Universidad Zaragoza
1542

Trabajo Fin de Máster

Análisis y comparación de técnicas de filtrado espacial y frecuencial para una interfaz mediante voz del usuario en una placa de inducción

Analysis and comparison of spatial and frequency filtering techniques for a voice user interface in an induction cooktop

Autor

María Molina Gracia

Directores
David Díaz-Guerra Aparicio
José Ramón Beltrán Blázquez

Escuela de Ingeniería y Arquitectura

2020/2021

Análisis y comparación de técnicas de filtrado espacial y frecuencial para una interfaz mediante voz del usuario en una placa de inducción

Resumen

El filtrado de señales para la reducción de ruido es un tema ampliamente estudiado en el que se han desarrollado numerosos algoritmos de filtrado tanto espacial como frecuencial. En este trabajo se va a abordar el caso concreto de tener como entorno sonoro una placa de inducción para analizar el control de la cocina mediante la voz del usuario. Para conseguir dicho objetivo, se van a implementar diferentes algoritmos de filtrado espacial y frecuencial con dos *arrays* de micrófonos, para proceder seguidamente con un estudio de las prestaciones ofrecidas por todos ellos a la hora de detectar la voz del usuario. Se va a trabajar con una placa con extractor de humos integrado, por lo que se ha trabajado suponiendo que éste es la principal fuente de ruido.

El método empleado será el siguiente. Para el filtrado espacial se utilizarán el algoritmo de *beamforming Delay and Sum* y uno superdirectivo, mientras que para el filtrado frecuencial se usará el filtro de Wiener adaptativo LMS. Los *arrays* empleados son un *array* de geometría rectangular y dimensiones similares a las de la placa de inducción y un *array* de geometría circular y dimensiones similares a las de un Echo Dot de Amazon. Para el estudio de prestaciones se llevarán a cabo unas estimaciones en términos de SNR y se evaluara la capacidad de detección de la palabra clave "ALEXA" que permitiría, una vez detectada, reducir la potencia del extractor para así facilitar el reconocimiento del resto de la orden dada por el usuario.

Analysis and comparison of spatial and frequency filtering techniques for a voice user interface in an induction cooktop

Abstract

Signal filtering for noise reduction is a widely studied subject in which numerous spatial and frequency filtering algorithms have been proposed. In this project, the specific case of having an induction cooktop as the sound environment will be addressed to analyze the control of the induction cooktop through the user's voice. To achieve this goal, several spatial and frequency filtering algorithms will be implemented with two microphone arrays to study the performance offered by all of them for detecting the user's voice. We are going to work with a cooktop with an integrated smoke extractor, so we have worked on the assumption that this is the main source of noise.

The method used will be the following. For the spatial filtering, the Delay and Sum and superdirective beamforming algorithm will be employed, while for the frequency filtering the Wiener adaptive LMS filter will be used. The arrays used are an array with rectangular geometry and dimensions similar to those of an induction cooktop and an array with circular geometry and dimensions similar to those of an Amazon Echo Dot. For the performance study, SNR estimations will be carried out and the ability to detect the keyword "ALEXA" will be evaluated, which would allow, once detected, to reduce the power of the extractor in order to facilitate the recognition of the rest of the commands given by the user.

Tabla de contenido

Índice de figuras

Índice de tablas

Capítulo 1. Introducción y objetivos

El objetivo principal de este proyecto es estudiar y comparar las prestaciones que ofrecería un *array* de micrófonos integrado en el interior de una placa de inducción frente a las de un *array* externo, similar a los utilizados en la mayoría de los altavoces inteligentes, de cara a usar un asistente de voz para controlar la cocina en un entorno de alto ruido.

El proyecto se ha llevado a cabo en el Laboratorio de Audio y Video de la Universidad de Zaragoza en colaboración con el departamento de pre-desarrollo de la empresa BSH Electrodomésticos España S.A., dentro del proyecto de investigación llamado *Análisis de viabilidad de incorporación en una cocina de un sistema de gestión por voz en un entorno de alto ruido*.

El proyecto de investigación nació de la idea de mejorar la experiencia de usuario en una cocina y siguiendo la línea de investigación de establecer una comunicación con una placa de inducción con extractor integrado para controlarla por voz, en la que también se enmarcó mi TFG [1]. Para implementar el control por voz, el grupo BSH ha decidido usar el asistente de voz Alexa, que pertenece a la empresa Amazon, y se han planteado dos posibles soluciones para captar la señal de voz del usuario: la primera solución es emplear un dispositivo externo a la placa de inducción, como un Echo Dot, y la segunda sería colocar un *array* de micrófonos dentro de la placa. Integrar los micrófonos en el interior de la placa de inducción supone una solución más compacta que ofrece mayor comodidad al usuario y un mayor valor añadido al producto de la empresa, pero implica tener los micrófonos en un entorno altamente ruidoso y bajo el cristal de la placa de inducción; aunque, al aumentar considerablemente las dimensiones del *array*, también permitiría obtener mayores beneficios del procesado de señal.

Al trabajar con un entorno de alto ruido, para alcanzar el objetivo de poder controlar la cocina mediante la voz del usuario, se han implementado dos algoritmos de filtrado espacial, también llamado *beamforming*, y un algoritmo de filtrado frecuencial. Todos los algoritmos han sido evaluados de igual modo en los dos *arrays* de micrófonos propuestos.

El *beamforming* es un tipo de filtrado espacial que tiene como principal premisa realzar las señales que provienen de la dirección en la que se encuentra la fuente sonora de interés. El *beamforming* de retardo y suma (*Delay And Sum*) es el más sencillo de todos y ha sido el primer algoritmo desarrollado debido a esta razón. Por otro lado, el *beamforming* superdirectivo trata de conseguir un nivel mayor de directividad y, por tanto, ofrece mejores prestaciones a priori, hipótesis que se ha ido estudiando a lo largo del proyecto. En cuanto al filtrado frecuencial, éste se ha basado en el empleo del filtro de Wiener adaptativo con el algoritmo LMS, ya que su implementación resulta más sencilla que la del filtro de Wiener óptimo. Esto se debe a que con el algoritmo LMS no es necesario estimar explícitamente las estadísticas de las señales ni llevar a cabo inversiones de matrices.

En cuanto a las medidas de nivel de potencia de ruido, estas se han realizado en términos de nivel de presión sonora (*Sound Pressure Level*, SPL), que representa en formato logarítmico el valor de presión eficaz respecto a un nivel de presión de referencia (el mínimo nivel audible) y para poder estudiar más concretamente el desempeño de los algoritmos desarrollados se han empleado medidas de relación señal a ruido (*Signal-to-noise ratio*, SNR), que indica el nivel de señal que hay con respecto al nivel de ruido también en formato logarítmico. Además, también se ha probado a pasar las señales procesadas al reconocedor de voz de Alexa para poder evaluar si las mejoras conseguidas eran suficientes o no para alcanzar el objetivo de controlar por voz una concina.

El primer paso que se ha seguido ha sido programar en Matlab los diferentes algoritmos de filtrado. A continuación, se ha realizado el estudio de prestaciones que se ha dividido en dos partes: primero se ha llevado a cabo una simulación acústica del sistema placa-micrófonos y seguidamente se han implementado los dos *arrays* para obtener medidas experimentales reales de las prestaciones. Se han usado las siguientes herramientas: un miniDSP MCHStreamer, que es una interfaz de audio USB multicanal y multiprotocolo; 16 micrófonos MEMS digitales omnidireccionales; un miniDSP UMA-8, que es un *array* comercial de 7 micrófonos; Audacity, que es un programa de grabación multicanal; Matlab para el desarrollo de los algoritmos y análisis de las señales realizas en las pruebas; una Raspberry Pi en la que se ha instalado el control por voz implementado por Amazon (Alexa) y el programa para análisis y medidas acústicas Room EQ.

El trabajo se estructura de la siguiente forma: tras la introducción, el segundo capítulo desarrolla en profundidad los diferentes conceptos teóricos en los que se basa el estudio realizado, el tercer capítulo explica los materiales y la metodología seguida para llevar a cabo las medidas y las diferentes pruebas con los dos *arrays* así como la simulación acústica, el cuarto capítulo de esta memoria expone los resultados tras la realización de dichas pruebas y medidas y, finalmente, en el último capítulo se encuentran las principales conclusiones y algunas propuestas de trabajo futuro.

Capítulo 2. Conceptos teóricos

En este capítulo se presentan los fundamentos teóricos de las técnicas de filtrado espacial y frecuencial cuyos resultados se compararán en capítulos posteriores.

Comenzamos con la explicación del filtrado espacial, junto con el tipo de filtros que se ha decidido utilizar, y seguidamente procedemos con el filtrado frecuencial de igual modo que con el filtrado espacial. Adicionalmente, se van a utilizar una serie de conceptos que se explican en el ***Anexo A. Conceptos teóricos*** (extraído de la memoria de mi TFG [1]), en concreto, se trata del algoritmo SRP-PHAT de localización de fuentes y una breve explicación de conceptos acústicos como el ruido o las medidas de nivel sonoro, ya que estos conceptos son fundamentales para cualquier tipo de estudio acústico y necesarios para entender las medidas realizadas durante la parte experimental del proyecto.

2.1 Filtrado espacial: *Beamforming*

El objetivo del filtrado espacial es conseguir una mejora de las señales que provienen de una determinada dirección al mismo tiempo que se atenúan las señales provenientes del resto de direcciones [2]. Este tipo de filtros son capaces de mostrar un pico en la dirección de llegada (*direction of arrival*, DOA) de la señal de interés que llega a un *array* mientras se atenúan el resto de las direcciones. Su eficacia estará limitada por las dimensiones del *array*, ya que la resolución del filtro es aproximadamente proporcional a la apertura del *array* (medida en longitudes de onda).

Un tipo de filtrado espacial es el *beamforming*, o conformado de haz, que permite distinguir las propiedades espaciales de una señal deseada u objetivo y el ruido de fondo que pueda haber. De hecho, se puede considerar al *beamforming* como la extensión del periodograma a las señales espaciales.

Dentro de este apartado vamos a analizar dos tipos de *beamformings*, el *Delay And Sum* [3] y el superdirectivo [4], ya que en los siguientes capítulos se diseñarán ambos *beamformers* o conformadores de haz y se evaluarán las prestaciones que ofrecen.

2.1.1 *Delay And Sum*

El *Delay And Sum* es el tipo de *beamforming* más simple de todos. El *beamformer* aplica un desplazamiento en el tiempo a las señales que llegan al *array* para compensar el retardo de propagación con el que dichas señales llegan desde la fuente original hasta cada micrófono. A continuación, las señales se alinean en el tiempo y se suman para formar una única señal de salida.

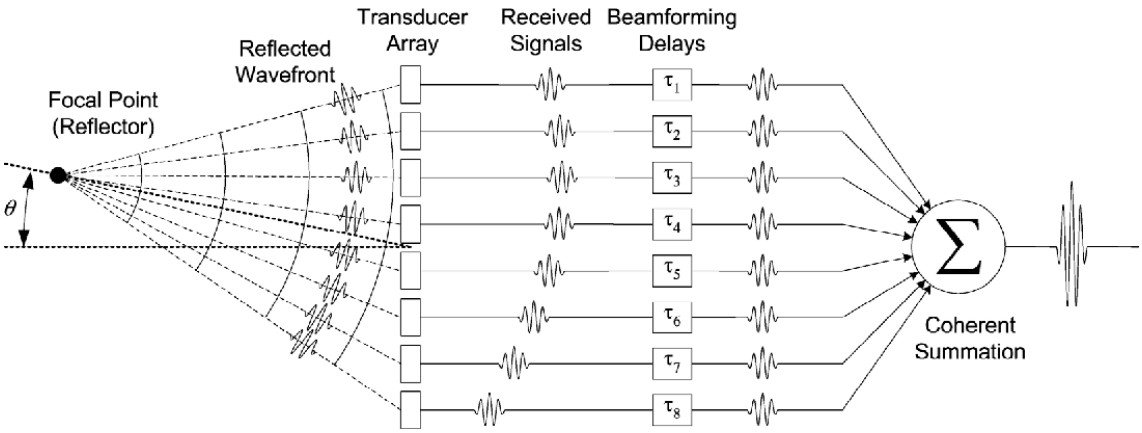


Figura 2.1: Esquema de funcionamiento de un *beamforming Delay And Sum* [5].

Cuanto mayor es el retardo de propagación, mayor es el ángulo de llegada, por lo que al sumar las señales se acaban incrementando aquellas señales que provienen de la fuente original mientras que el resto de las direcciones se atenúan.

Otros métodos más sofisticados de *beamforming* aplican un filtro a las señales que llegan al *array* junto con el alineamiento en el tiempo, el tipo de filtro usado es lo que distingue a un método de otro.

Cabe destacar que el *beamformer Delay And Sum* es el más adecuado a la hora de optimizar la ganancia frente a ruido blanco omnidireccional, WNG, y que gracias a su simplicidad a la hora de implementarlo es compatible con aplicaciones en tiempo real.

2.1.2 Superdirectivo

En comparación con el *beamforming Delay And Sum* explicado anteriormente, el *beamforming* superdirectivo es capaz de alcanzar un mayor nivel de directividad. Por lo que, si el objetivo principal es conseguir una directividad óptima, la suma realizada por el *Delay And Sum* deja de ser la opción adecuada a la hora de combinar las señales de los micrófonos. El término directividad hace referencia a la habilidad que tienen los *beamformers* para eliminar el ruido procedente de todas las direcciones sin llegar a afectar a la señal deseada que proviene de una única dirección.

Durante la primera mitad del siglo pasado los *beamformers* superdirectivos fueron usados únicamente a nivel académico, debido al ruido propio de los micrófonos de los *arrays,* así como los errores de ganancia y fase de estos mismos. No fue hasta los años 90 cuando se empezaron a ver aplicaciones reales con esta técnica. Los diseños modernos de los *beamformers* superdirectivos incluyen suposiciones de campo cercano (*nearfield*) y la posibilidad de adaptar las restricciones a los problemas actuales.

2.1.2.1 Evaluación de los *beamformers*

En primer lugar, se van a explicar las medidas necesarias para analizar el funcionamiento de este tipo de *beamformers* y, de este modo, poder entender mejor las características principales de los diseños de los *beamformers* óptimos. El modelo de señal se puede ver en la **Figura 2.2**:

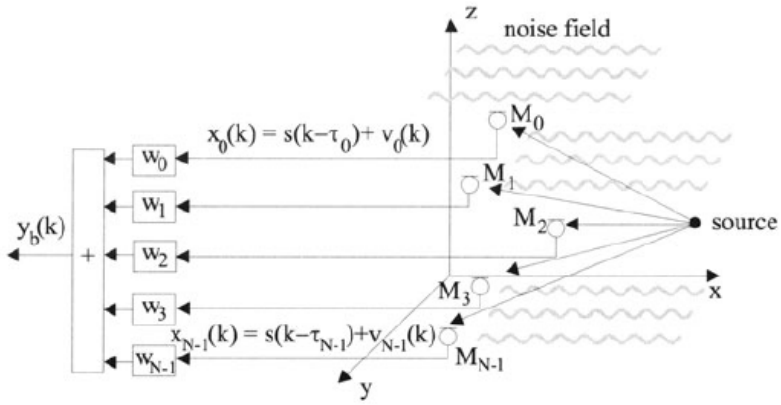


Figura 2.2: Modelo de señal con campo de ruido y la señal original deseada [4].

El modelo de señal se basa en que cada muestra de la señal discreta de entrada $x_n(k)$ en el sensor n se trata de una versión atenuada y retrasada de la señal deseada $a_n s(k - \tau_n)$, siendo τ_n el retardo con el que llega cada señal al sensor, y la componente de ruido $v_n(k)$ con estadísticas espaciales arbitrarias. Agrupando las señales de cada sensor en vectores, esto puede expresarse como:

$$\boldsymbol{x}(k) = \boldsymbol{a}s(k - \boldsymbol{\tau}) + \boldsymbol{v}(k) \tag{2.1}$$

En el dominio frecuencial, la ecuación (2.1) se puede expresar como:

$$\boldsymbol{X}(e^{j\Omega}) = \boldsymbol{S}(e^{j\Omega})\boldsymbol{d} + \boldsymbol{V}(e^{j\Omega}) \tag{2.2}$$

Donde $\boldsymbol{X}(e^{j\Omega})$, $\boldsymbol{S}(e^{j\Omega})$, $\boldsymbol{V}(e^{j\Omega})$ son, respectivamente, las transformadas de Fourier de la señal de entrada, la señal deseada o de interés y la componente de ruido.

La atenuación y los retardos en el dominio de la frecuencia se agrupan en el vector $\boldsymbol{d}$, y a su vez dependen de la geometría del *array* de micrófonos y de la dirección original de la señal:

$$\boldsymbol{d}^T = [a_0 \exp(-j\Omega\tau_0), a_1 \exp(-j\Omega\tau_1), \dots, a_{N-1} \exp(-j\Omega\tau_{N-1})] \tag{2.3}$$

Finalmente, la señal de salida del *beamformer* queda expresada del siguiente modo:

$$Y_b(e^{j\Omega}) = \sum_{n=0}^{N-1} W_n^*(e^{j\Omega}) X_n(e^{j\Omega}) = \boldsymbol{W}^H \boldsymbol{X} \tag{2.4}$$

En este caso, $W_n(e^{j\Omega})$ hace referencia a los coeficientes en el dominio frecuencial del *beamformer* del sensor n en la frecuencia Ω y el operador H es el operador hermítico. La inversa de la transformada de Fourier de la señal de salida en tiempo discreto es, por lo tanto, $y_b(k)$.

2.1.2.2 Ganancia del *array*

La ganancia del array es la mejora de la relación señal a ruido, SNR, a la salida del *array* completo con respecto a un único sensor del *array*, es decir:

$$G = \frac{SNR_{Array}}{SNR_{Sensor}} \tag{2.5}$$

Asumiendo que las señales son estacionarias, la SNR de cada sensor se calcula como la ratio entre la densidad espectral de potencia, PSD, de la señal Φ_{SS} y el ruido medio $\Phi_{V_aV_a}$.

Por lo tanto, la SNR a la salida del *array* completo se obtiene a partir de la PSD de la señal de salida.

$$\Phi_{Y_bY_b} = \boldsymbol{W}^H \boldsymbol{\Phi_{XX}} \boldsymbol{W} \tag{2.6}$$

Donde Φ_{XX} es la matriz de la densidad espectral de potencia de las señales de entrada del *array*. De este modo, cuando se presenta únicamente la señal deseada, la salida acaba siendo:

$$\Phi_{Y_bY_b}|_{Signal} = \Phi_{SS}|\boldsymbol{W}^H \boldsymbol{d}|^2 \tag{2.7}$$

Y en el caso de que solo haya ruido la señal de salida resulta ser:

$$\Phi_{Y_bY_b}|_{Noise} = \Phi_{V_aV_a} \boldsymbol{W}^H \boldsymbol{\Phi_{VV}} \boldsymbol{W} \tag{2.8}$$

En este caso, $\boldsymbol{\Phi_{VV}}$ se trata de la matriz de densidad espectral de potencia cruzada normalizada del ruido.

Por consiguiente, la ganancia del *array* se escribe como

$$G = \frac{|\boldsymbol{W}^H \boldsymbol{d}|^2}{\boldsymbol{W}^H \boldsymbol{\Phi_{VV}} \boldsymbol{W}}. \tag{2.9}$$

2.1.2.3 *Beampattern*

Otro dato importante para evaluar el funcionamiento de un *beamformer* es la respuesta del *array* frente a una onda que proviene de un ángulo y una frecuencia específicos, teniendo en cuenta los valores de azimut φ y elevación θ, en un sistema de coordenadas esféricas. De este modo, calculando la respuesta del *array* para todas los ángulos y frecuencias, obtenemos la función de transferencia espacio-frecuencial llamada *beampattern* de campo lejano, el cual se suele expresar en una escala logarítmica:

$$\left|H(e^{j\Omega}, \varphi, \theta)\right|^2\Big|_{dB} = -10\log_{10}\left(\frac{|\boldsymbol{W}^H\boldsymbol{d}|^2}{\boldsymbol{W}^H\boldsymbol{D}\boldsymbol{W}}\right) \tag{2.10}$$

Donde $\boldsymbol{D}$ representa las diferencias entre los retardos sufridos por las señales provenientes de la dirección (φ, θ) entre cada par de micrófonos:

$$\begin{gathered}\boldsymbol{D} = \boldsymbol{d}\boldsymbol{d}^H \\ D_{nm} = \exp(j\Omega\Delta\tau_{nm})\end{gathered} \tag{2.11}$$

$$\Delta\tau_{nm} = \tau_n - \tau_m \tag{2.12}$$

2.1.2.4 Diseño de *beamformers* superdirectivos

A la hora de diseñar un *beamformer* optimo, el objetivo principal es minimizar la potencia de la señal de salida $y_b(k)$ del *array*. La PSD de la salida, ecuación (2.6), se obtiene a partir de la señal de entrada y los coeficientes que se busca determinar. Para evitar la solución trivial $W_n = 0$, se añade la restricción a la hora de minimizar de que las señales en la dirección deseada no sufran ninguna distorsión:

$$\boldsymbol{W}^H\boldsymbol{d} = 1 \tag{2.13}$$

Por consiguiente, se trata de resolver el siguiente problema de minimización restringida:

$$\min_{\boldsymbol{W}} \boldsymbol{W}^H\boldsymbol{\Phi}_{\boldsymbol{XX}}\boldsymbol{W} \;\; sujeto\ a\ \boldsymbol{W}^H\boldsymbol{d} = 1 \tag{2.14}$$

Como el objetivo es una eliminación óptima del ruido, se asume que la correspondencia entre la dirección de la señal deseada y la dirección a la que apunta el *array* es perfecta, por lo que solo se usa la matriz de ruido PSD $\boldsymbol{\Phi}_{\boldsymbol{VV}}$. Se denota como el *Minimum Variance Distortionless Response* (MVDR) *beamformer* a la solución conocida de la ecuación (2.14). Se parte de

$$\boldsymbol{W} = \frac{{\boldsymbol{\Phi}_{\boldsymbol{VV}}}^{-1}\boldsymbol{d}}{\boldsymbol{d}^H{\boldsymbol{\Phi}_{\boldsymbol{VV}}}^{-1}\boldsymbol{d}} \tag{2.15}$$

y se deriva la matriz de ruido PSD $\boldsymbol{\Phi_{VV}}$ usando los multiplicadores de *Lagrange* o el gradiente. A partir de esta ecuación, podremos calcular los coeficientes óptimos para modelos teóricos de ruido o realizar una implementación adaptativa en base a las señales captadas por los sensores:

$$\boldsymbol{W} = \frac{\boldsymbol{R_{XX}}^{-1}\boldsymbol{d}}{\boldsymbol{d}^H\boldsymbol{R_{XX}}^{-1}\boldsymbol{d}} \qquad (2.16)$$
$$\boldsymbol{R_{XX}} = E\{\boldsymbol{X}\boldsymbol{X}^H\}$$

Donde E es el operador valor esperado.

2.2 Filtrado frecuencial: Wiener

El filtro de Wiener es una clase de filtro lineal de tiempo discreto que se usa en procesado de audio digital en diversas aplicaciones como la eliminación de ruido [2] [6]. El esquema básico de funcionamiento de este tipo de filtros es el siguiente:

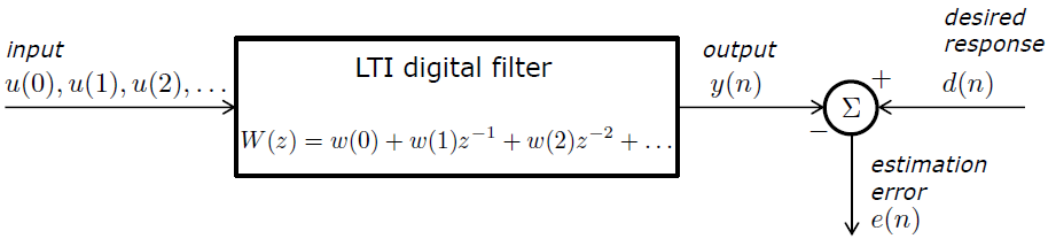


Figura 2.3: Esquema de bloques básico del funcionamiento de un filtro de Wiener [2].

