



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA**

FILTRADO ESPACIAL DE SEÑALES ACÚSTICAS

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA.
INGENIERÍA ELÉCTRICA – PROCESAMIENTO DIGITAL DE
SEÑALES

P R E S E N T A:

Ing. Bogdad Roberto Carlos Espinosa Vargas.

TUTOR:

M.I. Larry Hipólito Escobar Salguero

2012

JURADO ASIGNADO:

Presidente: Dr. Bohumil Psenicka.
Secretario: Dr. Pablo Roberto Pérez Alcázar.
Vocal: M. I. Larry Escobar Salguero.
1er Suplente: Dra. Lucia Medina Gómez.
2do Suplente: Dra. Fátima Moumtadi.

Lugar donde se realizo la tesis:

Laboratorio de procesamiento digital de señales, Posgrado de Ingeniería UNAM.
México DF.

TUTOR DE TESIS:

M. I. Larry Escobar Salguero.

FIRMA

Dedicatorias.

A mis padres, hermanos y sobrino por el apoyo que me han dado y que se que siempre tendré. Reciban este trabajo como muestra de gratitud y cariño, como símbolo de un triunfo y esfuerzo tanto personal como profesional, que hemos logrado juntos con paciencia, empeño y dedicación.

Agradecimientos.

*A la **Universidad Nacional Autónoma de México** y a la **Facultad de Ingeniería**, por haberme brindado una educación inmejorable.*

*Al M. en I. **Larry Escobar Salguero** por su apoyo para la realización de este trabajo.*

A todos los miembros del jurado por brindar parte de su tiempo.

A los mejores amigos en el mundo y que tuve la fortuna de encontrar ustedes saben quienes son, estamos siempre en las buenas y en las malas, esta es una de las buenas, a todos muchas gracias por ser mis amigos y apoyarme en todo momento, este logro es de todos!!!

Resumen

El presente trabajo muestra el diseño e implementación de un sistema de filtrado espacial de señales acústicas en un DSP para su funcionamiento en tiempo real.

Las señales acústicas son consideradas ondas planas para su procesamiento y el diseño del sistema. La técnica con la cual se implementó este sistema es el Beamforming o formador de haz, del cual existen diversas variaciones y métodos, ya sean fijos o adaptables; en este caso se simuló el formador de haz fijo y el cancelador de lóbulos laterales generalizado (GSC), observando un comportamiento mucho mejor del último, razón por la cual se eligió para ser el método utilizado en el sistema. Se diseñó un arreglo de micrófonos con características calculadas a partir de las condiciones del laboratorio de pruebas, en especial, de las señales acústicas que se quieren filtrar, en este caso el sistema se diseñó para señales acústicas de banda ancha, en especial, la voz.

El sistema integra un software que fue programado en el DSP elegido, el cual recibe las señales captadas por el arreglo de micrófonos, previamente amplificadas y acondicionadas por un circuito diseñado para ese propósito; hace una división de las señales por bandas de frecuencias; separa las señales de los micrófonos en subarreglos, implementando un formador de haz GSC por cada banda; y al final sumando los resultados parciales para tener el resultado total.

El sistema se probó con señales de tonos acústicos y con voz, mostrando un comportamiento esperado dentro de los parámetros teóricos con los que fue diseñado, se obtuvieron diferentes patrones direccionales parciales y totales, que indican que el sistema realiza el filtrado espacial de las señales acústicas que provienen de direcciones angulares diferentes a la de interés.

