA: -->	174.000000	etiqueta DDA: -->	201.000000	etiqueta DDA: -->	70.000000	etiqueta DDA: -->	92.000000
estimación DDA:-->	346.077348	estimación DDA:-->	336.132597	estimación DDA:-->	175.027624	estimación DDA:-->	200.883978	estimación DDA:-->	71.602210	estimación DDA:-->	91.491713
etiqueta DDA: -->	189.000000	etiqueta DDA: -->	359.000000	etiqueta DDA: -->	345.000000	etiqueta DDA: -->	359.000000	etiqueta DDA: -->	256.000000	etiqueta DDA: -->	100.000000
estimación DDA:-->	188.950276	estimación DDA:-->	358.011050	estimación DDA:-->	346.077348	estimación DDA:-->	7.955801	estimación DDA:-->	256.574586	estimación DDA:-->	99.447514
etiqueta DDA: -->	63.000000	etiqueta DDA: -->	147.000000	etiqueta DDA: -->	165.000000	etiqueta DDA: -->	352.000000	etiqueta DDA: -->	241.000000	etiqueta DDA: -->	231.000000
estimación DDA:-->	65.635359	estimación DDA:-->	147.182320	estimación DDA:-->	165.082873	estimación DDA:-->	352.044199	estimación DDA:-->	240.662983	estimación DDA:-->	232.707182
etiqueta DDA: -->	339.000000	etiqueta DDA: -->	121.000000	etiqueta DDA: -->	251.000000	etiqueta DDA: -->	0.000000	etiqueta DDA: -->	161.000000	etiqueta DDA: -->	147.000000
estimación DDA:-->	340.110497	estimación DDA:-->	121.325967	estimación DDA:-->	251.005525	estimación DDA:-->	0.000000	estimación DDA:-->	161.000000	estimación DDA:-->	145.193370
etiqueta DDA: -->	286.000000	etiqueta DDA: -->	270.000000	etiqueta DDA: -->	57.000000	etiqueta DDA: -->	272.000000	etiqueta DDA: -->	273.000000	etiqueta DDA: -->	268.000000
estimación DDA:-->	284.419890	estimación DDA:-->	270.497238	estimación DDA:-->	57.679558	estimación DDA:-->	270.497238	estimación DDA:-->	272.486188	estimación DDA:-->	268.508287
etiqueta DDA: -->	219.000000	etiqueta DDA: -->	224.000000	etiqueta DDA: -->	103.000000	etiqueta DDA: -->	344.000000	etiqueta DDA: -->	359.000000	etiqueta DDA: -->	280.000000
estimación DDA:-->	218.784530	estimación DDA:-->	224.751381	estimación DDA:-->	101.436464	estimación DDA:-->	342.099448	estimación DDA:-->	0.000000	estimación DDA:-->	278.453039
etiqueta DDA: -->	165.000000	etiqueta DDA: -->	337.000000	etiqueta DDA: -->	80.000000	etiqueta DDA: -->	17.000000	etiqueta DDA: -->	16.000000	etiqueta DDA: -->	21.000000
estimación DDA:-->	165.082873	estimación DDA:-->	336.132597	estimación DDA:-->	81.546961	estimación DDA:-->	15.911602	estimación DDA:-->	13.922652	estimación DDA:-->	21.878453
etiqueta DDA: -->	44.000000	etiqueta DDA: -->	9.000000	etiqueta DDA: -->	95.000000	etiqueta DDA: -->	206.000000	etiqueta DDA: -->	349.000000	etiqueta DDA: -->	30.000000
estimación DDA:-->	43.756906	estimación DDA:-->	7.955801	estimación DDA:-->	95.469613	estimación DDA:-->	206.879298	estimación DDA:-->	350.055249	estimación DDA:-->	29.834254
etiqueta DDA: -->	147.000000	etiqueta DDA: -->	32.000000	etiqueta DDA: -->	51.000000	etiqueta DDA: -->	48.000000	etiqueta DDA: -->	170.000000	etiqueta DDA: -->	237.000000
estimación DDA:-->	147.182320	estimación DDA:-->	29.000000	estimación DDA:-->	51.712707	estimación DDA:-->	47.734807	estimación DDA:-->	171.049724	estimación DDA:-->	238.674033
etiqueta DDA: -->	3.000000	etiqueta DDA: -->	241.000000	etiqueta DDA: -->	173.000000	etiqueta DDA: -->	109.000000	etiqueta DDA: -->	187.000000	etiqueta DDA: -->	336.000

Acrónimos

Adam Adaptive Movement Estimation algorithm.

AI Artificial Intelligence.

API Application Programming Interface.

ARM Advanced RISC Machine.

AWGN Additive White Gaussian Noise.

BF Beamforming.

Bf Beamformer.

BP Beam Pattern.

CNN Convolutional Neural Network.

DFT Discrete Fourier Transform.

DOA Direction of Arrival.

DSP Digital Signal Processor.

ESPIRIT Estimation of Signal Parameters via Rotational Invariant Techniques.