La entrada del filtro consiste en la serie temporal discreta $u(n)$ y el filtro en si se caracteriza por la respuesta al impulso $w(n)$, que a su vez produce la salida $y(n)$. El objetivo es procurar que la salida $y(n)$ se parezca lo máximo posible a la señal deseada $d(n)$ o señal objetivo; por tanto, se busca que la diferencia entre estas dos señales sea mínima, es decir, que el error estimado $e(n)$ sea lo más pequeño posible. Para conseguir dicho objetivo existen diferentes criterios:

- Valor cuadrático medio del error de estimación.
- Esperanza del valor absoluto del error de estimación.
- Esperanza de terceras potencias o mayores del valor absoluto del error de estimación.

La primera opción va a ser en la que nos vamos a centrar, ya que presenta una clara ventaja frente al resto de opciones, debido a la mayor simplicidad de las matemáticas a la hora de formular el problema.

2.2.1 Principio de ortogonalidad

La salida del filtro queda definida así:

$$y(n) = \sum_{k=0}^{k=\infty} w_k^* u(n-k) \tag{2.17}$$

Tal y como se ha explicado anteriormente, la señal de error es necesaria para la estimación de la señal deseada.

$$e(n) = d(n) - y(n) \tag{2.18}$$

A la hora de optimizar el diseño del filtro se va a usar el valor cuadrático medio para estimar el error, por consiguiente, la función de coste es:

$$J(n) = E[e(n)e^*(n)] = E[|e(n)|^2] \tag{2.19}$$

De este modo, el objetivo principal de este tipo de filtro queda resumido en determinar los valores del filtro que minimicen el valor de $J(n)$. Para que esto suceda es necesario que se cumpla la siguiente condición:

$$E[u(n-k){e_0}^*(n)] = 0, \qquad k = 0, 1, 2, \dots \tag{2.20}$$

Donde ${e_0}^*(n)$ es la estimación mínima del error. Esta condición es necesaria y suficiente para que la función de coste $J(n)$ alcance su valor mínimo, el cual corresponde con ${e_0}^*(n)$ y que es ortogonal a cada muestra de entrada, definiendo de este modo el principio de ortogonalidad.

Examinando la correlación que existe entre la salida del filtro y la estimación del error podemos escribir la siguiente ecuación:

$$E[y(n)e^*(n)] = E\left[\sum_{k=0}^{k=\infty} w_k^* u(n-k)e^*(n)\right] = \sum_{k=0}^{k=\infty} w_k^* E\left[u(n-k)e^*(n)\right] \tag{2.21}$$

Siendo $y_0(n)$ la salida del filtro optimizado con el error cuadrático medio. Aplicando el principio de ortogonalidad a la ecuación (2.21) se obtiene:

$$E[y_0(n){e_0}^*(n)] = 0 \tag{2.22}$$

Esto implica que, cuando el filtro trabaja en condiciones óptimas, la salida del filtro y la señal de error son ortogonales.

2.2.2 Ecuaciones de Wiener-Hopf

Reformulando la condición necesaria para conseguir un filtrado óptimo, se sustituye la ecuación (2.18) en la ecuación (2.20), obteniendo así:

$$E\left[u(n-k)\left[d^*(n)-\sum_{i=0}^{\infty}w_{0i}u^*(n-i)\right]\right]=0, \qquad k=\ 0,1,2,\dots \tag{2.23}$$

$$\sum_{i=0}^{\infty}w_{0i}E[u(n-k)u^*(n-i)]=E[u(n-k)d^*(n)], \qquad k=\ 0,1,2,\dots \tag{2.24}$$

La esperanza $E[u(n-k)u^*(n-i)]$ es la autocorrelación de la entrada del filtro y se expresan del siguiente modo:

$$r(i-k)=E[u(n-k)u^*(n-i)] \tag{2.25}$$

Mientras que la esperanza $E[u(n-k)d^*(n)]$ denota la correlación cruzada entre la entrada del filtro y la señal deseada.

$$p(-k)=E[u(n-k)d^*(n)] \tag{2.26}$$

Por lo tanto, los coeficientes óptimos del filtro se pueden definir como:

$$\sum_{i=0}^{\infty}w_{0i}r(i-k)=p(-k), \qquad k=\ 0,1,2,\dots \tag{2.27}$$

Cabe destacar que cuando el filtro de Wiener es un filtro FIR, R representa la matriz de correlación MxM (siendo M el orden del filtro) de las muestras de entrada y p es el vector de correlación cruzada Mx1 entre las muestras de entrada y la respuesta deseada. De este modo, podemos reescribir la ecuación (2.27) obteniendo el siguiente resultado:

$$\boldsymbol{R}\boldsymbol{w}_0=\boldsymbol{p} \tag{2.28}$$

Finalmente, el cálculo de los coeficientes óptimos del filtro resulta en:

$$\boldsymbol{w}_0=\boldsymbol{R}^{-1}\boldsymbol{p} \tag{2.29}$$

2.2.3 Algoritmo *Least Mean Square (LMS)*

La esperanza $E[e(n)e^*(n)]$ del error cuadrático medio en muchos casos no está disponible ya que requeriría saber cómo son R y p a priori y dicha estimación no suele formar parte de los datos disponibles en la formulación inicial del problema al cual queremos aplicar el filtrado.

Una posible solución es el algoritmo LMS de filtrado adaptativo [7], cuyo esquema de bloques es:

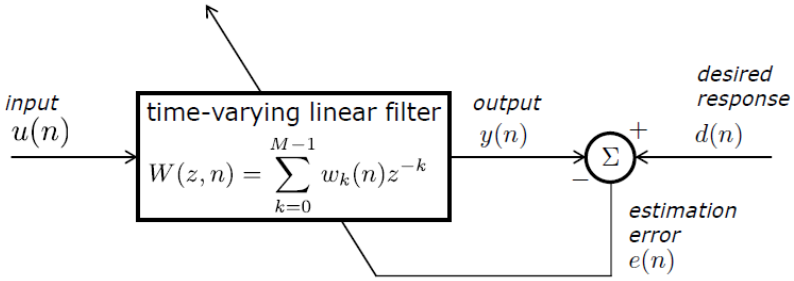


Figura 2.4: Esquema de bloques de un filtro de Wiener con LMS [2].

La primera parte del algoritmo consiste en realizar el proceso de filtrado del mismo modo que se ha explicado anteriormente, mientras que la segunda parte del algoritmo se trata de un proceso adaptativo en el que se van ajustando automáticamente los coeficientes del filtro en concordancia con la estimación del error. Por tanto, LMS se define así:

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \boldsymbol{u}(n)\boldsymbol{e}^*(n) \qquad (2.30)$$

Siendo μ el paso de adaptación, $\boldsymbol{w}(n)$ los coeficientes del filtro en el instante actual y $\boldsymbol{w}(n+1)$ los coeficientes del filtro en el instante siguiente. Es importante resaltar que con LMS el filtro deja de tener un comportamiento lineal.

2.2.4 Aplicación para la cancelación activa de ruido

En nuestro caso concreto de estudio se usará como referencia de ruido y entrada al filtro de Wiener $u(n)$ la señal grabada por un micrófono colocado en el espacio por donde absorbe el aire el extractor mientras que como señal deseada $d(n)$ usaremos la señal captada por los micrófonos del *array*. De esta forma, la señal de error estimado $e(n)$ será la salida de nuestro cancelador de ruido, ya que suponiendo que en la referencia de ruido no hay presencia de la señal de voz, la situación de ortogonalidad entre $u(n)$ y $e(n)$ a la que converge el algoritmo LMS se dará cuando en la señal $e(n)$ se haya cancelado el ruido y sólo quede señal de voz.

Capítulo 3. Metodología

Seguidamente, se exponen los métodos y herramientas usadas para la realización de las simulaciones y pruebas experimentales, tanto para el análisis de ambos tipos de filtrado como para comprobar la viabilidad de usar un sistema de control por voz en el entorno de alto ruido del que disponemos, tanto aplicando un filtrado como sin aplicarlo.

3.1 Entorno experimental

Comenzaremos explicando el prototipo de placa de inducción usado a la hora de realizar las medidas, así como los dos *arrays* de micrófonos y programas de edición acústica usados para grabar el ruido y las fuentes sonoras.

El prototipo de placa es una placa de Inducción con extractor integrado, es el modelo StudioLine de la marca Siemens con número de serie EX875LX67E [8] [9] y ha sido proporcionada por la empresa BSH Electrodomésticos España S.A. El interés de usar una placa de inducción con el extractor integrado en la misma radica en que en algunos países, como por ejemplo los países nórdicos, se prefiere usar este tipo de sistemas de ventilación para evitar la pérdida del calor del hogar que se produciría con una campana extractora como las que se emplean en España. Por este motivo, para lograr una óptima extracción de humos, la potencia del extractor debe ser muy alta, evitando así la tendencia natural del humo de ascender, lo que a su vez conlleva que el extractor se convierta en la principal fuente de ruido presente en una cocina.

Por consiguiente, en las pruebas experimentales la única fuente de ruido utilizada será el extractor, aunque se variará el nivel de potencia ofrecido por este mismo para obtener una mayor variabilidad en el experimento. Los niveles de potencia del extractor van desde el 1 hasta el 9, más un nivel máximo llamado *booster*. Según el fabricante, el nivel de contaminación acústica o ruido que ofrece la placa es de $69 dB_A$.

Figura 3.5: Placa de inducción con extractor incluido.

Para llevar a cabo las pruebas experimentales se ha utilizado el programa Audacity, que es un *software* de edición de audio y grabación de sonido digital. En concreto, se ha usado para llevar a cabo grabaciones multicanal de 6 canales hasta 16 canales, que son el número mínimo y máximo de micrófonos que contienen los dos *arrays* de micrófonos empleados.

Además, en las pruebas, en vez de usar como fuente sonora la voz de una persona, se empleará un audio reproducido a través de un altavoz con el objetivo de mantener fija la potencia acústica de la fuente sonora durante todas las pruebas y así mismo facilitar la reproducción de estas. Por esto, será necesario calibrar los altavoces para que emitan la potencia deseada, esto se llevará a cabo con el programa Room EQ, que es un *software* de análisis acústico para hacer mediciones acústicas y análisis de habitaciones y altavoces y el micrófono calibrado miniDSP UMIK-1.

3.1.1 *Array* rectangular

El primer *array* de micrófonos surge de la idea de integrar los micrófonos en el interior de la placa de inducción para así conseguir una mayor comodidad de cara al usuario y un mayor valor añadido a la placa. Sin embargo, esto supondrá una mayor dificultad ya que los micrófonos se encuentran en un entorno de alto ruido y la señal que capten tendrá una atenuación debida al cristal de la propia placa, estos efectos se estudiarán en profundidad en los resultados de las pruebas experimentales.

Integrar los micrófonos en el interior de la placa nos permite diseñar un *array* rectangular de dimensiones similares a la placa, que son 80x52 cm, lo cual supone una ventaja ya que nos permite aumentar el número de micrófonos con respecto a los que puede usar un dispositivo externo (6 micrófonos normalmente) y aumentar la resolución del *beamforming* que se utilice, como se explicó en ***Filtrado espacial: Beamforming***.

El diseño e implementación del *array* se ha llevado a cabo en el Laboratorio de Audio y Video de la Universidad de Zaragoza dentro del Trabajo de Fin de Máster del estudiante Raúl Gracia Escorihuela [10]. Se ha contado con hasta 16 micrófonos, por lo que se ha decidido emplear 3 versiones del *array* rectangular con 8, 12 y 16 micrófonos con el objetivo de estudiar cuál ofrece unas mejores prestaciones con y sin aplicar procesado de señal. El interior de la placa de inducción puede observarse en la **Figura 3.6**.

Figura 3.6: Interior de la placa de inducción con extractor integrado.

La distribución de los micrófonos que forman el array en el interior de la placa ha sido realizada observando los espacios disponibles dentro de la propia placa y tratando de realizar un reparto uniforme de dichos micrófonos y, sobre todo, evitando interferir en el correcto funcionamiento de la placa de inducción para poder realizar las pruebas experimentales con la placa encendida.

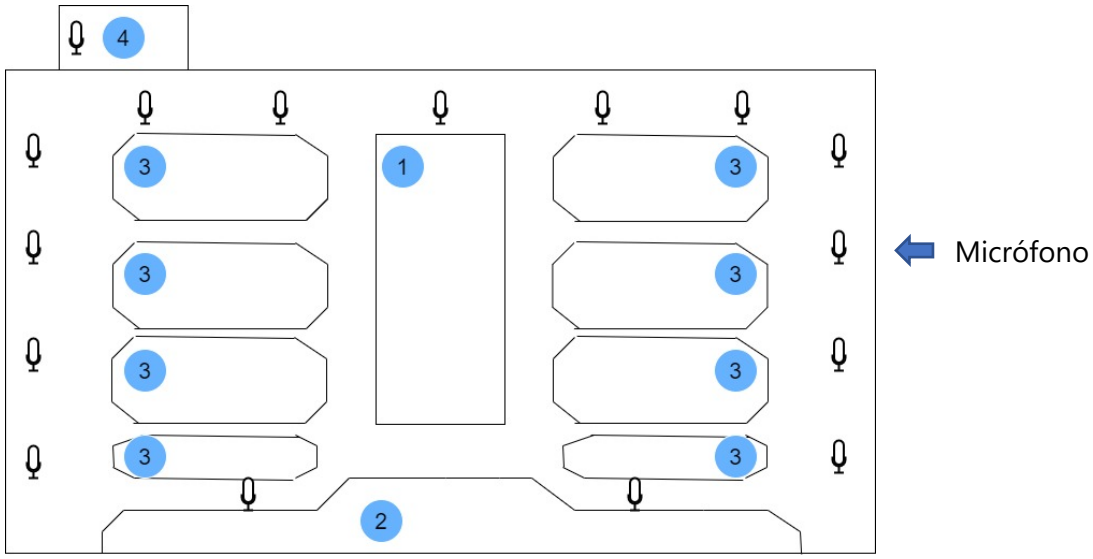


Figura 3.7: Distribución de los micrófonos en el interior de la placa de inducción.

Los elementos enumerados en la **Figura 3.7** son los siguientes:

1. Extractor de humos.
2. Panel de control.
3. Bobinas de inducción.
4. Ventilador del extractor de humos.

Los micrófonos utilizados son de tipo MEMS (*Micro-Electral-Mechanical-Systems*) [11], emplean el protocolo de comunicación PDM y tienen una sensibilidad de $-26\ dBFS$ que ha sido necesario tener en cuenta dicho valor a la hora de realizar los cálculos de SPL . Estos micrófonos se han utilizado integrados en una *beakout board*, diseñada por la empresa Adafruit [12], que resulta sencilla y que tiene el siguiente aspecto:

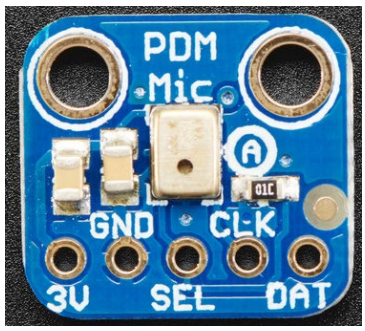


Figura 3.8: *Breakout board* del micrófono MEMS.

Finalmente, para capturar las señales de los micrófonos se ha usado la placa MCHStreamer de la empresa MiniDSP [13] [14]. Se trata de una interfaz de audio USB multicanal y que permite múltiples protocolos de comunicación a la hora de incorporar y leer micrófonos. A través de esta placa somos capaces de leer las señales de hasta 16 micrófonos y enviarlas por medio de una conexión USB a un ordenador, para así ver y escuchar la señal capturada por cada micrófono con Audacity. El aspecto de la placa MCHStreamer se observa en la **Figura 3.9**.

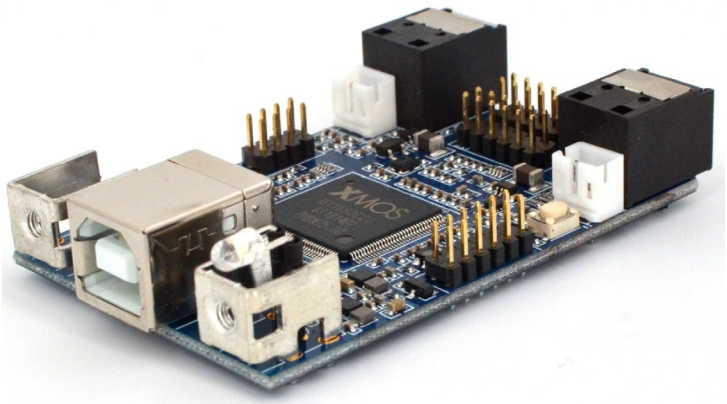


Figura 3.9: Placa MCHStreamer.

3.1.2 UMA – 8

El segundo *array* de micrófonos usado es un *array* de micrófonos circular llamado UMA-8 de MiniDSP [15]. Cabe destacar que cuenta con tecnología multinúcleo XMOS y 7 micrófonos MEMS de alto rendimiento, uno de ellos colocado en el centro y los 6 restantes en la circunferencia exterior, tal y como se muestra a continuación.

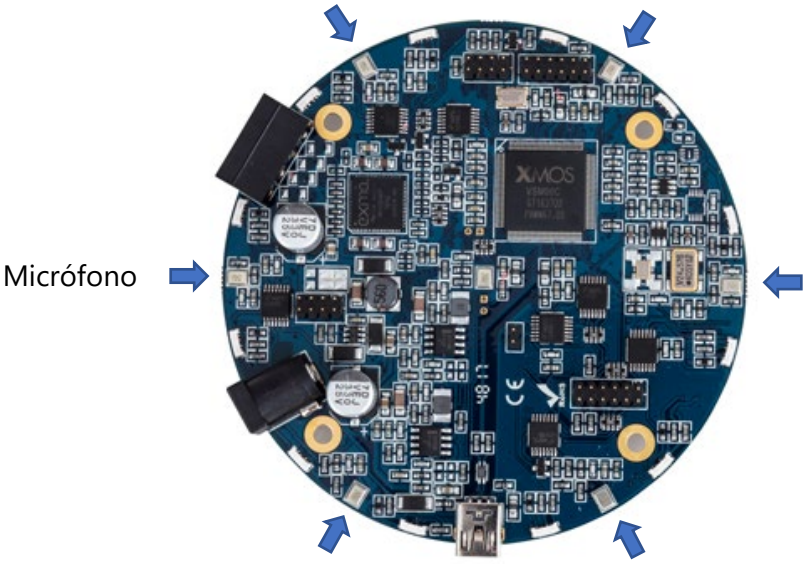


Figura 3.10: Vista superior del UMA-8. Las flechas indican la posición de los micrófonos.

Este dispositivo puede trabajar a una frecuencia de muestreo de $11/16/32/44{,}1/48\ kHz$ con una resolución de 24 bits y cuentan con 2 modos de funcionamiento, modo RAW y modo DSP. Sin embargo, en este caso únicamente nos interesa el modo RAW, ya que dispone de 8 canales correspondientes a los 7 micrófonos y a una entrada PDM. A lo largo del proyecto, este ha sido el modo empleado con una frecuencia de muestreo de $44{,}1\ kHz$, ya que nos permitía grabar las señales capturadas por cada micrófono para su posterior procesado en Matlab.

El interés de usar este *array* está en que tiene una estructura y funcionamiento similar a un Echo Dot de Amazon que implemente el asistente de voz Alexa, para así ver las diferencias que habría entre usar el *array* rectangular y usar un Echo Dot de Amazon.

3.2 Implementación en Matlab

A la hora de realizar el estudio de las grabaciones, se ha usado la herramienta Matlab [16], que es un sistema de análisis numérico que dispone de un entorno de desarrollo con su propio lenguaje de programación.

Se han desarrollado *scripts* de Matlab tanto para la simulación acústica como para analizar las pruebas experimentales. Para la simulación acústica se ha generado un escenario 3D en el que se calcula el retardo con el que llegarían las señales a los diferentes micrófonos para su posterior procesado tanto si fuera el caso del *array* rectangular (con 8, 12 y 16 micrófonos) como del *array* circular. También se han implementado los dos algoritmos de *beamforming* junto con sus respuestas espaciales y frecuenciales a lo largo del espectro frecuencial cuando la dirección de interés a la que apunta el *beamforming* es el punto donde estaría el usuario o apuntando al extractor. Asimismo, se ha programado el filtro de Wiener LMS. Además, se realiza el cálculo y

visualización de los mapas de localización de fuentes sonoras, en 2D y 3D, por medio del algoritmo SRP-PHAT, así como las representaciones de las señales antes y después de aplicar los diferentes procesados de señal y sus periodogramas.

Finalmente, para poder estudiar la viabilidad de incorporar control por voz, se ha llevado a cabo la representación y el análisis de los niveles de SPL y SNR de cada una de las grabaciones.

3.3 Alexa y la Raspberry Pi

Para estudiar la viabilidad de implementar un asistente voz para controlar una cocina, se ha utilizado el asistente de voz proporcionado por la empresa Amazon, llamado Alexa, en una Raspberry Pi 3 Model B+. Para implementar el asistente Alexa se ha seguido un tutorial de Amazon para la instalación en una Raspberry Pi [17].

Cabe destacar que, a la hora de usar Alexa, se ha implementado un *loopback* en la Raspberry Pi cuya función es mapear la entrada de audio del sistema con la salida de audio del sistema. El interés de realizar este *loopback* radica en poder estudiar si Alexa reconoce su palabra de despertar con diferentes niveles de ruido del extractor de una manera offline, de manera que al reproducir las grabaciones hechas con los dos arrays de micrófonos Alexa las analice como si estuvieran siendo capturadas por un micrófono en ese mismo instante. Esto también permite evaluar cuantas veces se reconoce la palabra de despertar tras haber aplicado los diferentes algoritmos de filtrado sin que tengan que estar implementados en tiempo real.

Figura 3.11: Esquema del *loopback*.

En este proyecto nos hemos centrado en la detección de la palabra de despertar de Alexa, esto se debe a que se planteó como hipótesis el que una vez detectada la palabra Alexa se pudiera bajar automáticamente la potencia del extractor para facilitar el reconocimiento de voz de las ordenes que diera el usuario.

Capítulo 4. Simulaciones y medidas reales

Con el objetivo de estudiar las prestaciones ofrecidas por los dos *arrays* y los algoritmos explicados anteriormente, se ha llevado a cabo una simulación acústica y una serie de pruebas, las cuales se exponen a continuación, y cuyos resultados se estudiaron tanto en términos de SNR como de localización espacial de las fuentes sonoras.

Como hipótesis inicial, la atenuación de las señales que llegan a los micrófonos del *array* rectangular está altamente influenciada por la presencia del cristal de la placa de inducción, mientras que, en el *array* circular, al estar los micrófonos en el exterior de la placa, no se produce dicha atenuación. Por tanto, se realizaron pruebas para ver cuál de los dos *arrays* obtenía mayor porcentaje de detección de la palabra de despertar de Alexa y también se estudió la respuesta del *array* rectangular sin la presencia del cristal, es decir, con la placa de inducción abierta.

Por último, para comprobar la hipótesis planteada de que el algoritmo superdirectivo consigue mejores resultados que el algoritmo *Delay and Sum* debido a que el primero es más directivo que el segundo, se realizaron pruebas para determinar la capacidad de detección de la palabra de despertar aplicando ambos algoritmos en los dos *arrays* y además se hicieron algunas pruebas con el filtro de Wiener LMS.

4.1 Simulación acústica

Mediante un proceso de simulación acústica se estudiaron las señales recibidas por los diferentes micrófonos y su respuesta a dichas señales. Para ello, se generó un escenario en 3D en el que se definieron las posiciones de los micrófonos, así como las posiciones de la fuente de ruido y la voz.

A la hora de realizar la simulación, se ha generado un *script* en Matlab en el que se ha creado el escenario 3D, con las señales y los micrófonos, y se han llevado a cabo mapas de localización de las fuentes sonoras y mapas de presión sonora.

La simulación ha permitido probar diferentes posiciones en las que colocar los micrófonos y por tanto diferentes geometrías de array de micrófonos. Se ha variado el número de micrófonos, se han implementado varios tipos de filtrado espacial y frecuencial y, por último, se han usado diferentes señales de voz y ruido para así poder parametrizar correctamente el sistema.