Índice

Capítulo 1. Introducción.	1
1.1 Objetivo.	3
1.2 Definición del problema.	3
Capítulo 2. Formador de haz y arreglo de sensores.	5
2.1 Ondas acústicas.	6
2.1.1 Ondas esféricas.	6
2.1.2 Campo lejano.	7
2.1.2.1 Principio de Huyghens.	9
2.1.3 Campo cercano.	9
2.1.4 Propagación de ondas acústicas.	11
2.2 Arreglo de Sensores.	12
2.2.1 Señales en sensores activos y pasivos.	12
2.2.2 Consideraciones del espaciamiento entre los elementos del arreglo.	12
2.2.2.1 Arreglo de sensores idénticos.	13
2.2.3 Arreglos lineales.	17
2.2.4 Arreglos planares.	24
2.2.5 Desempeño del arreglo.	26
2.3 Resumen.	26
Capítulo 3. Técnicas y métodos de formador de haz.	27
3.1 Fundamentos y tipos de formador de haz.	28
3.2 Formadores de haz fijos.	30
3.2.1 Formador de haz de retardo y suma.	30
3.2.2 Formador de haz de filtrado y suma.	31
3.2.3 Formador de haz de subarreglo.	32
3.3 Formadores de haz adaptables.	33
3.3.1 Cancelador de lóbulos laterales generalizado (GSC).	33
3.3.2 AMNOR.	36
3.3.3 Post-filtrado.	37
3.4 Formador de haz con descomposición wavelet.	40
3.4.1 Representación multiresolución.	41
3.4.2 Aproximación wavelet.	42
3.5 Resumen.	43
Capítulo 4. Diseño e implementación del sistema.	44
4.1 Diseño general del sistema.	45
4.2 Diseño del arreglo de micrófonos.	45
4.3 Determinación del método a usar.	50
4.4 Implementación del sistema en un DSP.	64
4.4.1 Hardware general del sistema.	64
4.4.2 Software del sistema.	66
4.5 Resumen.	75

Capítulo 5. Pruebas y Resultados.	76
5.1 Pruebas al algoritmo.	77
5.1.1 Operaciones.	77
5.1.2 Pruebas de etapas de entrada y salida.	78
5.2 Pruebas con señales fijas.	78
5.2.1 Patrón direccional teórico.	79
5.3 Pruebas con señales reales.	85
5.3.1 Patrón direccional de la banda 1.	86
5.3.2 Patrón direccional de la banda 2.	87
5.3.3 Patrón direccional de la banda 3.	88
5.3.4 Patrón direccional de la banda 4.	89
5.3.5 Patrón direccional total del sistema.	90
5.4 Pruebas con voz.	91
5.5 Desempeño del sistema.	93
5.6 Recursos utilizados.	94
5.7 Resumen.	95
Capítulo 6. Conclusiones.	96
Bibliografía.	99
Anexos.	102
Glosario de términos.	103
Tarjeta DSP.	105
Código principal.	107