FIR Finit Impulse Response.

GCC Generalized Cross-Correlation.

GCC-PHAT Generalised Cross-Correlation with Phase Transform.

GSC Generalized Sidelobe Canceller.

HL Hidden Layer.

ISM Image Source Method.

LCMV Linearly Constrained Minimum Variance.

LMS Least Mean Square.

MCAF Multi-Channel Adaptive Filter.

ML Machine Learning.

MLP Multilayer Perceptron.

MSC Multi-Sidelobes Canceller.

MSE Mean Square Error.

MUSIC Multiple Signal Classification.

MVDR Minimum Variance Distortionless Response.

NLMS Normalized Least Mean Square.

NN Neural Network.

PCA Principal Component Analysis.

PCM Pulse Code Modulation.

RIR Room Impulse Response.

RLS Recursive Least Square.

RMSProp Root Mean Square Propagation.

RNN Recurrent Neural Network.

SDL Sample Delay Lines.

SGD Stochastic Gradient Descent.

SNR Signal to Noise Ratio.

SVD Singular Value Decomposition.

TDOA Time Delay of Arrival.

UCA Uniform Circular Array.

ULA Uniform Linear Array.

VAD Voice Activity Detection.

VLSI Very Large-Scale Integration.

WAV Waveform Audio File Format.

Bibliografía

- [1] Allen B. and Ghavami M., “Adaptive Array Systems, Fundamentals and Applications”. John Wiley & Sons, Ltd, Chichester, UK, 2005.
- [2] Brandstein M.S. and Ward D. (Eds) “Microphone Arrays: Signal Processing Techniques and Applications”. Springer-Verlag, Berlin, Germany. 2001.
- [3] Fourikis N., “Advanced Array Systems, Applications and RF Technologies”. Academic Press, London, UK. 2000.
- [4] Haykin S., “Adaptive Filter Theory”, 3rd Edn. Prentice-Hall, Englewood Cliffs, NJ, USA. 1996.
- [5] Hudson J.E., “Adaptive Array Principles”, IEE Electromagnetic Waves. The Institution of Electrical Engineers, London, UK. 1981.
- [6] Johnson D.H. and Dudgeon D.E., “Array Signal Processing: Concepts and Techniques”, Signal Processing Series. Prentice-Hall, Englewood Cliffs, NJ, USA. 1993.
- [7] Monzingo R.A. and Miller T.W., “Introduction to Adaptive Arrays”. SciTech Publishing Inc., Raleigh, NC, USA. 2004.
- [8] Van Trees H.L. “Optimum Array Processing, Part IV of Detection, Estimation and Modulation Theory”. John Wiley & Sons, Inc., New York, NY, USA. 2002.
- [9] L. Wei and S. Weiss, "Wideband Beamforming Concepts and Techniques". John Wiley & Sons, Ltd, 2010.
- [10] K. M. Buckley and L. Griffiths, "Eigenstructure based broadband source location estimation". Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 11, pp. 1869– 1872, Los Angeles, CA, USA. 1986.
- [11] K. M. Buckley, “Spatial/spectral filtering with linearly-constrained minimum variance beamformers”, IEEE Trans. On ASSP, Vol. ASSP-35, pp. 249-266, Mar. 1987.
- [12] B. D. Van Veen and K. M. Buckley, “Beamforming: A versatile approach to spatial filtering,” IEEE ASSP Magazine, pp. 4-24, April. 1988.
- [13] B. Widrow, P. E. Mantez, L. J. Griffiths, and B. B. Goode. “Adaptive antenna systems”. Proceedings of the IEEE, 55(12):2143-2159, December 1967.
- [14] B. Widrow and SD Stearns, "Adaptive Signal Processing". Prentice-Hall, Englewood Cliffs, NJ, USA. 1985
- [15] O. L. Frost. “An algorithm for linearly constrained adaptive array processing”. IEEE Proceedings, 60:926-935, August 1972.
- [16] L. J. Griffiths and C. W. Jim, “An alternative approach to linearly constrained adaptive beamforming”, IEEE Trans. On AP, Vol. AP-30, pp. 27-34, Jan 1982.
- [17] S. Haykin, “Neural Network and Learning Machines”, 3th ed., Prentice- Hall, 2009.
- [18] Goodfellow, I., Bengio, Y. and Courville, A.: “Deep Learning”, <http://www.deeplearningbook.org>, (MIT Press, 2016).