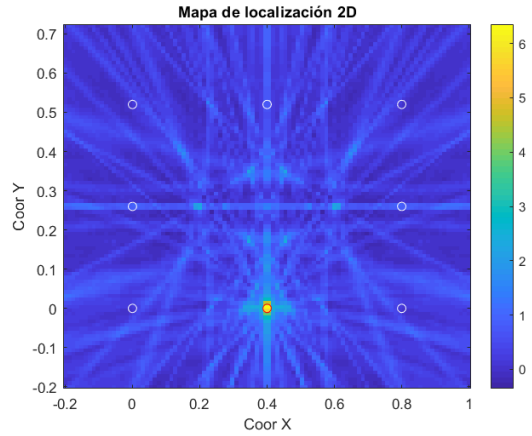
Cabe destacar que la principal limitación de la simulación acústica realizada es que se ha usado un modelo de propagación de las señales en el espacio libre, es decir, no se ha tenido en cuenta el cristal de la placa de inducción, ya que para poder su efecto habría sido necesario recurrir a modelos de elementos finitos que conllevaría una complicación sustancial añadida.

4.1.1 Mapas de localización sonora

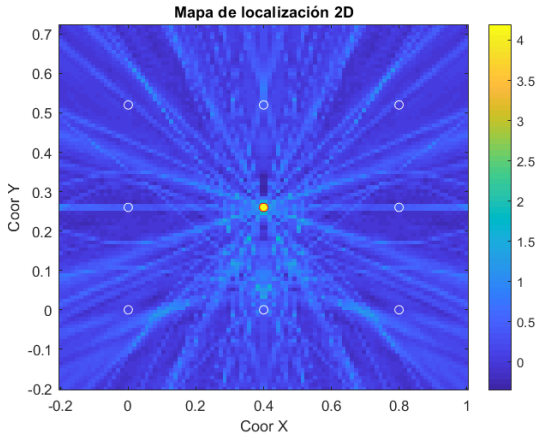
Lo primero que se ha simulado son los mapas de localización y de presión sonora con 8, 12 y 16 micrófonos, con una geometría rectangular del *array* como si los micrófonos estuvieran dentro de la placa de inducción. Aunque realmente la localización de fuentes sonoras no es uno de los objetivos principales, este tipo de mapas nos dan una buena imagen visual de la capacidad que tiene el *array* de distinguir unas direcciones de llegada de las señales de otras direcciones.

Se obtuvieron los mapas en 2D en la altura en la que estaría la voz del usuario y en la altura en la que estaría el extractor con 8 micrófonos.

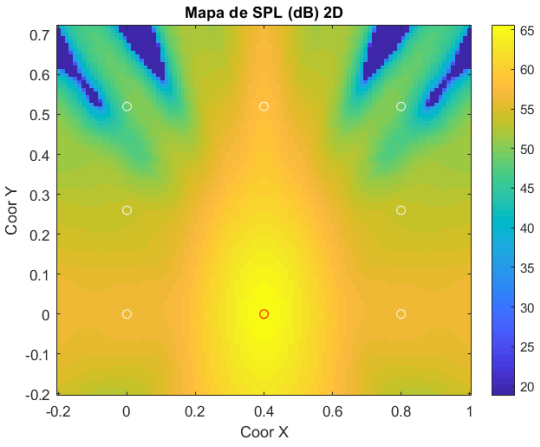
Mapa de localización altura voz



Mapa de localización altura extractor



Mapa de presión sonora (dB) altura voz



Mapa de presión sonora (dB) altura extractor

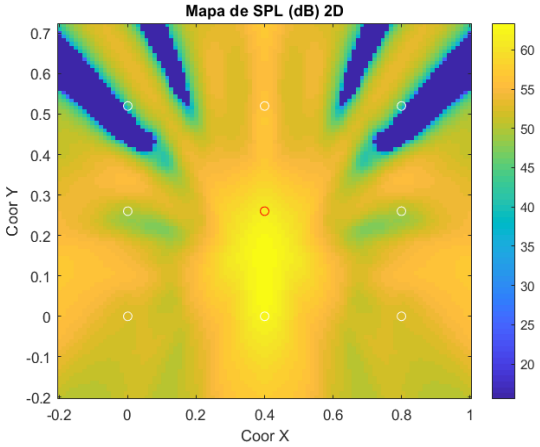
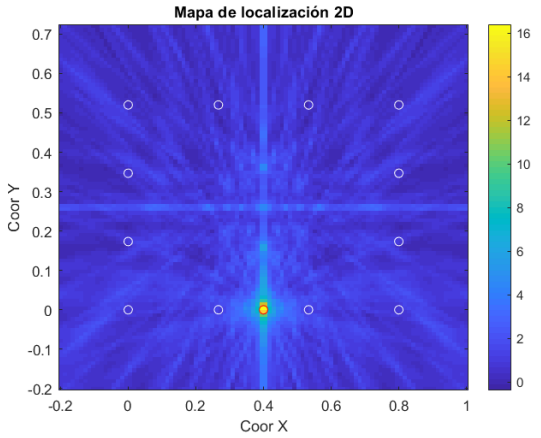


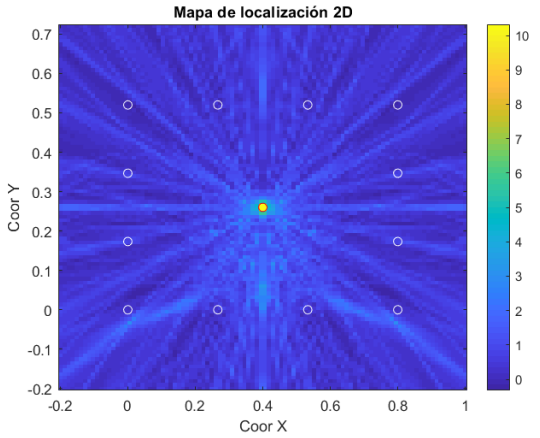
Figura 4.12: Mapas de localización y de presión sonora con 8 micrófonos.

La diferencia principal entre el mapa de localización y el mapa de presión sonora es que en el primero se aplica la transformación de fase del algoritmo SRP-PHAT mientras que en el segundo tipo de mapas no se aplica para poder obtener así los resultados reales de presión sonora. Como resultado, el primer tipo de mapa es capaz de llevar a cabo una localización muy ajustada, sin embargo, en el segundo tipo de mapa la localización se vuelve más difusa.

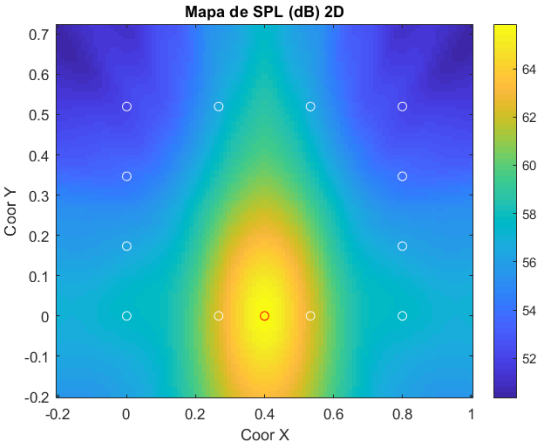
Mapa de localización altura voz



Mapa de localización altura extractor



Mapa de presión sonora (dB) altura voz



Mapa de presión sonora (dB) altura extractor

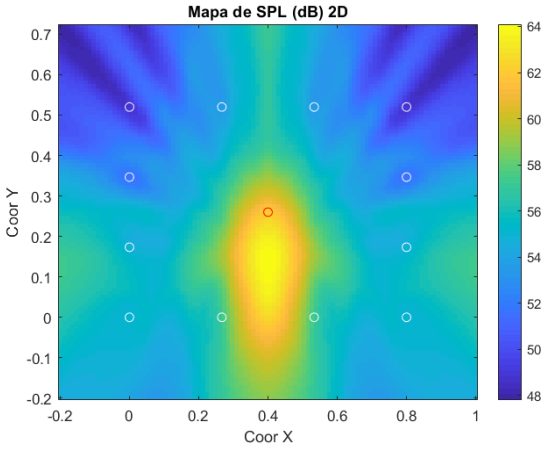
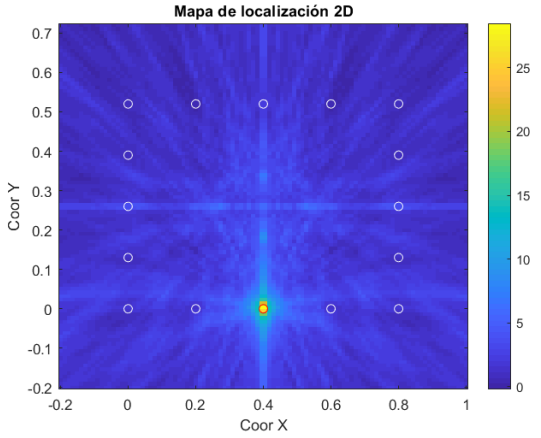
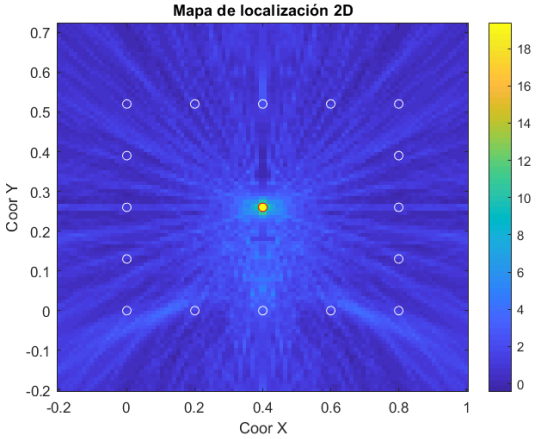


Figura 4.13: Mapas de localización y de presión sonora con 12 micrófonos.

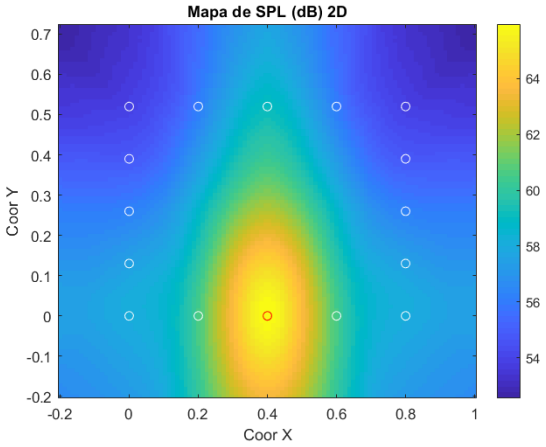
Mapa de localización altura voz



Mapa de localización altura extractor



Mapa de presión sonora (dB) altura voz



Mapa de presión sonora (dB) altura extractor

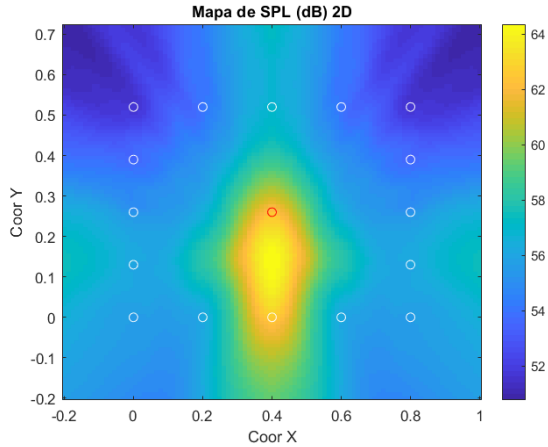


Figura 4.14: Mapas de localización y de presión sonora con 16 micrófonos.

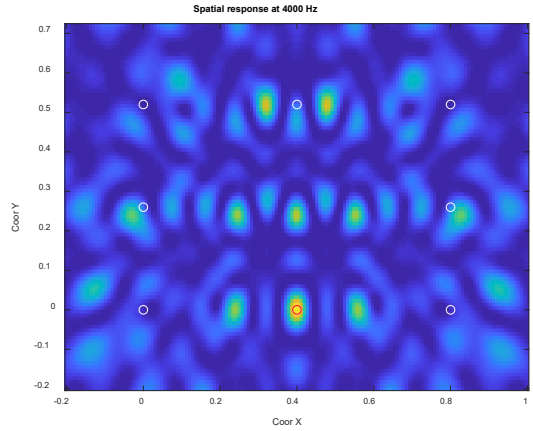
En los mapas las circunferencias blancas representan los puntos donde se situarían los micrófonos y las circunferencias rojas serían los puntos donde estaría la voz o el punto central del extractor. En dichos mapas se observa que conforme aumenta el número de micrófonos se localiza una menor cantidad de máximos, lo cual es todavía más evidente en los mapas de presión sonora, destacando el caso de 8 micrófonos en el que el máximo ocupa prácticamente toda la superficie simulada. También se puede apreciar que apenas hay diferencias entre los mapas con 12 y 16 micrófonos, detalle que se seguirá estudiando a continuación.

4.1.2 Filtrado espacial: *Delay and Sum*

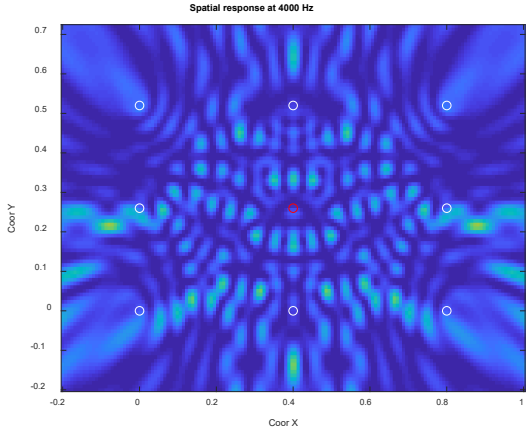
El siguiente paso en la simulación fue el desarrollo de un algoritmo de filtrado espacial o *beamforming:* el algoritmo se llama *Delay and Sum*. Se trata del algoritmo de *beamforming* más sencillo de implementar y se ha presentado brevemente en el apartado ***Delay And Sum***. Para poder evaluar los resultados del *Delay and Sum*, se ha simulado un escenario en el que la señal que llega a los micrófonos, que estarían de nuevo dentro de la placa de inducción, es una suma de la voz y del ruido del extractor. Por otro lado, el *beamforming* apunta a la dirección en la que se encuentra la voz, de modo que toda señal que llegue a los micrófonos proveniente de direcciones diferentes quedará filtrada.

Al simular se ha podido estudiar la respuesta que, aproximadamente, ofrecería el *Delay and Sum*, obteniendo, en primer lugar, la respuesta espacial del filtro en diferentes frecuencias con un array de 8, 12 y 16 micrófonos.

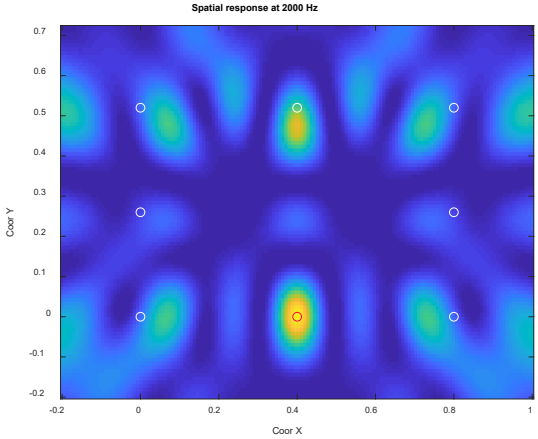
Corte del plano a la altura de la voz



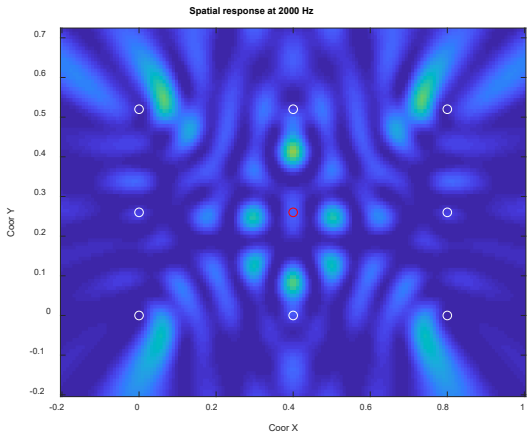
Corte del plano a la altura del extractor



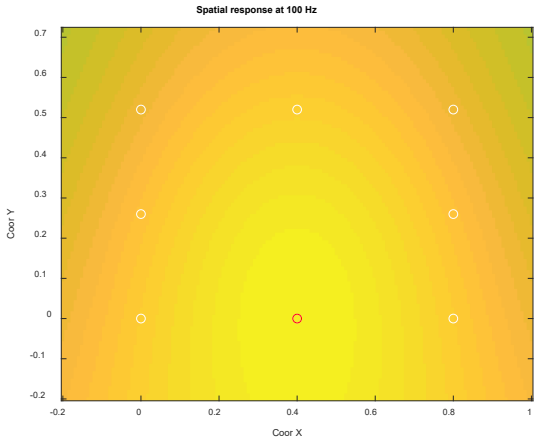
Corte del plano a la altura de la voz



Corte del plano a la altura del extractor



Corte del plano a la altura de la voz



Corte del plano a la altura del extractor

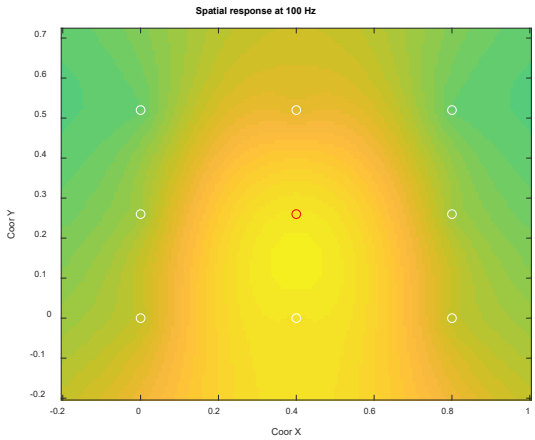
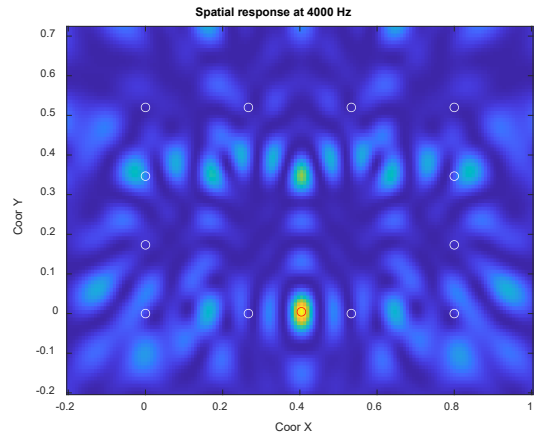
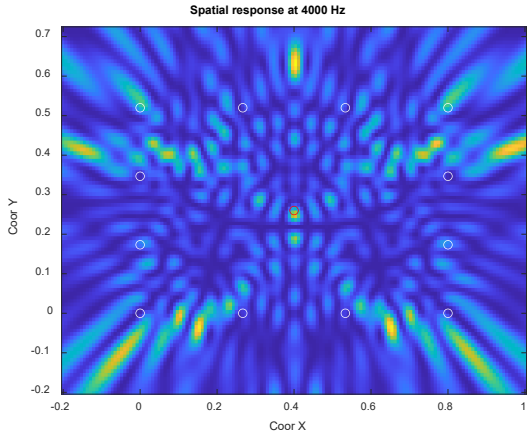


Figura 4.15: Respuesta espacial a $4\ kHz, 2\ kHz\ y\ 100\ Hz$ del *Delay and Sum* con 8 micrófonos.

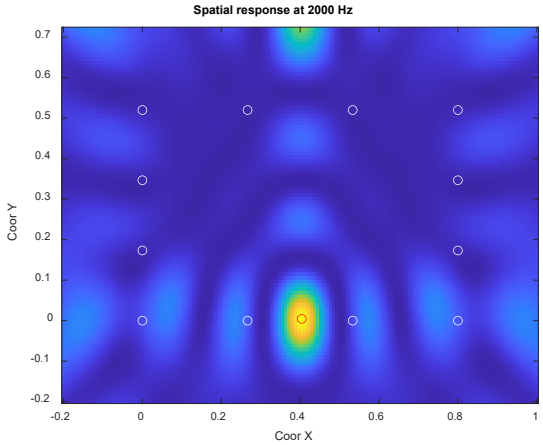
Corte del plano a la altura de la voz



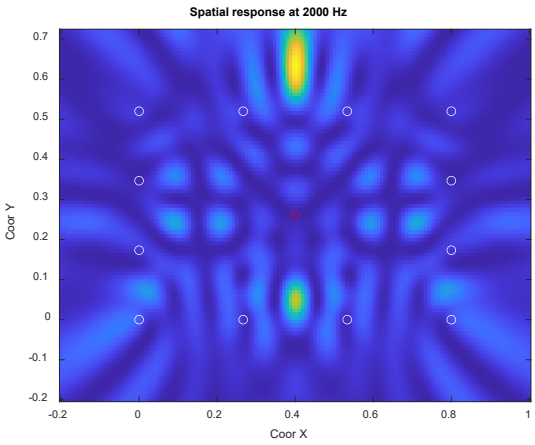
Corte del plano a la altura del extractor



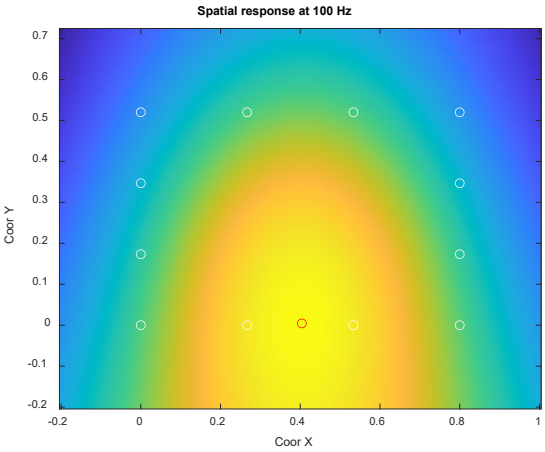
Corte del plano a la altura de la voz



Corte del plano a la altura del extractor



Corte del plano a la altura de la voz



Corte del plano a la altura del extractor

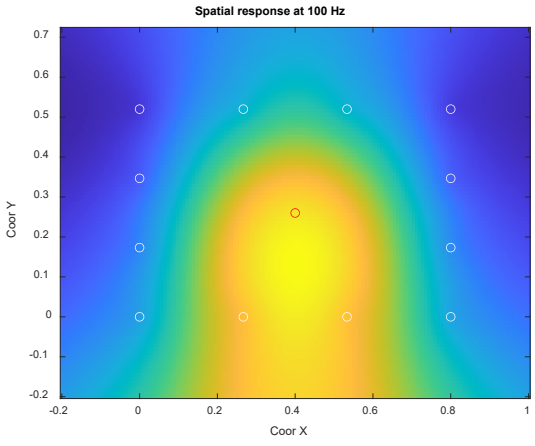


Figura 4.16: Respuesta espacial a $4\ kHz, 2\ kHz\ y\ 100\ Hz$ del *Delay and Sum* con 12 micrófonos.