Índice de Figuras

Figura 2.1 Campo lejano de dos fuentes simples.	7
Figura 2.2 Funciones de directividad para fuentes Simples.	8
Figura 2.3 Campo de interferencia de una reja de difracción.	9
Figura 2.4 Construcción grafica del campo cercano.	10
Figura 2.5 Par de sensores omnidireccionales idénticos.	13
Figura 2.6 (a) Patrón de haz del arreglo de dos sensores, $d/\lambda_0 = 0.5$.	15
Figura 2.6 (b) Patrón de haz del arreglo de dos sensores, $d/\lambda_0 = 1$.	16
Figura 2.6 (c) Patrón de haz del arreglo de dos sensores, $d/\lambda_0 = 1.5$.	16
Figura 2.7 Arreglo lineal.	17
Figura 2.8 Patrón de haz para un arreglo lineal de tres sensores, $(d/\lambda_0) = 0.5$.	18
Figura 2.9 Patrón de haz para un arreglo lineal de cuatro sensores, $(d/\lambda_0) = 0.5$.	19
Figura 2.10 (a) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 0.1$.	19
Figura 2.10 (b) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 0.2$.	20
Figura 2.10 (c) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 0.5$.	20
Figura 2.10 (d) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 1$.	21
Figura 2.11 Arreglo de sensores $N = 7$ y $d/\lambda_0 = 0.1 \rightarrow 1$.	22
Figura 2.12 Arreglo de sensores $d/\lambda_0 = 0.5$, $N = 1 \rightarrow 10$.	22
Figura 2.13 Arreglo lineal desplazado con elementos corridos en fase.	23
Figura 2.14 Patrón direccional de un arreglo lineal de siete elementos con $\delta = 15^\circ$.	24
Figura 2.15 Arreglo planar de forma rectangular.	24
Figura 3.1. Formador de haz digital básico.	28
Figura 3.2. Estructura de un formador de haz de filtrado y suma.	31
Figura 3.3. Estructura de subarreglo combinado.	32
Figura 3.4. Estructura del cancelador de lóbulos laterales generalizado.	34
Figura 3.5. Estructura del sistema AMNOR.	36
Figura 3.6. Formador de haz de filtrado y suma con post filtro.	38
Figura 3.7 Plano multiresolución tiempo-frecuencia.	40
Figura 3.8 Wavelet Morlet.	41
Figura 3.9 Banco de filtros para descomposición Wavelet.	42
Figura 3.10 Diagrama de bloques de un formador de haz modificado con descomposición Wavelet.	43
Figura 4.1 Esquema básico del sistema.	45
Figura 4.2 Número de micrófonos para las cuatro bandas a diferentes frecuencias.	48
Figura 4.3 Arreglo seleccionado con 15 micrófonos.	49
Figura 4.4 Distribución de micrófonos en el arreglo total y por banda.	50
Figura 4.5 Ventanas.	51
Figura 4.6 Matriz de dirección.	54
Figura 4.7 Diagrama a bloques del sistema formador de haz fijo multibanda.	55
Figura 4.8 Banda 1, ángulo 0° y formador de haz fijo.	56

Figura 4.9 Banda 2, ángulo 0° y formador de haz fijo.	56
Figura 4.10 Banda 3, ángulo 0° y formador de haz fijo.	57
Figura 4.11 Banda 4, ángulo 0° y formador de haz fijo.	57
Figura 4.12 Bandas 1 a 4, ángulo 0° y formador de haz fijo.	58
Figura 4.13 Banda total, ángulo 0° y formador de haz fijo.	58
Figura 4.14 Formador de haz fijo multibanda.	59
Figura 4.15. Diagrama a bloques del sistema con algoritmo GSC.	60
Figura 4.16 Banda 1, ángulo 0° y GSC.	60
Figura 4.17 Banda 2, ángulo 0° y GSC.	61
Figura 4.18 Banda 3, ángulo 0° y GSC.	61
Figura 4.19 Banda 4, ángulo 0° y GSC.	62
Figura 4.20 Bandas 1 a 4, ángulo 0° y GSC.	62
Figura 4.21 Banda total, ángulo 0° y GSC.	63
Figura 4.22 Banda total contorno con GSC.	63
Figura 4.23 Hardware del sistema.	64
Figura 4.24 Diagrama de flujo del sistema.	66
Figura 4.25 Recepción de datos en los buffers.	69
Figura 4.26 Números de los micrófonos y buffers.	70
Figura 4.27 Formador de haz fijo.	71
Figura 4.28 Matriz de bloqueo y vectores a .	72
Figura 5.1 Oscilador digital.	79
Figura 5.2 Patrón direccional de la banda 1 del sistema usando BF GSC en el DSP.	80
Figura 5.3 Patrón direccional de la banda 2 del sistema usando BF GSC en el DSP.	81
Figura 5.4 Patrón direccional de la banda 3 del sistema usando BF GSC en el DSP.	82
Figura 5.5 Patrón direccional de la banda 4 del sistema usando BF GSC en el DSP.	83
Figura 5.6 Patrón direccional total del sistema usando BF GSC en el DSP.	84
Figura 5.7 Diagrama del sistema para pruebas.	85
Figura 5.8 Patrón direccional de la banda 1 del sistema en el DSP con señales reales y GSC.	86
Figura 5.9 Patrón direccional de la banda 2 del sistema en el DSP con señales reales y GSC.	87
Figura 5.10 Patrón direccional de la banda 3 del sistema en el DSP con señales reales y GSC.	88
Figura 5.11 Patrón direccional de la banda 4 del sistema en el DSP con señales reales y GSC.	89
Figura 5.12 Patrón direccional total del sistema en el DSP con señales reales y GSC.	90
Figura 5.13 Patrón direccional total del sistema en el DSP con voz y GSC.	91
Figura 5.14 Patrón direccional total del sistema en el DSP con voz y GSC.	92
Figura 5.15 Patrón direccional total del sistema en el DSP con voz, GSC y ángulo -40°.	93
Figura A.1 Diagrama de bloques funcional del TMS320F28335.	105
Figura A.2 F28335 Vista superior.	106