- [19] Ferguson, E. L., Williams, S. B. and Jin, C. T.: “Sound Source Localization in a Multipath Environment Using Convolutional Neural Networks”, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2386–2390, doi: 10.1109/ICASSP.2018.8462024.
- [20] Qinglong, L., Zhang, X. and Li, H.: “Online Direction of Arrival Estimation Based on Deep Learning”, in: doi: 10.1109/ICASSP.2018.8461386.
- [21] He, W., Motlicek, P. and Odobez, J.-M.: “Deep Neural Networks for Multiple Speaker Detection and Localization”, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 74–79, doi: 10.1109/ICRA.2018.8461267.
- [22] Adavanne, S., Politis, A. and Virtanen, T.: “Direction of Arrival Estimation for Multiple Sound Sources Using Convolutional Recurrent Neural Network”, in: 2018 26th European Signal Processing Conference (EUSIPCO), pp. 1462–1466, doi: 10.23919/EUSIPCO.2018.8553182.
- [23] S. Chakrabarty, EAP. Habets, “Multi-speaker DOA estimation using deep convolutional networks trained with noise signals”, IEEE Journal of Selected Topics in Signal Processing. Volume: 13, Issue: 1, March 2019.
- [24] L. Perotin, R. Serizel, E. Vincent, “CRNN-based multiple DOA estimation using ambisonics acoustic intensity features”, IEEE Journal of Selected Topics in Signal Processing. Volume: 13, Issue: 1, March 2019.
- [25] S. Adavanne, A. Politis, J. Nikunen, “Sound event localization and detection of overlapping sources using convolutional recurrent neural networks”, IEEE Journal of Selected Topics in Signal Processing. Volume: 13, Issue: 1, March 2019.
- [26] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, “State-of-the-art in artificial neural network applications: A survey”, Heliyon, Volume 4, Issue 11, 2018, e00938, ISSN 2405-8440, <https://doi.org/10.1016/j.heliyon.2018.e00938>. (<https://www.sciencedirect.com/science/article/pii/S2405844018332067>).
- [27] D.E. Rumelhart, G. E. Hinton, and R.J. Williams. “Learning internal representations by error propagation,” in D.E. Rumelhart and J.L. McClelland, eds., vol 1, Chapter 8, Cambridge, MA: MIT Press. 1986
- [28] Y. LeCun, Y. Bengio, 2003. “Convolutional Networks for Images, Speech, and Time Series,” in M.A. Arbib, ed., The Handbook of Brain Theory and Neural Networks, 2d ed., Cambridge, MA: MIT Press.
- [29] K. M. Ahmed and R. J. Evans. “An adaptive array processor with robustness and broadband capabilities.” IEEE Transactions on Antennas and Propagation AP-32(9), 944– 950. 1984
- [30] M. Er and A. Cantoni. “A new set of linear constraints for broad-band time domain element space processors.” IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-34(6), 1376– 1379. 1986
- [31] Chung, P.-J., Viberg, M. and Yu, J.: “DOA Estimation Methods and Algorithms”, in: vol. 3, pp. 599– 650, isbn: 9780124115972, doi: 10.1016/B978-0-12-411597-2.00014-X.
- [32] LeCun, Y., Bengio, Y. and Hinton, G.: ‘Deep Learning’, in: Nature 521 (May 2015), pp. 436–44, doi: 10.1038/nature14539.
- [33] Xiao, X., Zhao, S., Zhong, X., Jones, D. L., Chng, E. S. and Li, H.: “A learning-based approach to direction of arrival estimation in noisy and reverberant environments”, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2814–2818, doi: 10.1109/ICASSP.2015.7178484.
- [34] Xiao, X., Watanabe, S., Erdogan, H. et al.: “Deep beamforming networks for multi-channel speech recognition”, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5745–5749, doi: 10.1109/ICASSP.2016.7472778.
- [35] John Wiseman: Python interface to the Google WebRTC Voice Activity Detector (VAD), Project description, <https://pypi.org/project/webrtcvad/>

- [36] John Wiseman: WebRTCvad source code, 2016, *https : //github.com/wiseman/py – webrtcvad/blob/master/cbits/webrtc/common_audio/vad/vad_core.c* (visitado en Mayo 2023), Github.
- [37] AMI Corpus, License: CC BY 4.0, University of Edinburgh, url: license : <https://creativecommons.org/licenses/by/4.0/legalcode>
- [38] Allen, J. and Berkley, D.: “Image method for efficiently simulating small-room acoustics”, in: The Journal of the Acoustical Society of America 65 (Apr. 1979), pp. 943–950, doi: 10.1121/1.382599.
- [39] Scheibler, R., Bezzam, E. and Dokmanić, I.: “Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms”, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 351–355, doi: 10.1109/ICASSP.2018.8461310.
- [40] Peña, Daniel. (2002). Análisis de Datos Multivariantes.
- [41] Pavlidi, D., Griffin, A., Puigt, M. and Mouchtaris, A.: “Real-Time Multiple Sound Source Localization and Counting Using a Circular Microphone Array”, in: IEEE Transactions on Audio, Speech, and Language Processing 21.10 (2013), pp. 2193–2206, doi: 10.1109/TASL.2013.2272524.
- [42] Blanco Galindo, M., Coleman, P. and Jackson, P.: “Microphone Array Geometries for Horizontal Spatial Audio Object Capture With Beamforming”, in: Journal of the Audio Engineering Society 68 (June 2020), pp. 324–337, doi: 10.17743/jaes.2020.0025.