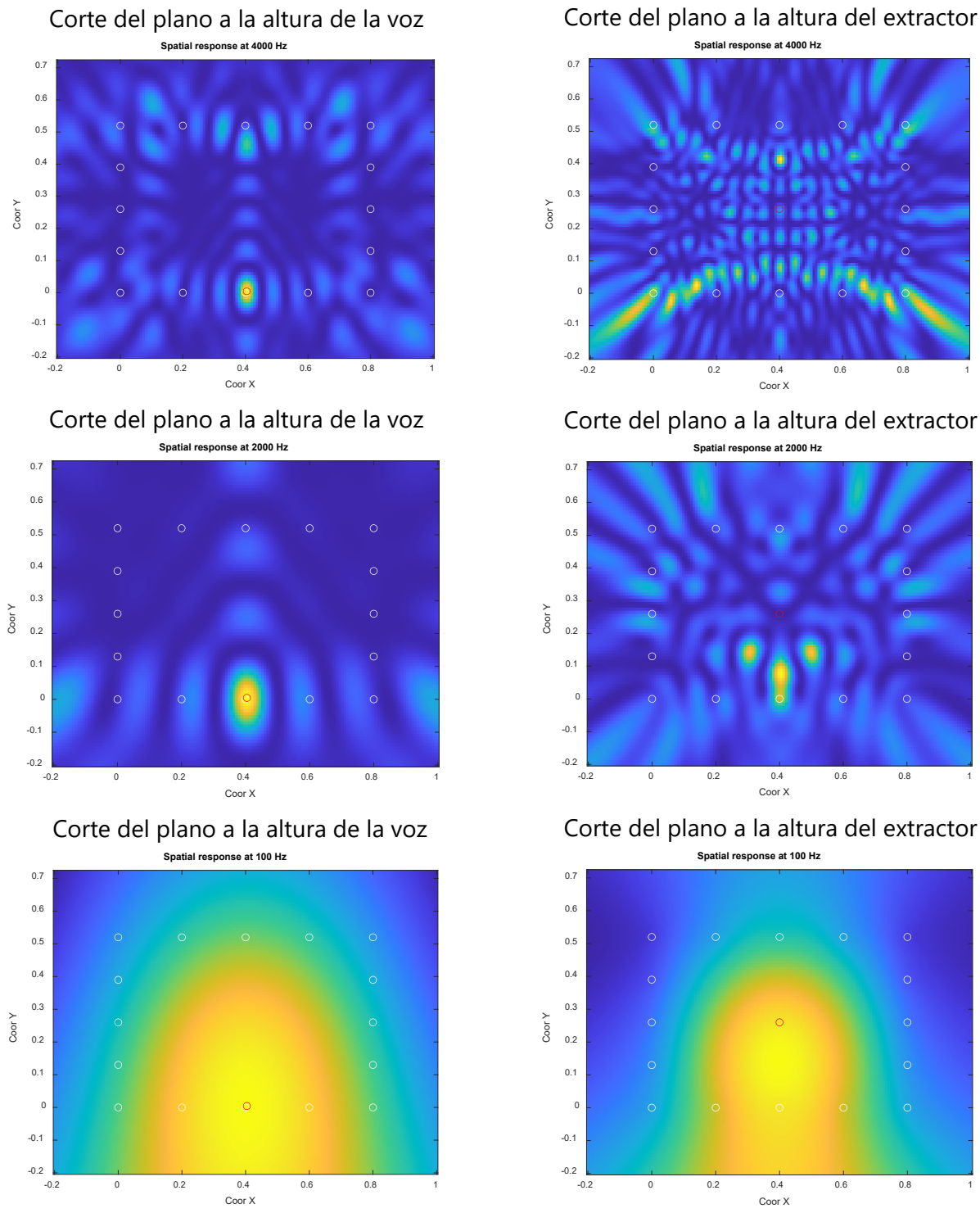


Figura 4.17: Respuesta espacial a $4\ kHz, 2\ kHz\ y\ 100\ Hz$ del *Delay and Sum* con 16 micrófonos.

Cuanto mayor es la frecuencia más estrecho es el filtro espacial, es decir, el máximo localizado en el punto en el que está la voz es más pequeño pero, adicionalmente, se produce un mayor efecto de *aliasing* y, por tanto, el filtro contiene una mayor cantidad de máximos no deseados. En cambio, en frecuencias bajas como $100\ Hz$, el filtro deja pasar todas las señales desde cualquier dirección ya que evalúa como un máximo todo el plano. Además, se observa una mejora en la localización

cuando se pasa de 8 micrófonos a 12, los máximos son más estrechos especialmente a $100\ Hz$, sin embargo, apenas hay mejora cuando se pasa de 12 micrófonos a 16.

Seguidamente, se ha obtenido la respuesta frecuencial en el punto central del extractor con 8, 12 y 16 micrófonos, respectivamente. Los resultados son los siguientes:

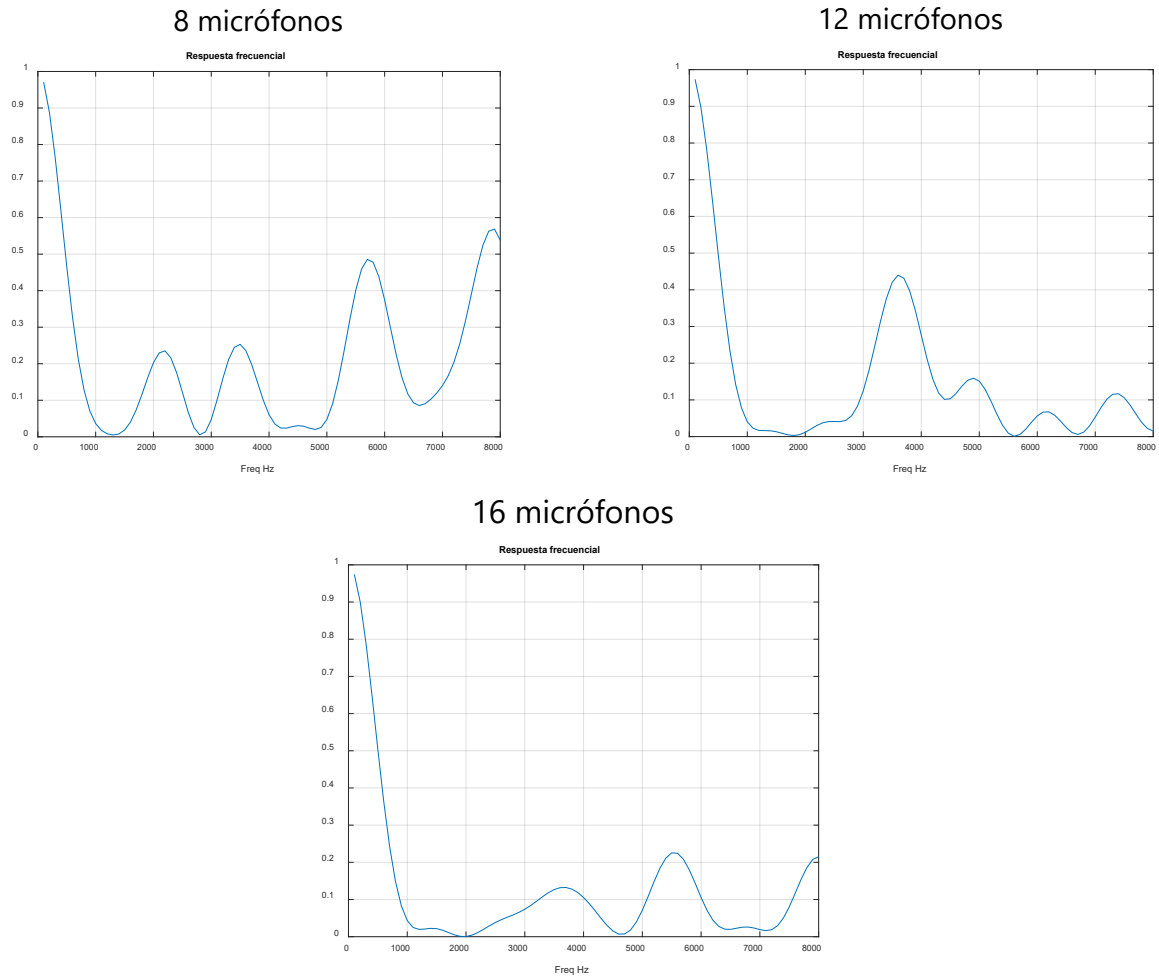


Figura 4.18: Respuesta frecuencial del *Delay and Sum* en el punto central del extractor con 8, 12 y 16 micrófonos.

En la respuesta frecuencial se puede ver que a bajas frecuencias la respuesta tiene valores muy altos, lo que significa que el filtro deja pasar todas las señales sin llegar a filtrar prácticamente nada. También se observa el efecto que tiene el número de micrófonos en la respuesta frecuencial, ya que conforme mayor es la cantidad de micrófonos menores picos se producen en frecuencias superiores a $1\ kHz$ ya que se reduce el *aliasing* espacial.

Por último, se ha proporcionado el periodograma, que es la estimación de la densidad espectral de la señal antes y después del *Delay and Sum*, nuevamente con 8, 12 y 16 micrófonos, pero en este caso la simulación se ha realizado solo con la señal de ruido.

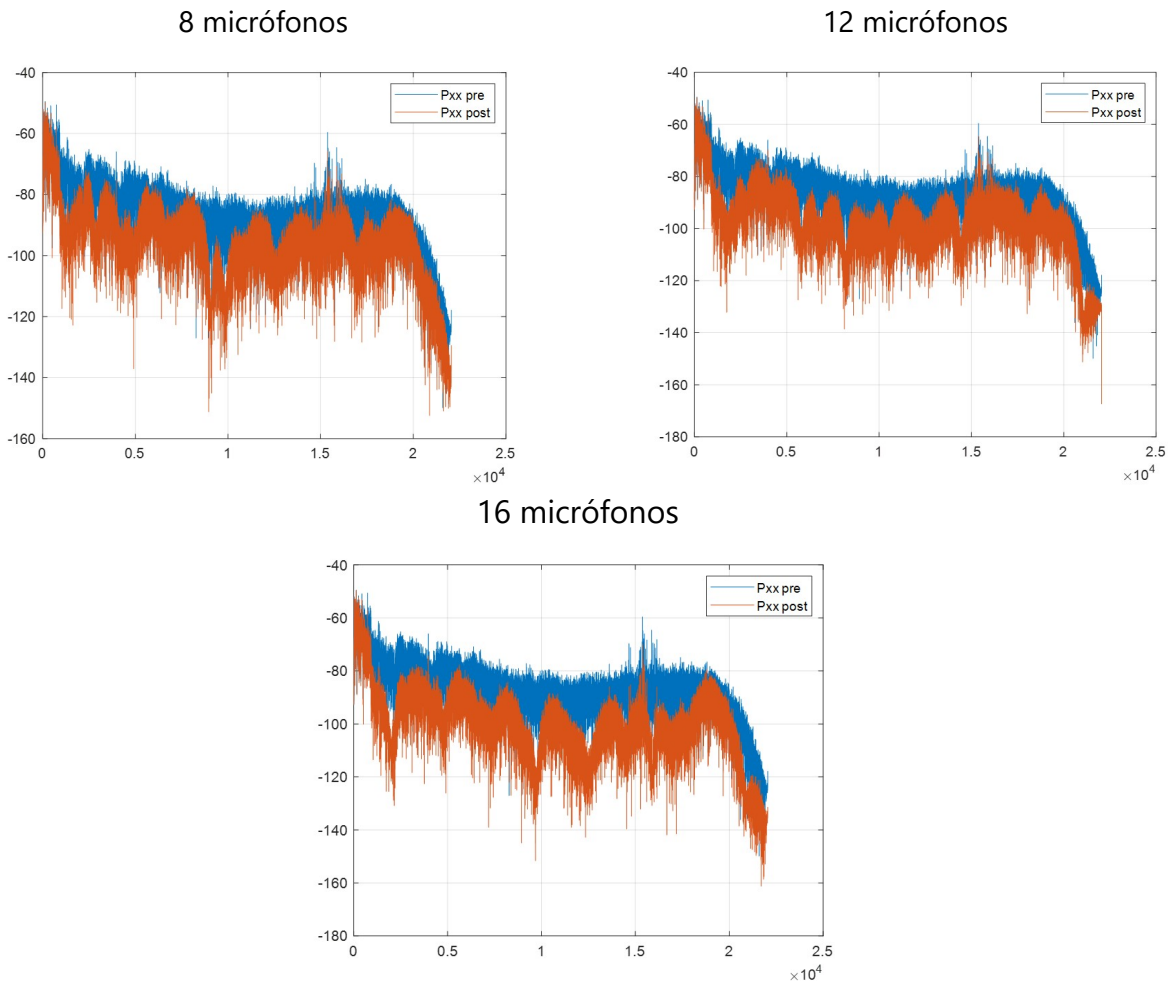


Figura 4.19: Periodogramas con 8, 12 y 16 micrófonos del *Delay and Sum*.

Con el periodograma se comprueba lo que se ha afirmado con las respuestas frecuenciales, a mayor número de micrófonos mayor es la capacidad de filtrado en frecuencias superiores a $1\ kHz$ pero en ningún caso se logran filtrar las frecuencias inferiores.

Adicionalmente, se ha evaluado el nivel de presión sonora, SPL, de la señal que llega a los micrófonos y de la señal obtenida después se haber realizado el *beamforming*. Los resultados se pueden ver en la siguiente tabla. Se observa una disminución de $3\ dB$ aproximadamente en todos los casos, por efecto del filtrado espacial.

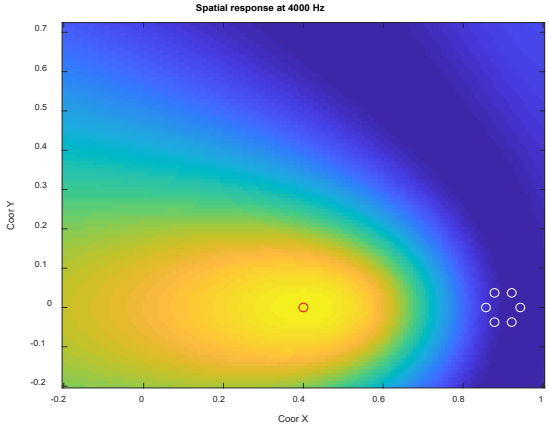
	8 MICRÓFONOS	**12 MICRÓFONOS**	**16 MICRÓFONOS**
SPL PRE	63.3576 dB	63.3576 dB	63.3576 dB
SPL POST	60.6825 dB	60.7529 dB	60.7062 dB

Tabla 4.1: Tabla de SPL del *Delay and Sum* con 8, 12 y 16 micrófonos.

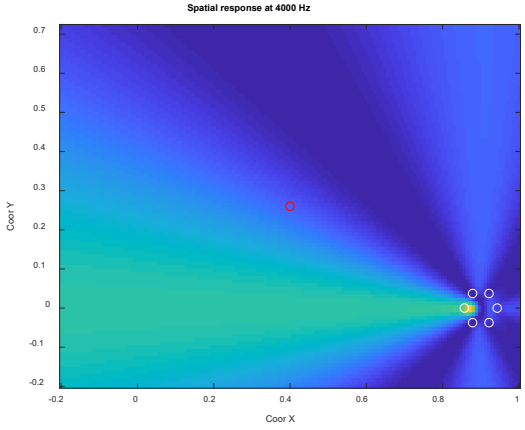
A continuación, se llevó a cabo una evaluación del *beamforming Delay and Sum* obteniendo la respuesta espacial y frecuencial con un *array* de 6 micrófonos circular, similar a un Echo Dot de Amazon, para poder comparar los resultados con los obtenidos anteriormente con el *array* rectangular. La simulación se ha realizado con el *array* circular

situado en la esquina inferior derecha de la placa de inducción y los resultados son los siguientes:

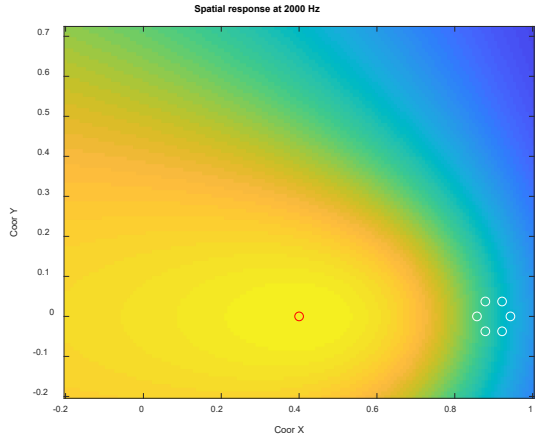
Corte del plano a la altura de la voz



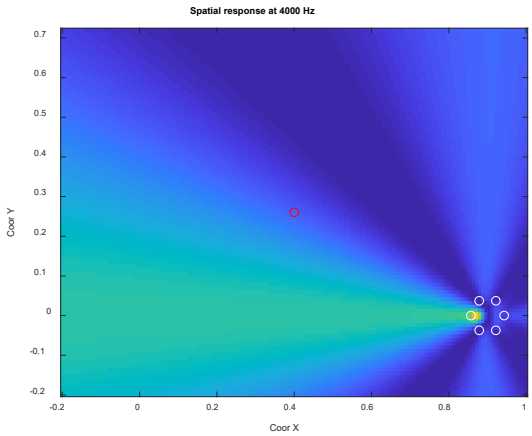
Corte del plano a la altura del extractor



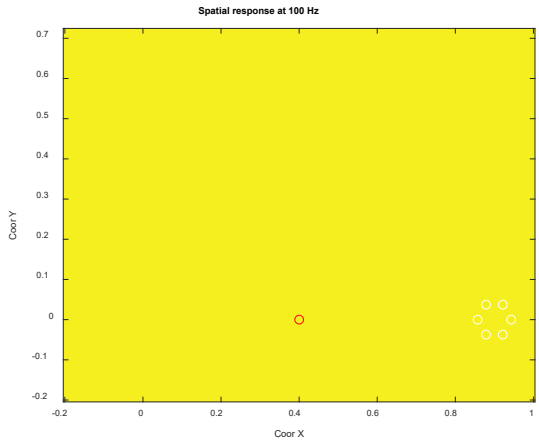
Corte del plano a la altura de la voz



Corte del plano a la altura del extractor



Corte del plano a la altura de la voz



Corte del plano a la altura del extractor

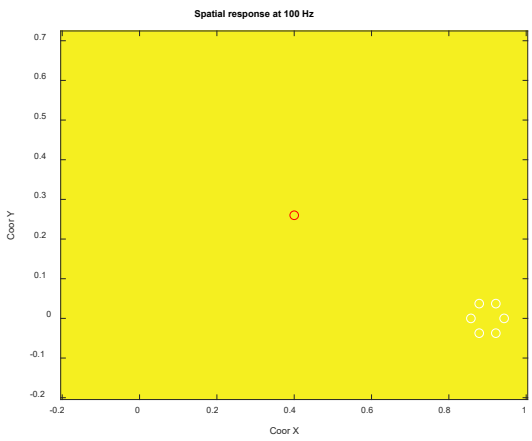


Figura 4.20: Respuesta espacial a $4\ kHz, 2\ kHz\ y\ 100\ Hz$ del *Delay and Sum* con el *array* circular.

Comparando los resultados con la **Figura 4.15**, la **Figura 4.16** y la **Figura 4.17** se puede concluir que el *array* circular en las frecuencias altas presenta un máximo mucho más grande, es decir, no es capaz de filtrar espacialmente tanto como cuando se usa un

array rectangular, y en las frecuencias bajas también tiene un rendimiento peor. Esto se debe a que las dimensiones del *array* circular son mucho más reducidas, reduciendo de este modo la capacidad del *array* de discriminar las señales provenientes de muchas direcciones. Sin embargo, este *array* tiene la ventaja de que es capaz de reducir en gran cantidad el efecto de *aliasing* debido a la menor separación entre sus micrófonos.

La respuesta frecuencial obtenida en la posición del extractor es:

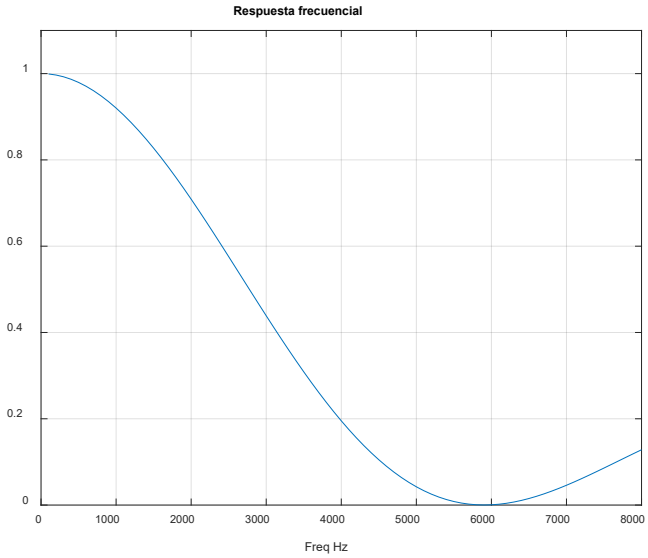


Figura 4.21: Respuesta frecuencial del Delay and Sum en el punto central del extractor con el array circular.

Al igual que sucede con la respuesta espacial, si comparamos el resultado de la figura anterior con el de la **Figura 4.18**, se puede observar como el *Delay and Sum* no ofrecería un buen filtrado hasta llegar a frecuencias superiores a los $5\ kHz$, siendo de este modo un resultado 5 veces superior de frecuencia de corte al obtenido con el *array* rectangular de 8 micrófonos. Recalcando la idea comentada anteriormente de que el filtro no sería capaz de filtrar señales de diferentes direcciones tan estrechamente como se pretende en este caso.

Siguiendo la misma metodología que con el *array* rectangular se han obtenido los valores de SPL y el periodograma de la señal de ruido que llega a los micrófonos y de la señal tras haber realizado el *beamforming*.

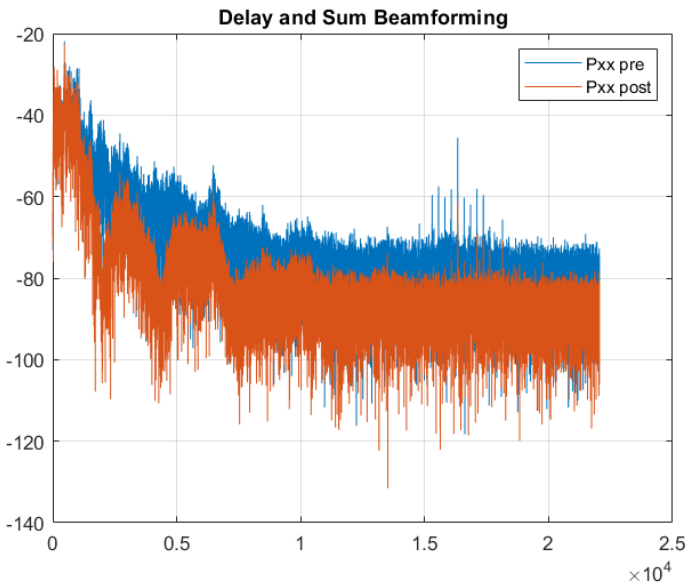


Figura 4.22: Periodogramas con el *array* circular del *Delay and Sum*.

	6 MICRÓFONOS
SPL PRE	63.24 dB
SPL POST	61.81 dB

Tabla 4.2: Tabla de SPL del *Delay and Sum* con el *array* circular.

En este caso, la disminución que se produce debida al filtrado es de $1.5\ dB$ aproximadamente y en comparación con los datos obtenidos en la **Tabla 4.1**, en los que la disminución del ruido del extractor era de unos $3\ dB$, el filtrado del ruido es inferior con la simulación del *array* circular.

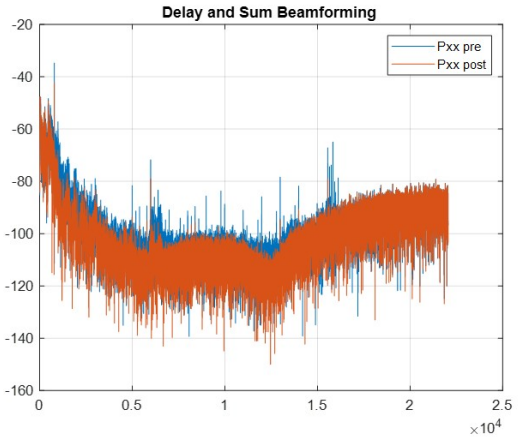
4.1.3 Filtrado espacial: algoritmo superdirectivo

Tras esta comparativa, se ha implementado un nuevo algoritmo de *beamforming* llamado superdirectivo. Se trata de un algoritmo más complejo que el *Delay and Sum* y que es capaz de filtrar espacialmente una dirección con mucha más precisión, como se explica en el apartado ***Superdirectivo***. De esta manera, el último paso de la simulación ha consistido en comparar los dos algoritmos de *beamforming* y evaluar sus prestaciones.

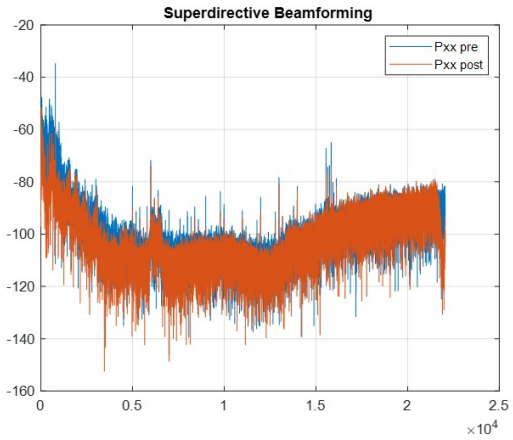
Para llevar a cabo este último paso, se han utilizado grabaciones de ruido de la placa de inducción con diferentes niveles de potencia del extractor y de nuevo simulando que los 16 micrófonos del *array* rectangular se encuentren dentro de la placa de inducción.

En primer lugar, se han obtenido los periodogramas del ruido para ambos algoritmos antes y después de aplicarlos, el resultado es:

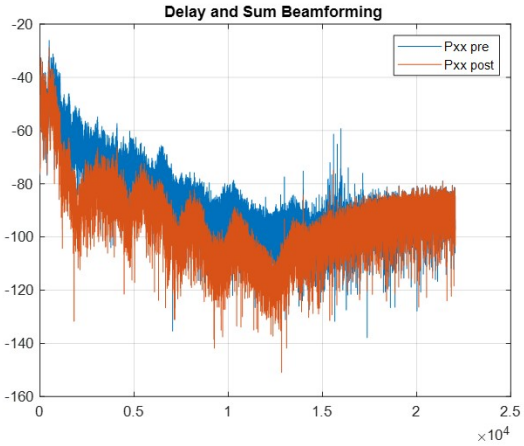
Extractor al 1



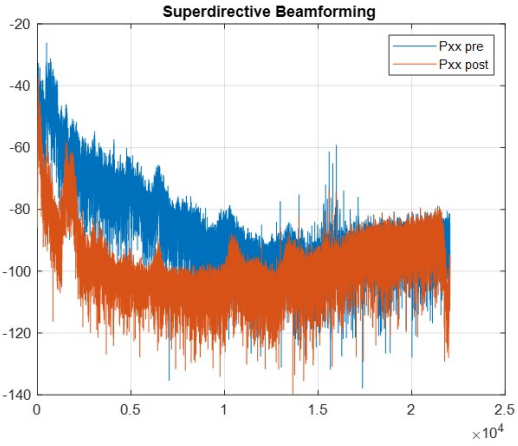
Extractor al 1



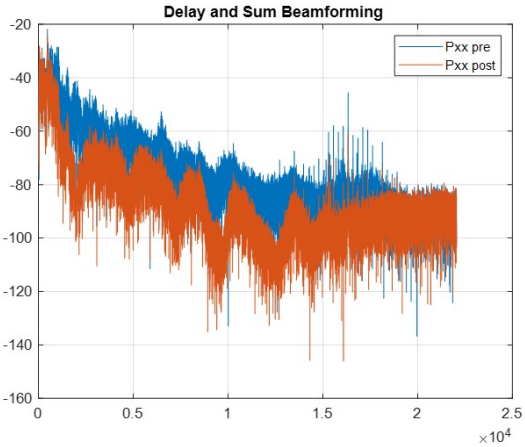
Extractor al 6



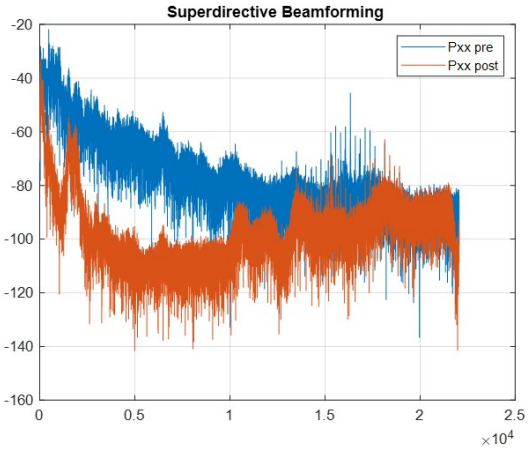
Extractor al 6



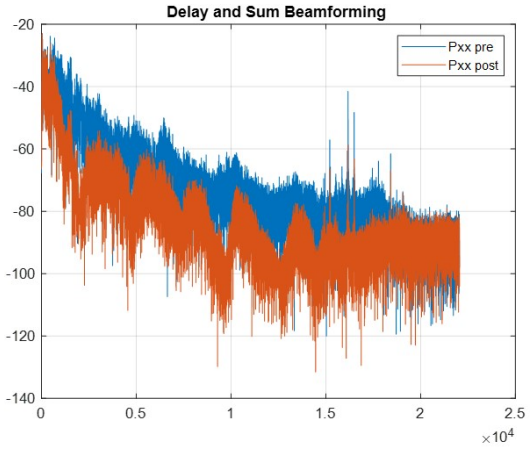
Extractor al 9



Extractor al 9



Extractor al Booster Max



Extractor al Booster Max

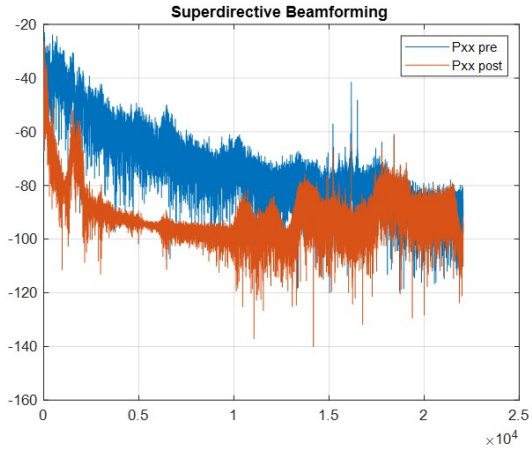
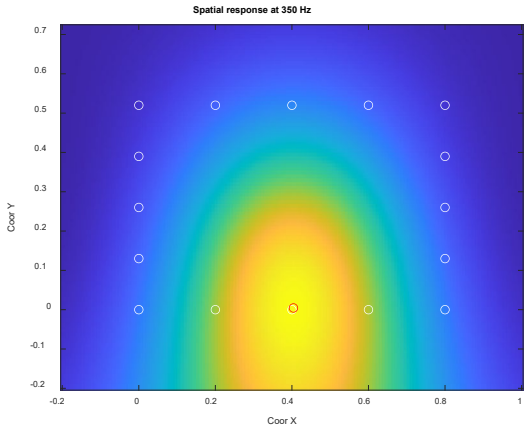


Figura 4.23: Periodogramas con 16 micrófonos del *Delay and Sum* y del superdirectivo.

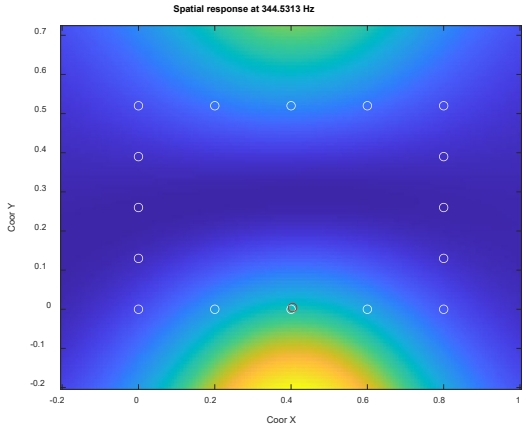
En los periodogramas se ve que el algoritmo superdirectivo es capaz de filtrar mayor cantidad de señal, especialmente cuanto mayor es la potencia del extractor de la placa de inducción. Aunque a bajas frecuencias ambos algoritmos ofrecen unas prestaciones no deseables ya que nos son capaces de filtrar demasiado, se puede destacar que el algoritmo superdirectivo tiene un mejor resultado.

Adicionalmente a los periodogramas, se ha estudiado la respuesta espacial en diferentes frecuencias, desde $350\ Hz$ hasta $4100\ Hz$ aproximadamente, tanto a la altura de la voz como del extractor, así como la respuesta frecuencial en el punto central del extractor, todo ello con una potencia del extractor de 9. Algunas de las frecuencias estudiadas se muestran seguidamente, el resto de las frecuencias se encuentran en el ***Anexo B. Simulación acústica***.

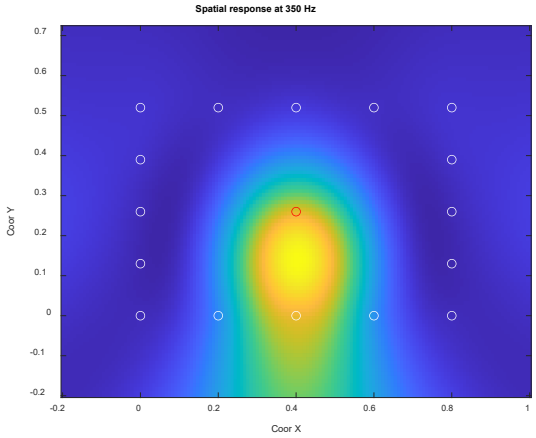
Delay and Sum Voz



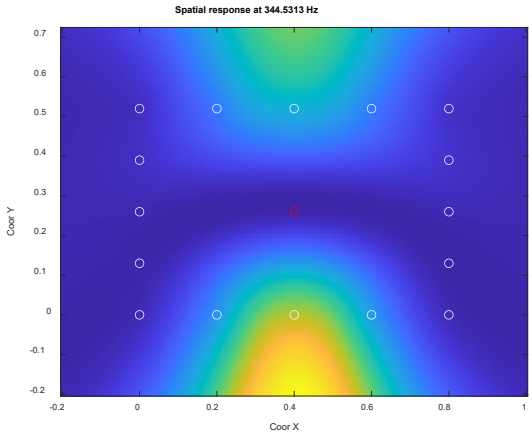
Superdirectivo Voz



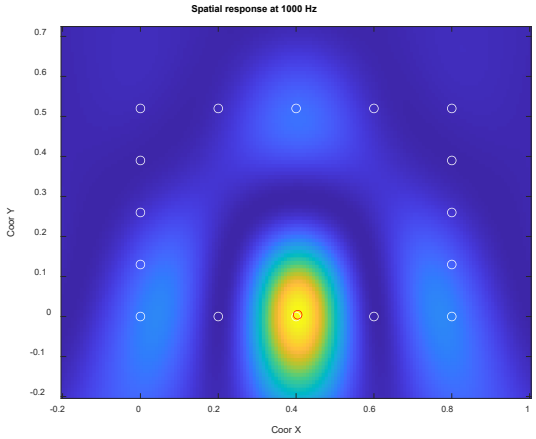
Delay and Sum Extractor



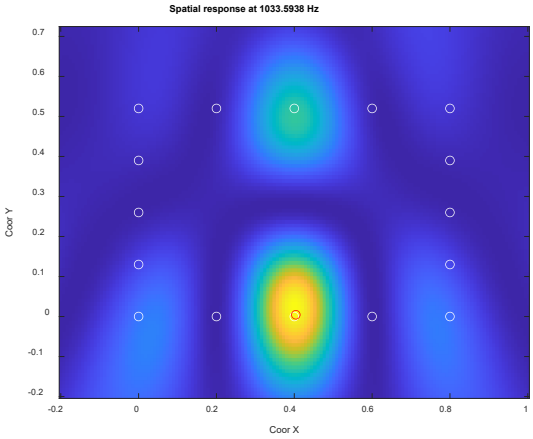
Superdirectivo Extractor



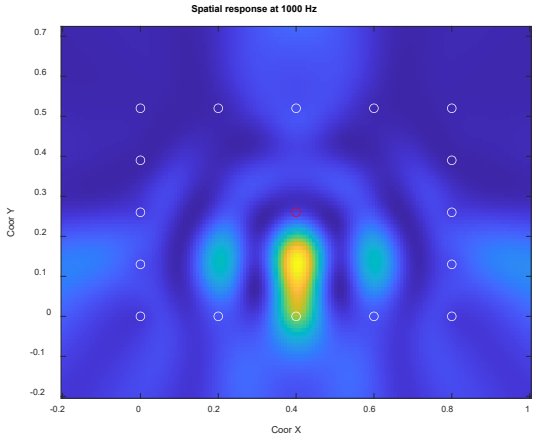
Delay and Sum Voz



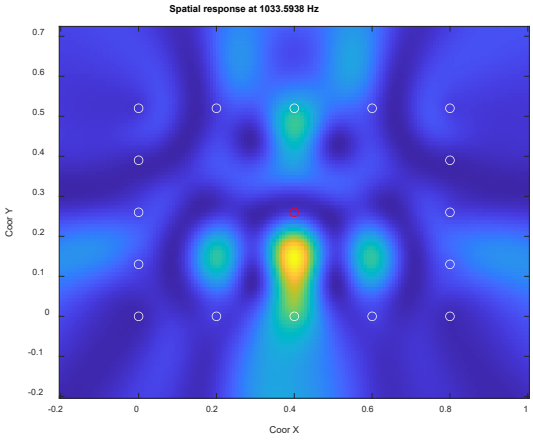
Superdirectivo Voz



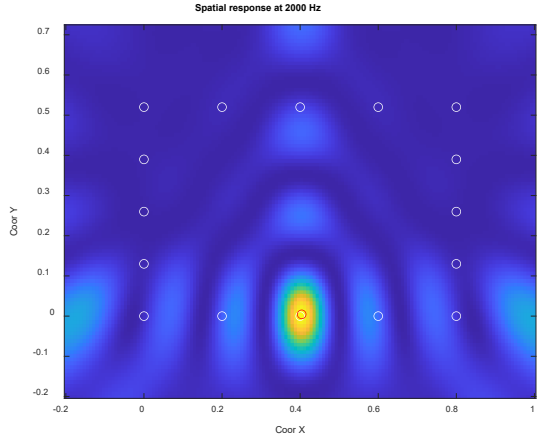
Delay and Sum Extractor



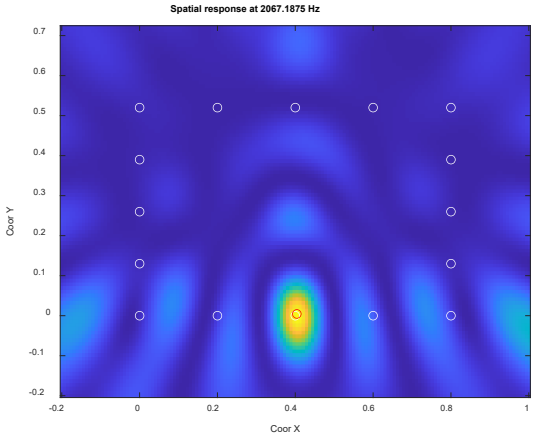
Superdirectivo Extractor



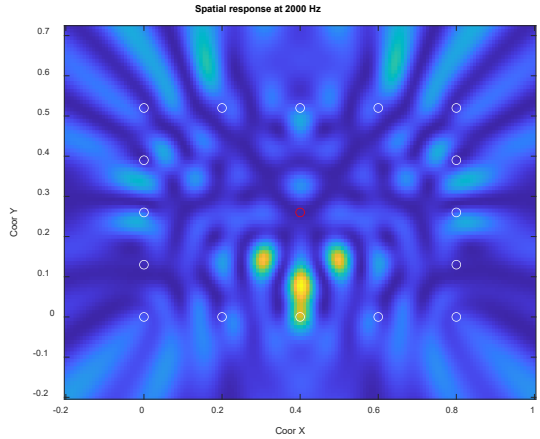
Delay and Sum Voz



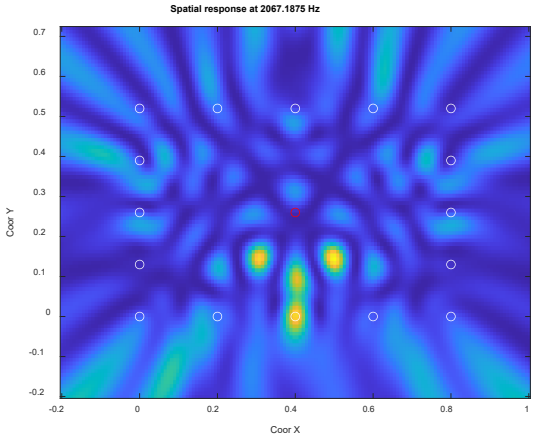
Superdirectivo Voz



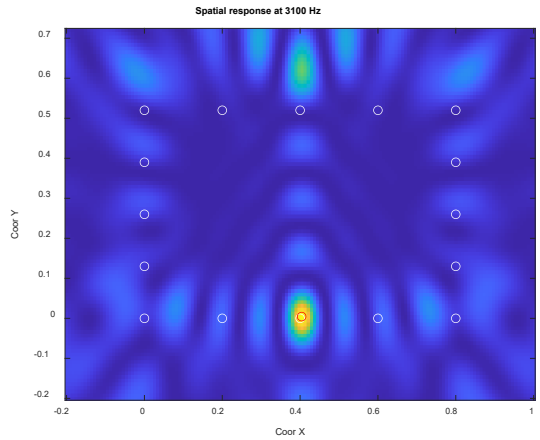
Delay and Sum Extractor



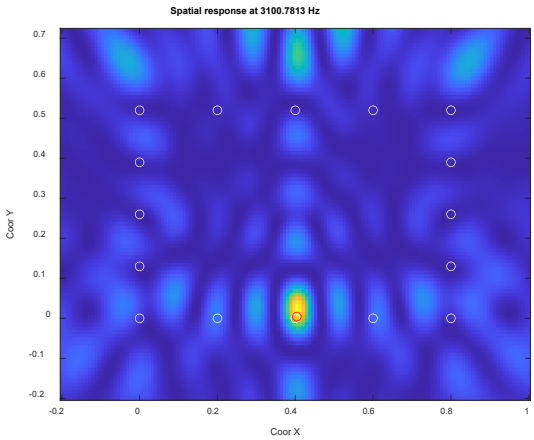
Superdirectivo Extractor



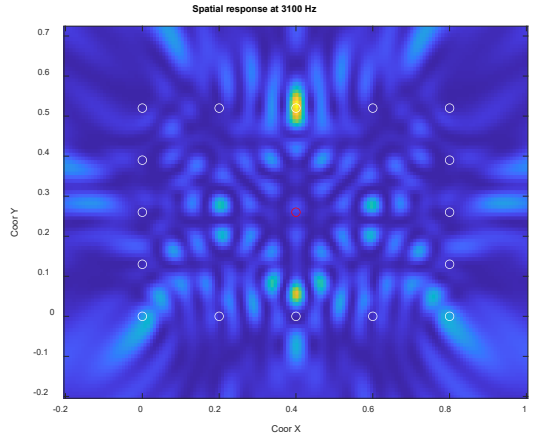
Delay and Sum Voz



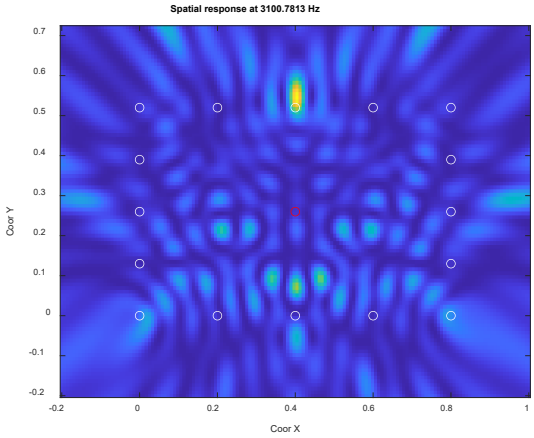
Superdirectivo Voz



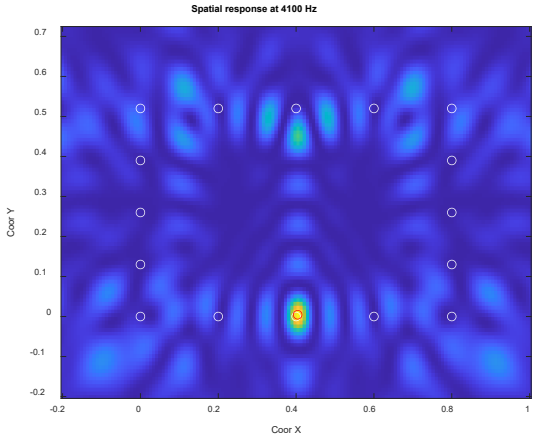
Delay and Sum Extractor



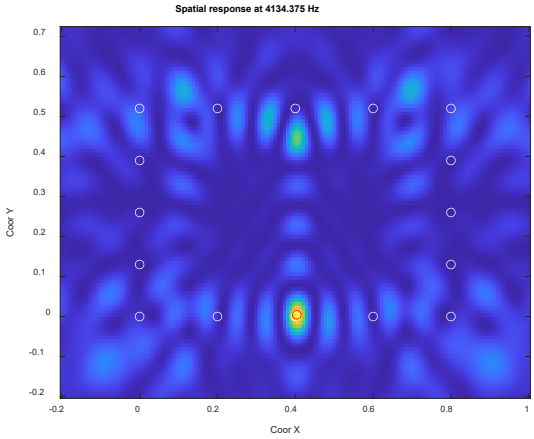
Superdirectivo Extractor



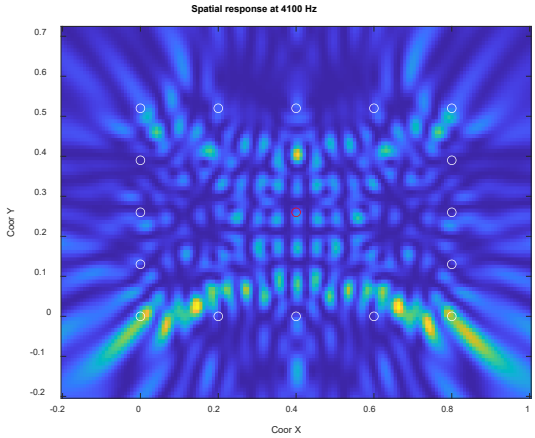
Delay and Sum Voz



Superdirectivo Voz



Delay and Sum Extractor



Superdirectivo Extractor

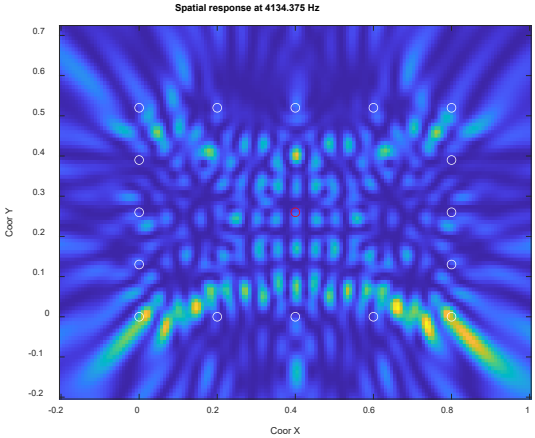


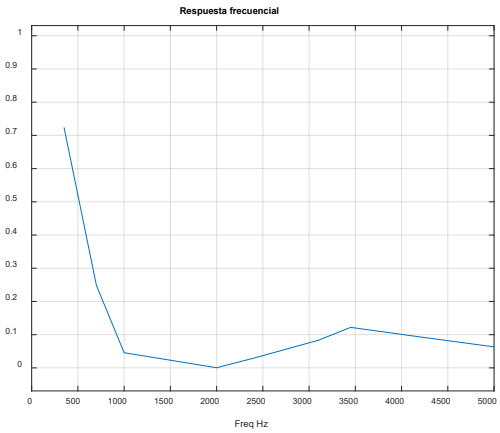
Figura 4.24: Respuesta espacial en diferentes frecuencias del *Delay and Sum* y del superdirectivo.

En la **Figura 4.24** se pueden ver una serie de efectos, como que el algoritmo superdirectivo es capaz de conseguir mínimos ligeramente más pequeños, especialmente a bajas frecuencias, y por tanto el filtrado es algo más directivo, sin embargo, también se produce mayor *aliasing* en el superdirectivo por lo que a frecuencias más altas genera un mayor número de máximos que el *Delay and Sum*. Y

también que el algoritmo superdirectivo consigue reducir su ganancia a frecuencias bajas en la posición del extractor a costa de desplazar su máximo hacia atrás en la dirección a la que si quiere apuntar realmente y aumentar su ganancia global para que en la posición deseada sea unitaria. Esto da buen resultado en este caso porque la única fuente de ruido es el extractor y la prioridad del *beamforming* es anular dicho ruido, pero podría dar más problemas si hubiese más fuentes de ruido.

Por otro lado, la respuesta frecuencial en el punto central del extractor es la siguiente:

Delay and Sum



Superdirectivo

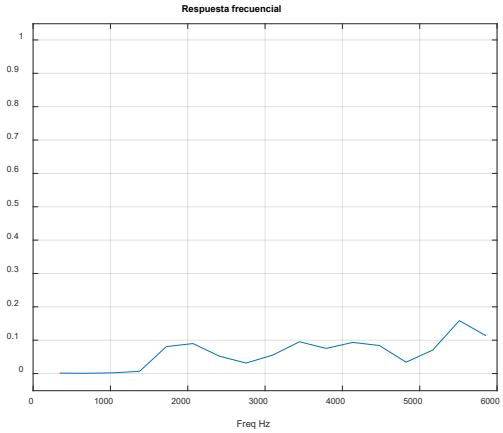


Figura 4.25: Respuesta frecuencial del *Delay and Sum* y del superdirectivo.

En la respuesta frecuencial es donde más claramente se observa que el algoritmo superdirectivo es capaz de filtrar más, especialmente en bajas frecuencias donde apenas hay picos.

Finalmente, se ha realizado una tabla con la SNR antes y después de los algoritmos de *beamforming* en diferentes potencias del extractor. El método para calcular la SNR consiste en el caso del algoritmo superdirectivo, en calcular el filtro del *beamforming* con una señal con voz y ruido presentes para después aplicar el filtro a una señal con solo voz y sacar el valor de SPL y seguidamente volver a aplicar el filtro a una señal con ruido únicamente y volver a sacar el valor de SPL y restar ambas SPL respectivamente. En el caso del *beamforming Delay and Sum* aplicamos directamente el algoritmo a una señal con voz y obtenemos el valor de SPL y repetimos el proceso para una señal con ruido solo y de nuevo restamos los valores de SPL respectivos.

Potencia Extractor	Previo al *beamforming*	*Delay and Sum*	Superdirectivo
1	-10,5985 dB	-8,0289 dB	-5,2524 dB
6	-27,8648 dB	-24,5593 dB	-18,1529 dB
9	-34,0550 dB	-30,5401 dB	-20,6181 dB
Booster Max	-36,3662 dB	-33,1775 dB	-24,0239 dB

Tabla 4.3: Tabla de SNR del *Delay and Sum* y del superdirectivo

En la tabla se reafirman la conclusión de que el algoritmo superdirectivo ofrece mejores resultados ya que se consigue mayores valores de SNR.

4.2 Implementación y caracterización

En cuanto a la implementación de los *arrays* y la caracterización de los algoritmos de *beamforming* desarrollados, se realizaron dos *setups* de pruebas obteniendo de este modo diferentes medidas reales con las que realizar una comparativa entre el *array* circular similar al Echo Dot y el *array* rectangular implementado con hasta 16 micrófonos.

El primer *setup* se llevó a cabo con el objetivo de realizar una primera aproximación para evaluar las prestaciones de un Echo Dot en nuestro entorno ruidoso. La estructura y los resultados obtenidos con este *setup* se muestran en ***Anexo C. Primer setup de pruebas***.

4.2.1 Segundo *Setup*

El segundo *setup* se basó en una disposición de dispositivos y altavoces proporcionado por BSH, en concreto de un grupo de investigación de Múnich que comparaba los resultados de un test realizado por Amazon con el asistente de voz Alexa con un test realizado sobre un dispositivo propio de BSH. La estructura del *setup* es la que se puede ver en la **Figura 4.26**:

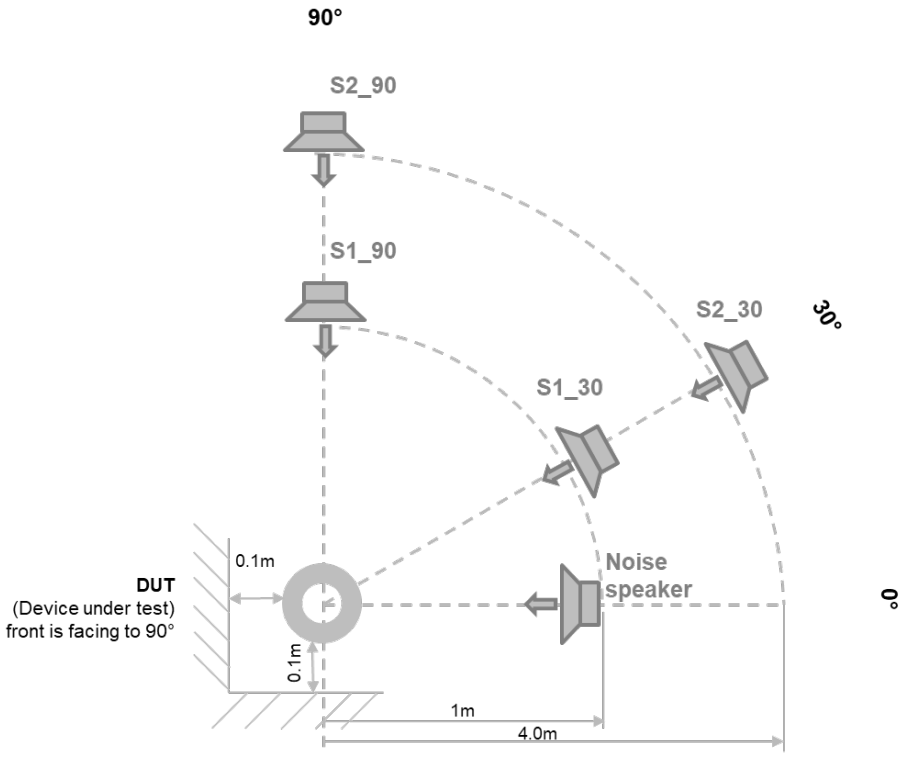


Figura 4.26: Segundo *setup* de pruebas.

Siguiendo esta estructura se han realizado las pruebas colocando un altavoz en cada una de las 4 posiciones que marca a 1 metro y 4 metros y con ángulos de 90º y 30º respecto al *array*, que en nuestro caso serán el *array* circular y el rectangular. El *array* debe estar colocado en una esquina de la habitación y la colocación en el laboratorio para las pruebas tanto del altavoz como del *array* circular y rectangular se muestra en la **Figura 4.27**. Adicionalmente se llevaron a cabo pruebas sin usar el cristal de la placa de inducción para ver la respuesta del *array* rectangular sin la atenuación del cristal, por lo que realmente se usaron 3 entornos diferentes. En todos los entornos el *noise speaker* es el extractor de humos, aunque en los casos del *array* rectangular no es posible la separación de 1 metro entre el array y la fuente de ruido, al estar el extractor incorporado en la placa de inducción.

Además, se usó un audio a una potencia fija de 70 dB para las pruebas, que se calibró usando Room Eq y el altavoz, el cual es un audio de test de Amazon y contiene la palabra "ALEXA". Las pruebas que se realizaron permitieron evaluar las prestaciones de los algoritmos de *beamforming* con el *array* rectangular y, así, poder llevar a cabo una comparación final con el *array* circular. Los 3 entornos empleados son:

Array rectangular

Array circular

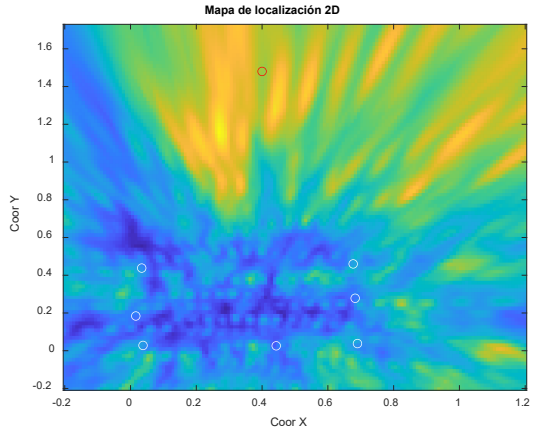
Placa sin cristal

Figura 4.27: Entornos de pruebas del segundo *setup*.

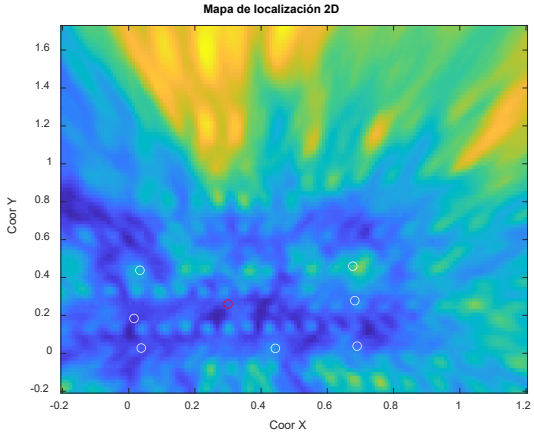
Dentro de estos 3 entornos de pruebas es importante destacar que cuando se usa el *array* rectangular con el cristal de la placa, los micrófonos están en el interior de la placa a la misma altura que los inductores. Para usar el audio que se ha comentado anteriormente como fuente de voz, se usó un altavoz situado a una altura de $1.37\ m$ que correspondería con una altura estándar de una boca de una persona.

El *array* rectangular se ha usado con 8, 12 y 16 micrófonos y para comenzar a compararlo con el *array* circular se han obtenido los mapas de localización de fuentes sonoras a la altura de la voz y a la altura del extractor, en una situación sin ruido. Las primeras pruebas se realizaron con 8 y 16 micrófonos, respectivamente, y con el cristal puesto en la placa de inducción. Los resultados se pueden ver en la **Figura 4.28**.

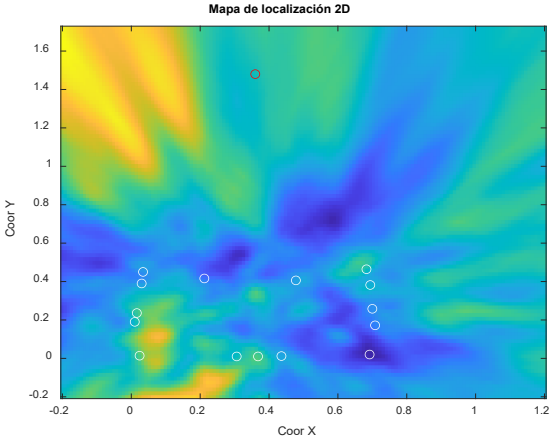
Corte del plano a la altura de la voz



Corte del plano a la altura del extractor



Corte del plano a la altura de la voz



Corte del plano a la altura del extractor

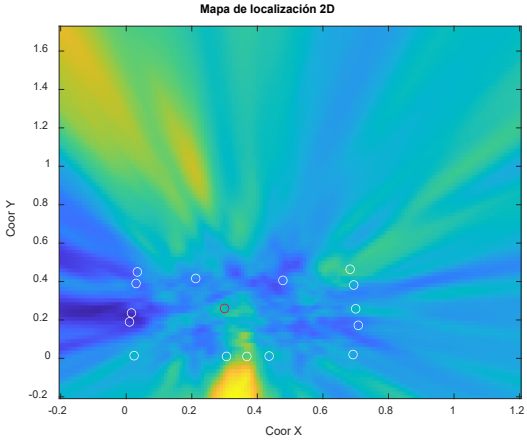
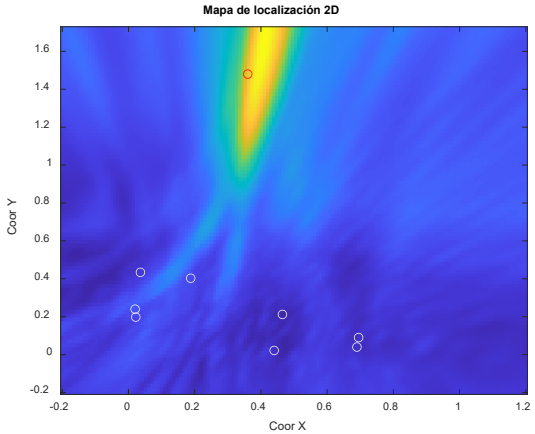


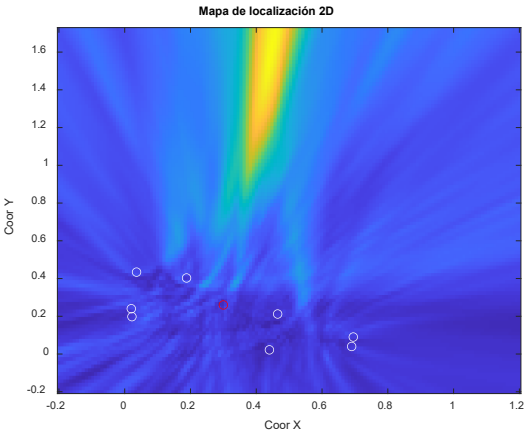
Figura 4.28: Mapas de localización con 8 y 16 micrófonos, respectivamente en la primera y segunda fila, con el cristal.

También se llevaron a cabo pruebas con 8, 12 y 16 micrófonos respectivamente, pero en este caso sin el cristal de la placa de inducción.

Corte del plano a la altura de la voz



Corte del plano a la altura del extractor



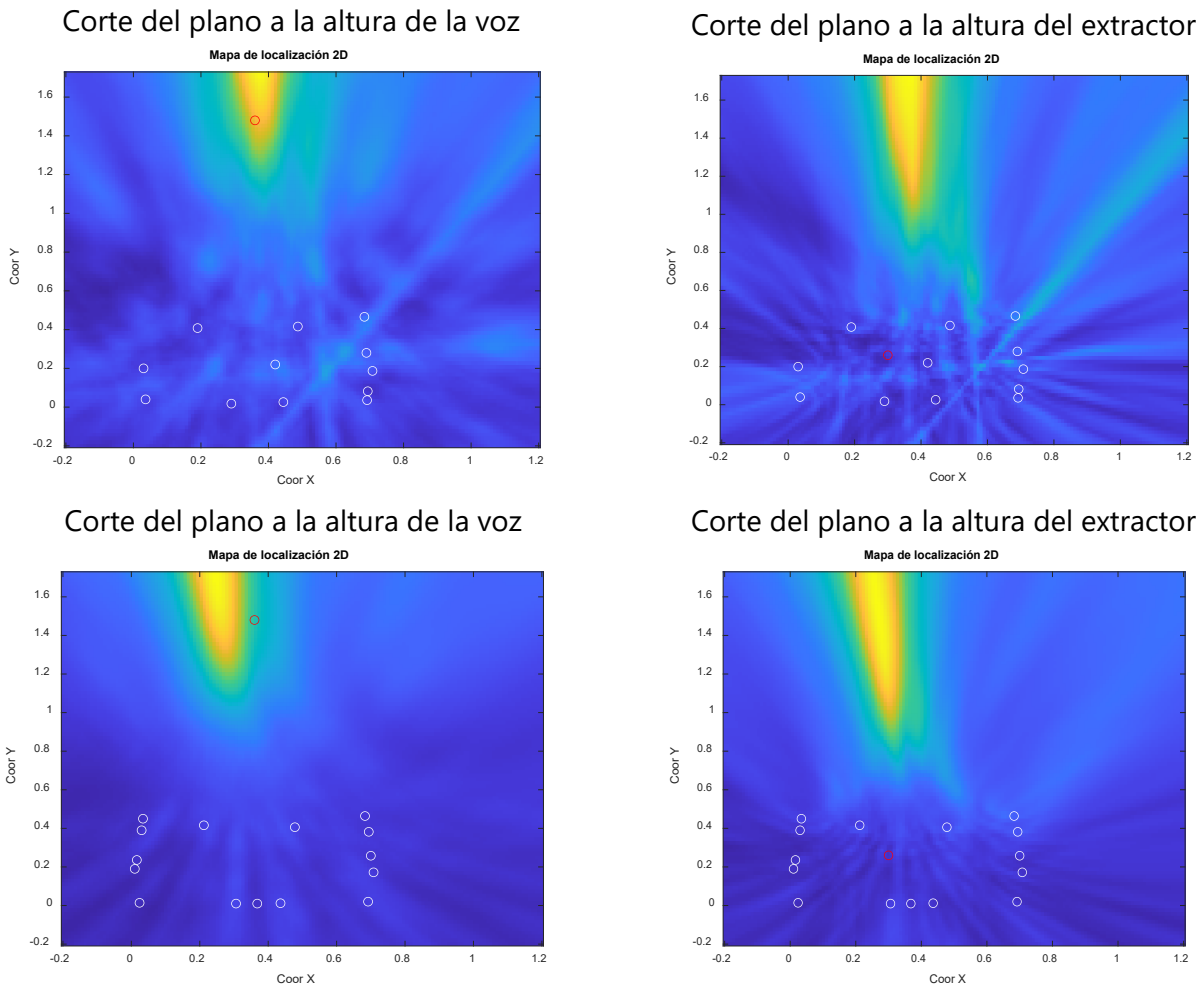
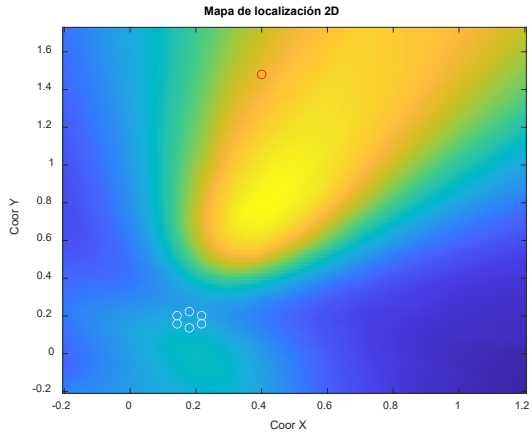


Figura 4.29: Mapas de localización con 8, 12 y 16 micrófonos, respectivamente en la primera, segunda y tercera fila, sin el cristal.

Comparando los resultados de la **Figura 4.28** con los de la **Figura 4.29** se ve claramente que el cristal de la placa es el principal inconveniente a la hora de localizar las fuentes ya que las pruebas que se hicieron sin el cristal ofrecen localizaciones de las fuentes con bastante precisión. También es importante destacar que conforme se usa un mayor número de micrófonos la localización mejora considerablemente.

Y para comparar con el *array* circular se sacaron los mapas de localización en dicho entorno de prueba. Los mapas de localización se pueden ver en la **Figura 4.30**.

Corte del plano a la altura de la voz



Corte del plano a la altura del extractor

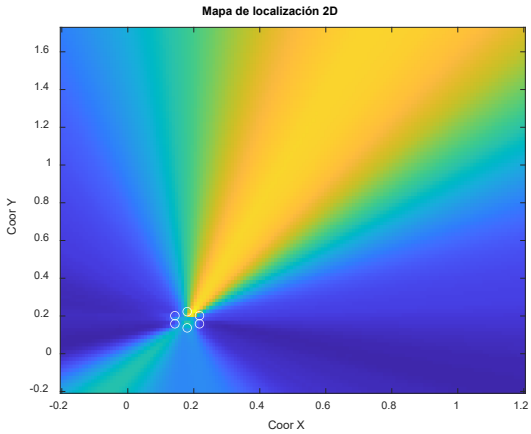


Figura 4.30: Mapas de localización con el *array* circular.

Comparando estos resultados con los anteriores, se observa que el *array* circular ofrece peores prestaciones en cuanto a la localización de la fuente. Esto se debe, especialmente, al menor tamaño del *array*.

Se hicieron una serie de pruebas para evaluar el nivel de SNR y de detección con diferentes potencias del extractor, pero en este caso sin aplicar ningún *beamforming* a las señales, aplicando el *Delay and Sum* y finalmente aplicando el superdirectivo. Estas pruebas se llevaron a cabo con el *array* rectangular de 8, 12 y 16 micrófonos y con el *array* circular en las 4 posiciones comentadas anteriormente, los resultados de las pruebas realizas a 4 metros de distancia se encuentran en el ***Anexo D. Segundo setup de pruebas***.

Primero se llevaron a cabo pruebas a 1 metro de distancia y con un ángulo de 90º entre el altavoz y el *array* rectangular de 8 micrófonos, la placa se utilizó con y sin cristal y los resultados se presentan en la **Tabla 4.4** y la **Tabla 4.5**.

Distancia = 1m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-12,65	-8,11	-5,51	0%	10%	90%
3	-22,8	-17,38	-11,01	0%	0%	0%
5	-28,06	-22,06	-13,23	0%	0%	0%
7	-32,17	-26,01	-15,38	0%	0%	0%
9	-36,11	-30,61	-18,63	0%	0%	0%
Booster	-38,14	-32,67	-21,15	0%	0%	0%
Booster max	-39,61	-34,26	-22,73	0%	0%	0%

Tabla 4.4: Niveles de SNR y detección con 8 micrófonos con el cristal. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 1m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	4,04	7,46	24,28	100%	100%	100%
3	-0,6	0,21	14,28	100%	100%	100%
5	-4,95	-4,03	10,25	100%	100%	100%
7	-9,57	-9,08	5,21	0%	0%	100%
9	-13,76	-13,43	0,72	0%	0%	100%
Booster	-15,7	-15,4	-0,91	0%	0%	100%
Booster max	-15,85	-15,58	-1,41	0%	0%	100%

Tabla 4.5: Niveles de SNR y detección con 8 micrófonos sin el cristal. DS: *Delay and Sum*. SP: superdirectivo.

Con 12 micrófonos con y sin el cristal se obtienen los resultados mostrados en la **Tabla 4.6** y la **Tabla 4.7**.

Distancia = 1m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-13,12	-7,85	-0,24	0%	20%	100%
3	-22,39	-16,21	-7,72	0%	0%	20%
5	-27,58	-21,39	-13,32	0%	0%	0%
7	-32,24	-25,75	-12,71	0%	0%	0%
9	-36	-29,52	-13,58	0%	0%	0%
Booster max	-36,27	-31,78	-22,19	0%	0%	0%

Tabla 4.6: Niveles de SNR y detección con 12 micrófonos con el cristal. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 1m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	0,69	3,08	17,45	100%	100%	100%
3	-7,85	-5,51	9,73	20%	100%	100%
5	-13,11	-11,03	5,51	0%	0%	100%
7	-17,55	-15,45	0,01	0%	0%	100%

Tabla 4.7: Niveles de SNR y detección con 12 micrófonos sin el cristal. DS: *Delay and Sum*. SP: superdirectivo.

Con 16 micrófonos con y sin el cristal tenemos los resultados que se pueden ver en la **Tabla 4.8** y la **Tabla 4.9**.

Distancia = 1m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-13,12	-7,43	1,29	0%	20%	100%
3	-22,39	-16	-7,37	0%	0%	20%
5	-27,58	-21,24	-12,45	0%	0%	0%
7	-32,24	-26,17	-12,46	0%	0%	0%
9	-36	-29,71	-13,63	0%	0%	0%
Booster max	-36,27	-33,72	-19,02	0%	0%	0%

Tabla 4.8: Niveles de SNR y detección con 16 micrófonos con el cristal. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 1m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	0,69	3,05	18,03	100%	100%	100%
3	-7,85	-5,69	10,49	20%	100%	100%
5	-13,11	-11,3	4,7	0%	0%	100%
7	-17,55	-15,68	-0,86	0%	0%	100%

Tabla 4.9: Niveles de SNR y detección con 16 micrófonos sin el cristal. DS: *Delay and Sum*. SP: superdirectivo.

En todos estos resultados volvemos a observar el efecto negativo que tiene el cristal, obteniendo un nivel de detección del 100% sin el cristal con el algoritmo superdirectivo, mientras que con el cristal únicamente se logra la detección en las potencias bajas del extractor mayoritariamente con el algoritmo superdirectivo.

Al aumentar el número de micrófonos se produce una mejora de los niveles de SNR tras procesar, sin embargo, en las detecciones la mejora no es tan considerable. Hay que destacar que tras estos resultados se ha observado que la detección de la palabra "ALEXA" se realiza a partir de aproximadamente $-5\ dB$ de SNR del extractor.

Y para comparar con el *array* circular se han calculado los niveles de SNR y de detección con dicho *array* también en la posición de 1 metro de distancia y 90º entre el altavoz y el *array*.

Distancia = 1m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	14,32	15,55	17,87	100%	100%	100%
3	10,3	11,27	16,45	100%	100%	100%
5	5,79	6,74	11,32	100%	100%	100%
7	2,52	3,52	11,1	100%	100%	100%
9	-2,12	1,16	2,29	100%	100%	100%
Booster	-4,55	-3,53	2,02	100%	100%	100%
Booster max	-5,28	-4,17	8,96	100%	100%	100%

Tabla 4.10: Niveles de SNR y detección con el *array* circular de 6 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Claramente el *array* circular ofrece mejores prestaciones en cuanto al nivel de SNR y de detección ya que prácticamente logra el 100% de detección en todas las potencias del extractor de la placa de inducción.

En segundo lugar, se realizaron las mismas pruebas a 1 metro de distancia, con un ángulo de 30º y todas con el cristal de placa. Los resultados obtenidos con 8, 12 y 16 micrófonos se muestran en la **Tabla 4.11**, la **Tabla 4.12** y la **Tabla 4.13**.

Distancia = 1m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-7,25	-4,79	-0,09	20%	100%	100%
3	-16,33	-13,53	-6,13	0%	0%	30%
5	-21,54	-18,52	-11,35	0%	0%	0%
7	-25,93	-22,74	-15,47	0%	0%	0%
9	-29,92	-26,24	-18,44	0%	0%	0%
Booster max	-31,89	-28,25	-19,82	0%	0%	0%

Tabla 4.11: Niveles de SNR y detección con 8 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 1m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-7,25	-5,02	1,26	20%	100%	100%
3	-16,33	-13,79	-6,14	0%	0%	30%
5	-21,54	-18,72	-11,43	0%	0%	0%
7	-25,93	-22,88	-15,32	0%	0%	0%
9	-29,92	-26,44	-18,11	0%	0%	0%
Booster max	-31,89	-28,48	-29,82	0%	0%	0%

Tabla 4.12: Niveles de SNR y detección con 12 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 1m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-7,25	-6,75	1,35	20%	30%	100%
3	-16,33	-15,54	-6,23	0%	0%	30%
5	-21,54	-20,58	-11,41	0%	0%	0%
7	-25,93	-24,91	-15,47	0%	0%	0%
9	-29,92	-28,55	-18,03	0%	0%	0%
Booster max	-31,89	-30,47	-20,01	0%	0%	0%

Tabla 4.13: Niveles de SNR y detección con 16 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Por otro lado, los resultados que se consiguen con el *array* circular en esta posición son:

Distancia = 1m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	4,75	6,79	6,19	100%	100%	100%
3	4,24	5,88	7	100%	100%	100%
5	2,93	4,53	8,59	100%	100%	100%
7	0,7	2,22	10,18	100%	100%	100%
9	-3,17	-1,73	11,35	100%	100%	100%
Booster max	-4,74	-3,23	10,08	100%	100%	100%

Tabla 4.14: Niveles de SNR y detección con el *array* circular de 6 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

En esta posición debido al cambio de ángulo se obtienen niveles de SNR superficialmente superiores en el *array* rectangular mientras que en el *array* circular se produce el efecto contrario ya que en este caso el *array* está en una posición más cercana al extractor por lo que la dirección de la señal de ruido con la de la voz se distingue peor.

Adicionalmente, se han llevado a cabo algunas pruebas con el filtro de Wiener LMS, a continuación, se muestran los niveles de SNR y de detección para 8 y 16 micrófonos con el *array* rectangular en la **Tabla 4.15** y la **Tabla 4.16**, en los dos casos la placa de inducción se ha usado con el cristal puesto. Ambas tablas corresponden con la primera posición de 1 metro y 90º.

Distancia = 1m			Angulo = 90	
	SNR (dB)		Detección	
Extractor	Sin procesado	LMS	Sin procesado	LMS
1	-12,65	-11,94	0%	0%
3	-22,8	-22,05	0%	0%
5	-28,06	-26,59	0%	0%
7	-32,17	-30,32	0%	0%
9	-36,11	-33,48	0%	0%
Booster	-38,14	-35,16	0%	0%
Booster max	-39,61	-36,78	0%	0%

Tabla 4.15: Niveles de SNR y detección con el *array* rectangular de 8 micrófonos con LMS.

Distancia = 1m			Angulo = 90	
	SNR (dB)		Detección	
Extractor	Sin procesado	LMS	Sin procesado	LMS
1	-13,12	-12,61	0%	0%
3	-22,39	-20,42	0%	0%
5	-27,58	-23,8	0%	0%
7	-32,24	-24,22	0%	0%

Tabla 4.16: Niveles de SNR y detección con el *array* rectangular de 16 micrófonos con LMS.

Concluyendo, con este tipo de filtrado vemos que se consigue una mejora de la SNR de alrededor de 3dB y que junto con un *beamforming* superdirectivo se podrían mejorar los resultados de detección obtenidos anteriormente.

Y, por último, se ha llevado a cabo una comparativa final de los resultados obtenidos en la primera posición de 1 metro de distancia y 90º para el *array* circular y para el *array* rectangular de 8 micrófonos cuando no se usó el cristal de la placa de inducción.

Distancia = 1m			Angulo = 90	
	6 mic SNR (dB)		8 mic SNR (dB)	
Extractor	Sin procesado	SP	Sin procesado	SP
1	14,32	17,87	4,04	24,28
3	10,3	16,45	-0,6	14,28
5	5,79	11,32	-4,95	10,25
7	2,52	11,1	-9,57	5,21
9	-2,12	2,29	-13,76	0,72
Booster	-4,55	2,02	-15,7	-0,91

Tabla 4.17: Comparativa final entre el *array* circular y el *array* rectangular de 8 micrófonos y sin el cristal. SP: superdirectivo.

En esta comparativa seguimos viendo que el *array* circular ofrece mejores resultados, aunque la diferencia con respecto a aplicar el algoritmo superdirectivo sobre el *array* rectangular sin el cristal es pequeña, teniendo en cuenta que los micrófonos de dicho *array* están situados muchos más cerca del extractor.

Capítulo 5. Resultados y conclusiones

El objetivo principal de este trabajo era estudiar y comparar las prestaciones ofrecidas por dos *arrays* de micrófonos de diferentes geometrías para valorar la posibilidad de incorporarlos a una placa de inducción con el extractor integrado. Tras la simulación y las pruebas realizas han quedado demostradas las ventajas e inconvenientes de cada uno de ellos. Otro objetivo importante del proyecto era el desarrollo de diferentes algoritmos de filtrado y finalmente se han conseguido implementar hasta 4 algoritmos.

Tal y como se planteaba inicialmente, el cristal de la placa de inducción es el elemento que mayores problemas nos ha presentado, ya que la atenuación que provoca en las señales que llegan a los micrófonos del interior de la placa es muy alta. Este hecho se observa claramente en la diferencia que hay entre los mapas de localización con el cristal, en los que no es capaz de localizar la fuente de voz, mientras que en los mapas sin el cristal el *array* rectangular localiza perfectamente la fuente. Con los niveles de SNR, la diferencia puede llegar a ser incluso superior a los 10dB.

El número de micrófonos y el tamaño del *array* también influye en las prestaciones, ya que cuanto mayor es su número y el tamaño, mejores niveles de SNR y detección se logran, así como una localización más exacta de las fuentes. Especialmente, al reducirse la distancia entre los micrófonos en el *array* rectangular, por el uso de un número mayor de micrófonos, se consigue una mejor resolución espacial y a su vez se reducen los problemas de *aliasing* en las frecuencias altas mientras que, por el contrario, el *array* circular, aun teniendo una distancia entre micrófonos reducida, no es capaz de obtener una buena resolución espacial debido a que las dimensiones del *array* son mucho menores, aunque no tiene problemas de *aliasing*. Pese a esto, la diferencia que se produce entre 12 y 16 micrófonos es tan pequeña que se ha llegado a la conclusión de que con 12 micrófonos sería suficiente al conseguir un buen compromiso entre el coste de implementación y los resultados ofrecidos por el *array*.

En cuanto a los algoritmos de *beamforming*, ha quedado claro que el superdirectivo tiene mayor capacidad de filtrado y por tanto ofrece un mejor funcionamiento del sistema, ya que en muchos casos es capaz de conseguir una detección del 100% mientras que el *Delay and Sum* solo obtiene el 100% de detección cuando la potencia del extractor es la más baja posible.

Es importante destacar la comparativa mostrada en la **Tabla 4.17**, donde se puede observar cómo el algoritmo superdirectivo logra una ganancia en la SNR de hasta $15\ dB$ en el *array* rectangular sin el cristal de la cocina frente a los $6\ dB$ que logra en el array circular. Sin embargo, estos $9\ dB$ de ganancia adicional que se logran por usar un *array* de mayores dimensiones no son suficientes para compensar el hecho de que la situación inicial del *array* rectangular (incluso sin el cristal) es $11\ dB$ inferior que en el caso del *array* circular externo ya que sus micrófonos se encuentran mucho más cerca de la fuente principal de ruido: el extractor de humos. Por tanto, podemos concluir que,

en términos de SNR, el uso de un *array* externo es preferible incluso si sus dimensiones son menores; aunque, a la vista de los resultados de las **Tablas 4.5, 4.7 y 4.9**, en las que se muestra que el *beamformer* superdirectivo logra tasas de detección del 100% en el *array* rectangular sin cristal, podría ser interesante el uso de un *array* integrado si fuera posible realizar orificios en el cristal de la cocina en la posición de los micrófonos.

Finalmente, como líneas futuras a seguir, quedaría abierta la posibilidad de unir el algoritmo de *beamforming* superdirectivo con el filtro de Wiener LMS para así poder combinar las ventajas que ambos ofrecen. De igual modo, sería necesaria una implementación del filtrado y la detección en tiempo real, para lo que sería interesante usar el algoritmo SRP-PHAT para localizar la dirección en la que está la fuente de voz, después aplicar el filtrado espacial y frecuencial para obtener una buena detección y, una vez detectada la palabra de despertar, bajar la potencia del extractor para que el usuario diera la orden que quisiera que ejecutara la placa de inducción. Para poder desarrollar esta idea sería interesante que la placa de inducción se comunicara con el *array* de micrófonos tanto si fuera externo como si no lo fuera, y para ello se podría implementar una *skill* de Alexa, ya que es el asistente de voz que se ha usado durante el proyecto. En el caso de que se quisiera usar un *array* de micrófonos dentro de la placa de inducción, sería imprescindible realizar una serie de agujeros por los que pudiera entrar el sonido y habría que cubrirlo con algún tipo de membrana impermeable para evitar que se introdujese cualquier partícula o liquido dentro de la placa.

Referencias

[1] M. Molina Gracia, «Trabajo Fin de Grado Análisis y localización de fuentes de ruido en una placa de inducción.», Universidad de Zaragoza, 2019.

[2] F. Antonacci, A. Canclini, y A. Sarti, «Sound Analysis Lecture Notes (Draft)», p. 201.

[3] V. Perrot, M. Polichetti, F. Varray, y D. Garcia, «So you think you can DAS? A viewpoint on delay-and-sum beamforming», *Ultrasonics*, vol. 111, p. 106309, mar. 2021, doi: 10.1016/j.ultras.2020.106309.

[4] M. Brandstein y D. Ward, Eds., *Microphone Arrays: Signal Processing Techniques and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. doi: 10.1007/978-3-662-04619-7.

[5] J. A. Johnson, M. Karaman, y B. T. Khuri-Yakub, «Coherent-array imaging using phased subarrays. Part I: basic principles», *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 52, n.º 1, pp. 37-50, ene. 2005, doi: 10.1109/TUFFC.2005.1397349.

[6] Jingdong Chen, J. Benesty, Yiteng Huang, y S. Doclo, «New insights into the noise reduction Wiener filter», *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, n.º 4, pp. 1218-1234, jul. 2006, doi: 10.1109/TSA.2005.860851.

[7] F. J. López y J. M. Salamanca, «ALGORITMOS LMS DE FILTRADO ADAPTATIVO PARA CANCELACIÓN DE ECO ACÚSTICO EN SISTEMAS DE TELECOMUNICACIONES», p. 8.

[8] Siemens, «Datasheet Placa Induccón (EX875LX67E)». https://media3.bsh-group.com/Documents/specsheet/es-ES/EX875LX67E.pdf (accedido jun. 09, 2021).

[9] Siemens, «Manual Usuario Placa Induccón EX875LX67E». https://media3.bsh-group.com/Documents/9001469420_C.pdf (accedido jun. 09, 2021).

[10] R. Gracia Escorihuela, «Trabajo Fin de Master Estudio de la viabilidad de incorporar una ayuda técnica controlada por voz en una cocina de inducción», Universidad de Zaragoza, 2020.

[11] STMicroeletronics, «MEMS audio sensor omnidirectional digital microphone», p. 17.

[12] Adafruit, «Adafruit PDM Microphone Breakout», *Adafruit Learning System*. https://learn.adafruit.com/adafruit-pdm-microphone-breakout/downloads (accedido jun. 09, 2021).

[13] MiniDSP, «Product Brief-MCHStreamer.pdf». https://www.minidsp.com/images/documents/Product%20Brief-MCHStreamer.pdf (accedido jun. 09, 2021).

[14] MiniDSP, «MCHStreamer User Manual.pdf». https://www.minidsp.com/images/documents/MCHStreamer%20User%20Manual.pdf (accedido jun. 09, 2021).

[15] MiniDSP, «Product Brief UMA-8», oct. 19, 2017. https://www.minidsp.com/products/usb-audio-interface/uma-8-16-usb-mic-array/uma-8-sp-detail

[16] «MATLAB - El lenguaje del cálculo técnico». https://es.mathworks.com/products/matlab.html (accedido jun. 12, 2021).

[17] «Register a Product | Alexa Voice Service». https://developer.amazon.com/es/docs/alexa-voice-service/register-a-product.html (accedido may 24, 2019).

[18] J. H. DiBiase, H. F. Silverman, y M. S. Brandstein, «Robust Localization in Reverberant Rooms», en *Microphone Arrays*, M. Brandstein y D. Ward, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 157-180. doi: 10.1007/978-3-662-04619-7_8.
[19] Cha Zhang, D. Florencio, y Zhengyou Zhang, «Why does PHAT work well in lownoise, reverberative environments?», en *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, mar. 2008, pp. 2565-2568. doi: 10.1109/ICASSP.2008.4518172.
[20] G. Ballou, Ed., *Handbook for sound engineers*, 4th ed. Amsterdam ; Boston: Focal Press, 2008.
[21] J. de Mingo y Í. Salinas, «Tema 3. Fundamentos de transmisión y recepción acústica», *Propag. Medios Transm. Univ. Zaragoza*, p. 79.

Anexo A. Conceptos teóricos

A.1. Localización de fuentes sonoras

En cuanto a la localización de fuentes sonoras, se ha usado el algoritmo SRP-PHAT, el cual combina las ventajas de los métodos SRP junto con la transformación de fase que proponen los métodos basados en TDOA [18]. De este modo, se trata de una técnica robusta frente a las reverberaciones y el ruido con un coste computacional relativamente bajo. Antes de describir el algoritmo SRP-PHAT, será necesario desarrollar una serie de conceptos teóricos.

- La función GCC y la transformación de fase PHAT

  Para un par de micrófonos $n, m = 1,2$, su TDOA, τ_{12}, resulta:

$$\tau_{12} = \tau_2 - \tau_1 \tag{2.31}$$

  Aplicando esta definición, las señales recibidas quedan así:

$$\begin{aligned} x_1(t) &= \frac{1}{r_1} s(t-\tau_1) * g_1(\boldsymbol{q_s}, t) + v_1(t) \\ x_2(t) &= \frac{1}{r_2} s(t-\tau_1-\tau_{12}) * g_2(\boldsymbol{q_s}, t) + v_2(t) \end{aligned} \tag{2.32}$$

  Si ambas son similares, entonces (2.32) muestra que hay una versión escalada de $s(t-\tau_1)$ en la señal del micrófono 1 y una versión de $s(t-\tau_1)$ desplazada en el tiempo y escalada en la señal del micrófono 2. Por lo tanto, la correlación cruzada de las dos señales debería mostrar un pico en el retardo donde coincidan las versiones modificadas de $s(t)$, el cual corresponde a la TDOA, τ_{12}. La correlación cruzada de las señales se define como:

$$c_{12}(\tau) = \int_{-\infty}^{+\infty} x_1(t)\, x_2(t+\tau) dt \tag{2.33}$$

  La función GCC, $R_{12}(\tau)$, es la correlación cruzada de dos versiones filtradas de $x_1(t)$ y $x_2(t)$, cuyos filtros son $G_1(\omega)$ y $G_2(\omega)$, y puede expresarse en el dominio frecuencial de este modo:

$$R_{12}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \big(G_1(\omega) X_1(\omega)\big) \big(G_2(\omega) X_2(\omega)\big)^* e^{j\omega\tau} d\omega \tag{2.34}$$

  Utilizando la transformación $\psi_{12} \equiv G_1(\omega) G_2(\omega)^*$, la función GCC puede reescribirse de la siguiente manera:

$$R_{12}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \psi_{12}(\omega)\, X_1(\omega)X_2(\omega)^* e^{j\omega\tau} d\omega \qquad (2.35)$$

El TDOA entre los dos micrófonos será el máximo global de esta función:

$$\hat{\tau}_{12} = \arg\max_{\tau} R_{12}(\tau)\,. \qquad (2.36)$$

En general, $R_{12}(\tau)$ tendrá múltiples máximos locales que pueden ocultar el verdadero máximo del TDOA, produciéndose de este modo una estimación incorrecta. En entornos reales, para enfatizar el verdadero valor de la GCC en el verdadero máximo del TDOA, se utiliza la transformación de fase PHAT, que se expresa así:

$$\psi_{12}(\omega) \equiv \frac{1}{|X_1(\omega)X_2^*(\omega)|} \qquad (2.37)$$

La transformación de fase hace que todas las frecuencias de las señales tengan el mismo módulo, diferenciándose solo en la fase, que es donde se encuentra la información de τ_{12}. Esta transformación ha demostrado un mejor rendimiento que el resto de los métodos en condiciones de ruido y reverberación [19].

- Localización de fuente basada en SRP

Los *beamformers* de retardo y suma alinean y suman todas las señales recibidas en los micrófonos, tal y como se ha explicado en la sección *Delay And Sum*, y se definen como:

$$y(t, \boldsymbol{q}) = \sum_{n=1}^{N} x_n(t + \Delta_n) \qquad (2.38)$$

donde Δ_n son los retardos en las direcciones a las que están dirigidos cada uno de los micrófonos. Dichas direcciones están enfocadas hacia la dirección $\boldsymbol{q}$, y compensan los retardos del camino directo de las señales recibidas en cada micrófono. Para hacer que todas las operaciones de desplazamientos sean causales, se usa uno de los micrófonos de la agrupación como referencia. Esto permite dirigir el *beamformer* hacia las direcciones que nos interesen sin tener una localización explicita de la fuente.

En el caso ideal de tener un entorno sin ruido aditivo ni efectos de canal, la salida del *beamformer* de retardo y suma será una versión potencialmente retardada y escalada de la señal deseada. En la práctica, los efectos de canal no son triviales y el ruido aditivo suele estar presente. El problema de este tipo de *beamforming* es que el nivel de reducción del ruido y reverberación es mínimo y difícil de analizar, por lo que tienen mayor utilizad otros métodos de *beamforming*

como los de filtrado y suma, ya que estos aplican un filtrado adaptativo a las señales de los micrófonos antes de que estén alineadas y sumadas.

La salida de un *beamformer* de filtrado y suma se puede definir en el dominio de la frecuencia como:

$$Y(\omega, \boldsymbol{q}) = \sum_{n=1}^{N} G_n(\omega) X_n(\omega) e^{j\omega \Delta_n} \tag{2.39}$$

donde $X_n(\omega)$ y $G_n(\omega)$ son las Transformadas de Fourier de la señal del micrófono n-ésimo y su filtro respectivamente. De manera análoga al *beamforming* en el dominio del tiempo, las señales recibidas se alinean en fase a partir de los retardos en las direcciones enfocadas a la localización de la fuente $\boldsymbol{q}$. Así mismo, la suma del filtrado permite que se puedan compensar ciertos efectos perjudiciales del entorno y del canal. Elegir el filtro apropiado depende de varios factores, incluida la naturaleza de la fuente y el tipo de ruido y reverberación presentes.

Dirigiendo el *beamforming* hacia ciertos puntos espaciales de interés y evaluando la salida, normalmente su potencia, habrá un máximo en el SRP si la dirección a la que se ha apuntado el *beamforming* coincide con la fuente de sonido. El problema está cuando se producen varios máximos, por ejemplo, debido a reflexiones, que pueden generar localizaciones incorrectas. El SRP para un posible punto puede expresarse como:

$$P(\boldsymbol{q}) = \int_{-\infty}^{+\infty} |\, Y(\omega)|^2 d\omega \tag{2.40}$$

y la estimación de la localización se encuentra a partir de:

$$\widehat{\boldsymbol{q}}_{\boldsymbol{s}} = \underset{\boldsymbol{q}}{\operatorname{argmax}}\, P(\boldsymbol{q}). \tag{2.41}$$

- El algoritmo SRP-PHAT

El objetivo del algoritmo SRP-PHAT es combinar las ventajas del *beamformer* dirigido y de la transformación de fase PHAT y lograr también una implementación con un menor coste computacional.

El SRP del *beamformer* de filtrado y suma se presenta como:

$$P(\boldsymbol{q}) = \sum_{l=1}^{N} \sum_{k=1}^{N} \int_{-\infty}^{\infty} \psi_{lk}(\omega)\, X_l(\omega) X_k^*(\omega) e^{j\omega(\Delta_k - \Delta_l)} d\omega \tag{2.42}$$

donde la transformación $\psi_{lk}(\omega) = G_l(\omega) G_k^*(\omega)$ es similar a la expresada en (2.35) y suele emplearse la transformación PHAT:

$$\psi_{lk}(\omega) = \frac{1}{\left|X_l(\omega)X_k^*(\omega)\right|} \tag{2.43}$$

Por otro lado, se puede demostrar que (2.42) es equivalente a la suma de las GCC de todas las combinaciones de parejas de N micrófonos en el dominio del tiempo como:

$$P(\boldsymbol{q}) = 2\pi \sum_{l=1}^{N} \sum_{k=1}^{N} R_{lk}(\Delta_k - \Delta_l)\,. \tag{2.44}$$

Se trata de la suma de todas las posibles combinaciones de parejas de GCC que están desplazadas por las diferencias en las direcciones de los retardos. Para conseguir la localización de la fuente, se busca un máximo en $P(\boldsymbol{q})$ en un conjunto de posibles localizaciones de la fuente. El SRP-PHAT disminuye la importancia de máximos extraños y mejora sustantivamente la resolución del verdadero máximo. Estas características hacen que haya una menor sensibilidad al ruido y reverberaciones y que las estimaciones de localización sean más precisas que las localizaciones conseguidas por métodos como los comentados anteriormente. Adicionalmente, esto se logra mediante un intervalo de análisis muy corto, consiguiendo así una reducción del coste computacional y reduciendo el tiempo en el que la fuente debe permanecer estacionaria.

A.2. Ruido acústico y medidas de nivel de sonido

El término ruido se utiliza comúnmente para designar las señales no deseadas que aparecen en los sistemas de comunicaciones y sobre las que no tenemos ningún control. Se trata de una visión muy general del término y en nuestro caso es necesario hacer una clasificación menos general para poder abordar todas las perturbaciones existentes en el ambiente que se estudia. Adicionalmente, las señales se verán afectadas por diferentes tipos de distorsiones, las cuales son modificaciones de las señales producidas.

Dentro de todas las perturbaciones que puede haber en el entorno acústico estudiado, las principales son las siguientes: ruido aditivo, el cual será todo aquel ruido procedente las diferentes fuentes que hay en el entorno; reverberación, producida por la propagación multitrayecto de las señales que se dan en los entornos acústicos cerrados o semi cerrados; y, por último, el eco, que se trata de una versión de la señal que se produce como la reverberación pero con un retardo mayor que esta.

Debido a lo comentado anteriormente, será útil el uso de ciertas medidas acústicas como, por ejemplo, el nivel de presión sonora, la potencia acústica y la relación señal a ruido. En un evento sonoro el nivel de volumen que se percibe se refiere al nivel sonoro que hay, el cual se mide en presión o potencia sonora expresada en decibelios [20].

A la hora de medir presiones sonoras, hay muchos modos de hacer que los resultados sean relevantes para la percepción humana. Dado que el rango dinámico de presiones eficaces de los sonidos audibles es muy amplio, y para poder usarlo fácilmente, se introduce el término de nivel de presión sonora (*Sound Pressure Level*, SPL), como se explica en [21]. Este representa en formato logarítmico, en decibelios (dBs), el valor de presión eficaz respecto a un nivel de presión de referencia (el mínimo nivel audible).

$$SPL = 20\log_{10}\left(\frac{p_{ef}}{p_{ref}}\right) \tag{2.45}$$

Cuyo mínimo audible es: $p_{ref} = 2 \cdot 10^{-5}\, N/m^2 = 2 \cdot 10^{-5}\, Pascales$

Otra medida importante en términos acústicos es la relación señal a ruido (*Signal-to-noise* ratio, SNR), esta medida se define como la proporción existente entre la potencia de la señal que se transmite y la potencia de ruido que la corrompe. De igual modo que la SPL se mide en decibelios normalmente.

$$SNR = 10\log_{10}\left(\frac{S}{N}\right) \tag{2.46}$$

Por último, cabe destacar que el rango dinámico audible para el oído humano, en amplitud y en frecuencia, es:

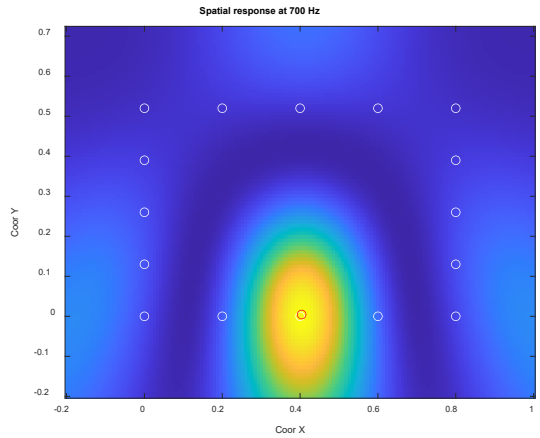
$$0dB_{SPL}\ a\ 120dB_{SPL}$$
$$16\ Hz\ a\ 20\ KHz$$

A pesar de que lo que oímos es la presión sonora, ésta es provocada por la potencia acústica emitida por la fuente de ruido, haciendo que su estudio sea igualmente interesante. Aunque se pueden considerar similares, existen diferencias importantes entre ellas, por ejemplo, la presión sonora depende de la distancia a la fuente, del entorno acústico, del tamaño de la habitación y de la absorción acústica de las superficies de dicha habitación; en cambio, la potencia acústica es independiente del entorno y depende únicamente de la fuente.

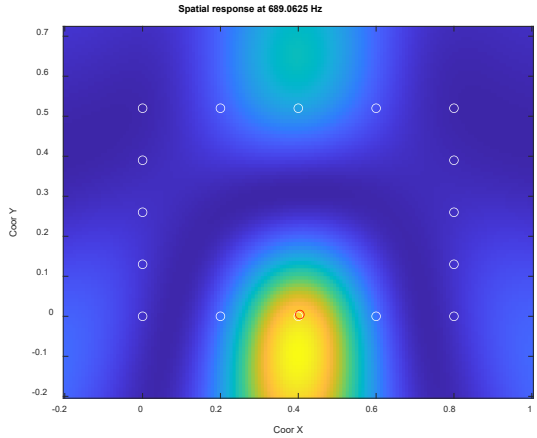
Anexo B. Simulación acústica

La respuesta espacial en diferentes frecuencias, desde $350\ Hz$ hasta $4100\ Hz$ aproximadamente, tanto a la altura de la voz como del extractor, todo ello con una potencia del extractor de 9, para los dos algoritmos de *beamforming* se muestra a continuación:

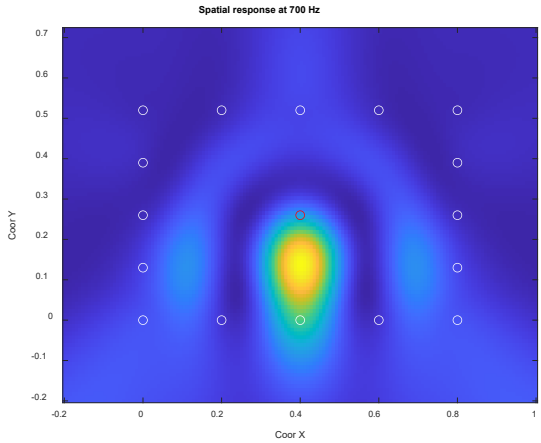
Delay and Sum Voz



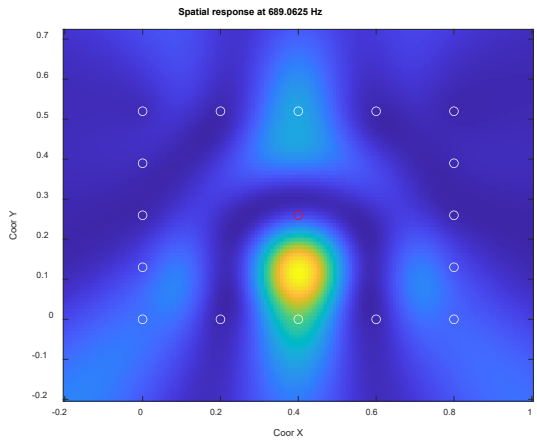
Superdirectivo Voz



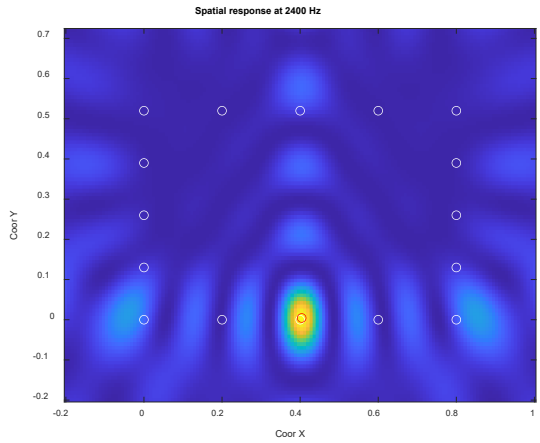
Delay and Sum Extractor



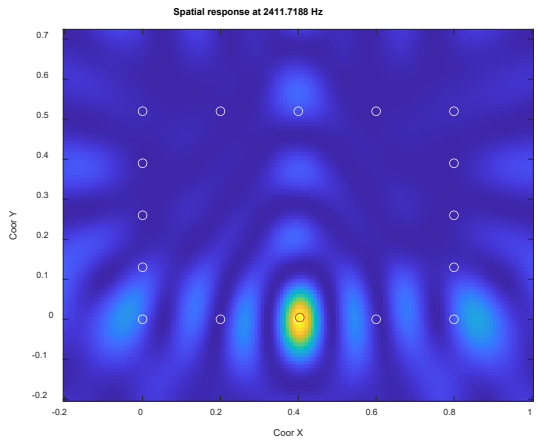
Superdirectivo Extractor



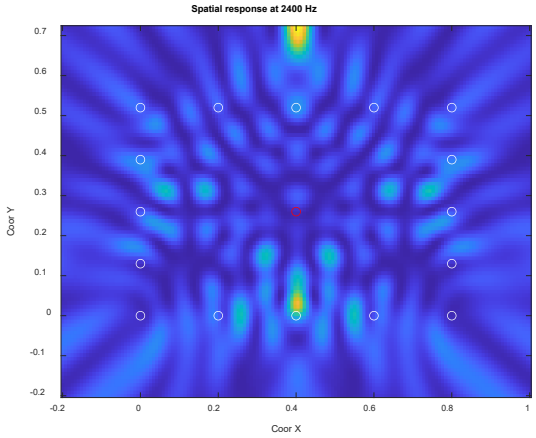
Delay and Sum Voz



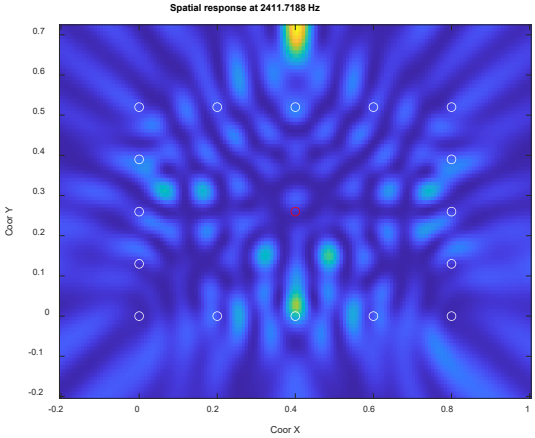
Superdirectivo Voz



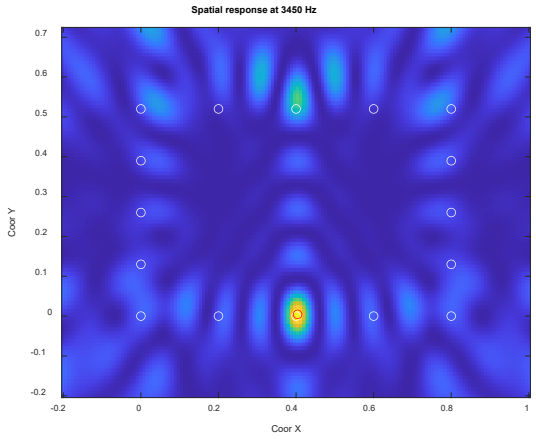
Delay and Sum Extractor



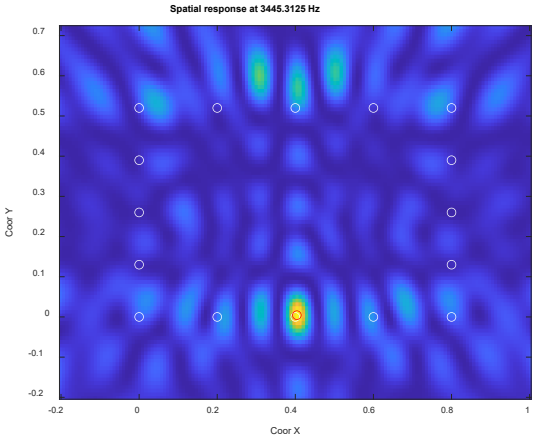
Superdirectivo Extractor



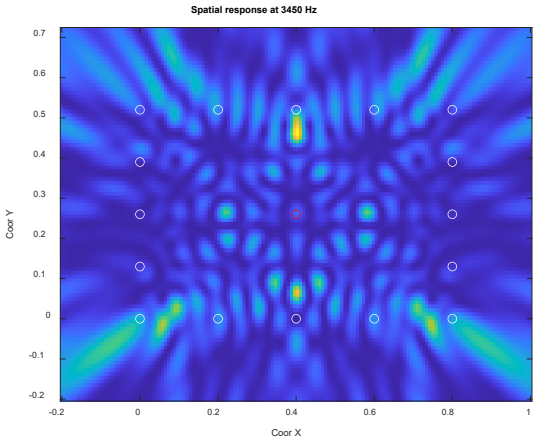
Delay and Sum Voz



Superdirectivo Voz



Delay and Sum Extractor



Superdirectivo Extractor

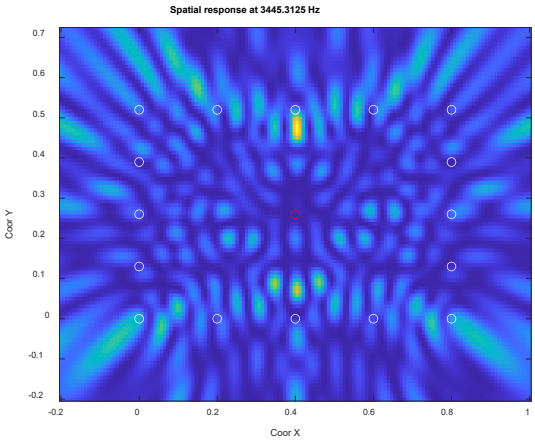


Figura B.31: Respuesta espacial en diferentes frecuencias del *Delay and Sum* y del superdirectivo.

Anexo C. Primer *setup* de pruebas

En este *setup* se hicieron medidas con dos posiciones diferentes del *array* circular, A y B, y también se usaron 7 posiciones en las que una persona decía la palabra de despertar del sistema Alexa, por lo que de este modo la potencia de la voz no era constante.

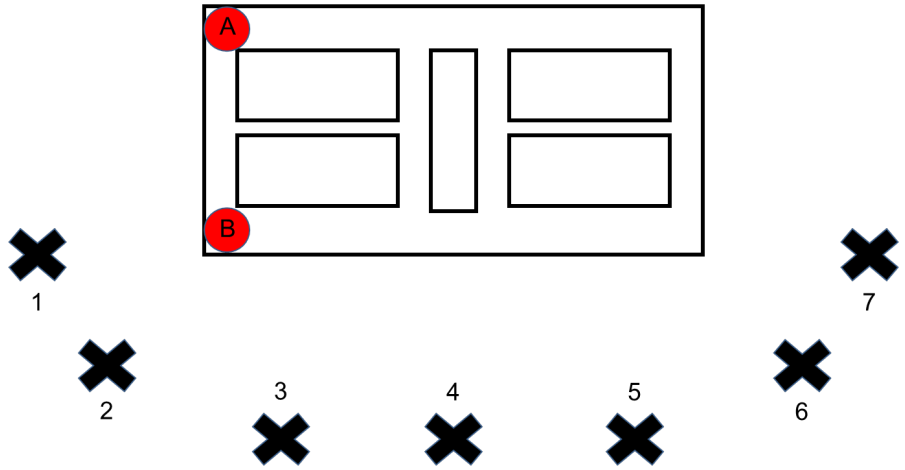


Figura C.32: Primer *setup* de pruebas.

En cada una de las posiciones del *setup* se hizo un barrido con las potencias del extractor de la placa de inducción y se midió tanto el nivel de SPL como la detección de la palabra de despertar Alexa.

La evaluación del nivel de SPL, en la posición A y B del *array* y con cada una de las 7 posiciones del hablante obtuvo los siguientes resultados.

	Posición hablante 1	Posición hablante 2	Posición hablante 3	Posición hablante 4	Posición hablante 5	Posición hablante 6	Posición hablante 7
Extractor	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)
1	67,83	67,61	64,74	64,4	65,79	66,04	74,24
2	71,4	65,12	65,64	63,38	68,06	70,59	75,23
3	70,02	68,87	66,54	67,13	69,94	71,97	73,54
4	71,3	69,32	73,61	69,35	72,46	71,64	74,77
5	70,05	66,64	68,67	70,31	71,47	71,2	72,1
6	71,67	71,02	69,54	71,31	70,5	72,77	70,93
7	75	70,58	68,8	72,61	73,07	73,81	75,23
8	71,22	71,95	72,69	73,08	73,85	73,69	72,54
9	74,84	72,88	73,17	74,63	76,2	74,7	76,37
Booster	75,94	74,61	74,99	74,25	78	78,91	77,29
Booster max	77,02	76,19	76,31	77,17	77,58	75,94	75,88

Figura C.33: Nivel de SPL en la posición A del *array*.

	Posición hablante 1	Posición hablante 2	Posición hablante 3	Posición hablante 4	Posición hablante 5	Posición hablante 6	Posición hablante 7
Extractor	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)	SPL (dB)
1	68,36	68,1	66,68	68,48	68,29	67,69	69,3
2	64,56	69,82	73,93	68,9	69,49	69,42	66,86
3	69,57	67,89	70,85	68,34	68,43	68,18	67,92
4	68,03	70,95	71,26	69,51	71,58	71,12	69,02
5	69,76	73,72	72,76	71,02	71,45	71,9	73,56
6	69,62	72,34	71,44	74,16	71,78	72,86	72,58
7	72,28	75,38	73,03	74,35	73,43	73,27	72,54
8	73,73	76,97	72,72	70,9	70,35	72,71	71,28
9	72,25	74,35	74,61	73,97	73,1	70,72	71,62
Booster	74,01	77,49	76,52	75,98	76,92	75,24	73,96
Booster max	76,1	75,1	75,93	78,16	74,1	72,88	74,01

Figura C.34: Nivel de SPL en la posición B del *array*.

En cuanto al nivel de SPL, la posición del *array* marca la diferencia ya que en la posición A el nivel de SPL es superior porque se capta más ruido del extractor que en la posición B.

Y, por último, a partir de las grabaciones realizadas para conseguir los valores de SPL se obtuvo el nivel de detección de la palabra de despertar de Alexa. Los resultados de la detección son los siguientes:

	Posición hablante 1	Posición hablante 2	Posición hablante 3	Posición hablante 4	Posición hablante 5	Posición hablante 6	Posición hablante 7
Extractor	Detección	Detección	Detección	Detección	Detección	Detección	Detección
1	SI	SI	SI	SI	SI	SI	SI
2	SI	SI	SI	SI	SI	SI	SI
3	SI	SI	SI	SI	SI	SI	SI
4	SI	SI	SI	SI	SI	SI	SI
5	SI	SI	SI	SI	SI	SI	SI
6	SI	SI	SI	SI	SI	SI	SI
7	SI	SI	SI	SI	SI	SI	SI
8	NO	SI	NO	SI	SI	SI	SI
9	NO	NO	NO	NO	NO	SI	NO
Booster	NO	NO	NO	NO	NO	NO	NO
Booster max	NO	NO	NO	NO	NO	NO	NO

Figura C.35: Detección palabra Alexa en la posición A del *array*.

	Posición hablante 1	Posición hablante 2	Posición hablante 3	Posición hablante 4	Posición hablante 5	Posición hablante 6	Posición hablante 7
Extractor	Detección	Detección	Detección	Detección	Detección	Detección	Detección
1	SI	SI	SI	SI	SI	SI	SI
2	SI	SI	SI	SI	SI	SI	SI
3	SI	SI	SI	SI	SI	SI	SI
4	SI	SI	SI	SI	SI	SI	SI
5	SI	SI	SI	SI	SI	SI	SI
6	SI	SI	SI	SI	SI	SI	SI
7	SI	SI	SI	SI	SI	SI	SI
8	SI	SI	SI	SI	SI	SI	SI
9	SI	SI	SI	SI	SI	SI	SI
Booster	SI	SI	SI	SI	SI	NO	NO
Booster max	SI	SI	NO	SI	NO	NO	NO

Figura C.36: Detección palabra Alexa en la posición B del *array*.

Se puede concluir que con el *array* circular se consiguen buenos resultados en la mayoría de las potencias del extractor de la placa de inducción. Se puede observar que la capacidad de detección de la palabra "ALEXA" no depende tanto de la distancia al sensor, como de la propia posición del sensor. En la posición A, la palabra puede ser detectada hasta potencias en torno al 7-8. Sin embargo, en la posición B se puede detectar en casi todo el rango de potencias, salvo en las posiciones más alejadas en las que se puede detectar la palabra clave con potencias del extractor al 9.

Anexo D. Segundo *setup* de pruebas

Se hicieron una serie de pruebas para evaluar el nivel de SNR y de detección con diferentes potencias del extractor, pero en este caso sin aplicar ningún *beamforming* a las señales, aplicando el *Delay and Sum* y finalmente aplicando el superdirectivo. Estas pruebas se llevaron a cabo con el *array* rectangular de 8, 12 y 16 micrófonos y con el *array* circular en 4 posiciones.

Se realizaron pruebas primero a una distancia de 4 metros y ángulo de 90º entre el altavoz y el *array* rectangular de 8, 12 y 16 micrófonos, la placa se utilizó con el cristal y los resultados se presentan en la **Tabla D.18**, la **Tabla D.19** y la **Tabla D.20**.

Distancia = 4m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-13,89	-9,4	-0,44	20%	100%	100%
3	-22,99	-18,02	-8,59	0%	0%	30%
5	-27,88	-23,2	-13,72	0%	0%	0%
7	-32,41	-27,05	-19,09	0%	0%	0%
9	-36,06	-30,72	-19,99	0%	0%	0%
Booster max	-38,3	-32,85	-22,44	0%	0%	0%

Tabla D.18: Niveles de SNR y detección con 8 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 4m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-13,89	-9,18	-1,1	0%	0%	100%
3	-22,99	-17,68	-8,56	0%	0%	0%
5	-27,88	-22,85	-13,3	0%	0%	0%
7	-32,41	-26,64	-18,75	0%	0%	0%
9	-36,06	-30,28	-19,98	0%	0%	0%
Booster max	-38,3	-32,44	-22,89	0%	0%	0%

Tabla D.19: Niveles de SNR y detección con 12 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 4m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-13,89	-10,84	-2,44	0%	0%	100%
3	-22,99	-20,08	-8,47	0%	0%	0%
5	-27,88	-25,44	-13,01	0%	0%	0%
7	-32,41	-28,98	-18,43	0%	0%	0%
9	-36,06	-32,82	-19,69	0%	0%	0%
Booster max	-38,3	-34,93	-23,02	0%	0%	0%

Tabla D.20: Niveles de SNR y detección con 16 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Por otro lado, los resultados que se consiguen con el *array* circular en esta posición son:

Distancia = 4m				Angulo = 90		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	1,92	2,03	3,45	100%	100%	100%
3	-1,04	-1,41	1,75	100%	100%	100%
5	-1,83	-2,01	1,71	100%	100%	100%
7	-5,34	-5,58	3,25	100%	100%	100%
9	-9,08	-9,28	1,75	0%	0%	100%
Booster max	-11,28	-11,45	2,58	0%	0%	100%

Tabla D.21: Niveles de SNR y detección con el *array* circular de 6 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Mientras que cuando el ángulo es de 30º y la distancia de 4 metros se obtienen los resultados mostrados en la **Tabla D.22**, la **Tabla D.23** y la **Tabla D.24** para el *array* rectangular con 8, 12 y 16 micrófonos.

Distancia = 4m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-14,5	-10,95	-5,76	0%	0%	100%
3	-23,76	-19,63	-14,21	0%	0%	0%
5	-28,3	-23,77	-17,41	0%	0%	0%
7	-33,01	-28,79	-22,89	0%	0%	0%
9	-37,24	-32,46	-25,77	0%	0%	0%
Booster max	-39,04	-34,33	-27,63	0%	0%	0%

Tabla D.22: Niveles de SNR y detección con 8 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 4m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-14,5	-10,64	-6,98	0%	0%	20%
3	-23,76	-19,32	-14,12	0%	0%	0%
5	-28,3	-23,46	-17,87	0%	0%	0%
7	-33,01	-28,61	-23,5	0%	0%	0%
9	-37,24	-32,15	-26,28	0%	0%	0%
Booster max	-39,04	-33,97	-28,46	0%	0%	0%

Tabla D.23: Niveles de SNR y detección con 12 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Distancia = 4m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	-14,5	-12,67	-6,28	0%	0%	30%
3	-23,76	-21,26	-14,02	0%	0%	0%
5	-28,3	-26,03	-17,23	0%	0%	0%
7	-33,01	-30,63	-23,05	0%	0%	0%
9	-37,24	-34,57	-26,53	0%	0%	0%
Booster max	-39,04	-36,35	-28,69	0%	0%	0%

Tabla D.24: Niveles de SNR y detección con 16 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

Y finalmente los resultados para el *array* circular en la posición de 30º son:

Distancia = 4m				Angulo = 30		
	SNR (dB)			Detección		
Extractor	Sin procesado	DS	SP	Sin procesado	DS	SP
1	2,09	2,15	2,51	100%	100%	100%
3	0,58	0,08	2,55	100%	100%	100%
5	-1,14	-2	2,48	100%	100%	100%
7	-4,77	-5,89	2,57	100%	100%	100%
9	-8,11	-9,36	1,67	0%	0%	100%
Booster max	-10,4	-11,66	-3,25	0%	0%	100%

Tabla D.25: Niveles de SNR y detección con el *array* circular de 6 micrófonos. DS: *Delay and Sum*. SP: superdirectivo.

En las dos posiciones que se acaban de mostrar, la principal conclusión obtenida es que el *array* circular es el que mejores valores de detección y SNR es capaz de obtener y por tanto mejores prestaciones ofrece para el control de la voz del usuario.