Capítulo 1

Introducción



En muchas ocasiones se tienen diversas fuentes de sonido que son captadas por un sistema, el cual las procesará para obtener un cierto resultado; sin embargo, no necesariamente todas las fuentes de sonido son de interés, por tanto la entrada al sistema es una mezcla de señales acústicas que no pueden ser separadas de manera sencilla una vez captadas. En muchas aplicaciones se requiere de un sistema capaz de dirigir la recepción hacia una dirección en específico, captando así solo la fuente de sonido de interés y eliminando las demás señales, que pueden ser consideradas como interferencias.

Existen varias técnicas para filtrar o separar señales, siendo el Beamforming o formador de haz (BF) una técnica que ha sido utilizada por muchas décadas en telecomunicaciones, radar, en aplicaciones de arreglos de antenas y recientemente, adoptada para aplicaciones de banda ancha como mejora de voz.

Un formador de haz está compuesto por un conjunto de sensores, llamado arreglo, que puede orientar la energía de las señales en cierto ángulo. Los arreglos pueden ser lineales, planos o tridimensionales. La orientación electrónica utiliza técnicas de procesamiento de señales que explotan la información espacial en las señales del sensor, requiriendo operaciones de filtrado espacio-temporales [1].

En general las fuentes de señales emiten ondas esféricas que eventualmente llegan al arreglo de sensores, el cual consta de micrófonos lineales que extraen información espacial de las ondas esféricas. Dicha información espacial obtenida de las ondas esféricas es usada por el formador de haz para selección y separación espacial.

El formador de haz es la técnica más común para selección espacial de señales acústicas, siendo también muy útil para supresión de distorsiones y de reverberación. Los formadores de haz clásicos tratan con señales de banda angosta y se basan en el método de retardo y suma. En este método las señales de los sensores son retardadas y después sumadas. Al variar los retardos, el diseñador puede orientar el haz en una dirección de interés.

Un formador de haz basado en retardos y sumas tiene un patrón de selección angular específico y una supresión de lóbulos laterales restringida. Una extensión de esto es el formador de haz de suma ampliada, en el cual las señales de los sensores son retrasadas, amplificadas y sumadas, permitiendo al diseñador formar la selectividad angular con el patrón deseado.

Las técnicas clásicas de formador de haz mencionadas anteriormente fueron desarrolladas para aplicaciones de telecomunicaciones de banda angosta. Las características de los formadores de haz de retardo y suma, y de suma ampliada son extremadamente dependientes de la frecuencia, por lo que para superar esta dependencia, son requeridas otras extensiones para formadores de haz de banda ancha acústicos. [1]

1.1 Objetivo

Diseñar y realizar un sistema para filtrar espacialmente una señal acústica de interés mediante un arreglo de sensores e implementación de varios algoritmos en tiempo real en una plataforma DSP¹.

1.2 Definición del Problema

En diversas aplicaciones relacionadas con señales acústicas, las señales no deseadas o consideradas como interferencias son un problema que está siempre presente y que genera complicaciones en el desempeño de los procesos que utilizan dichas señales acústicas, produciendo resultados no deseados e incorrectos en las aplicaciones.

Una manera de evitar estos malos resultados es filtrar las señales acústicas para eliminar las interferencias que se presentan. El filtrado analógico, es decir, filtrar la señal antes de ser adquirida por el sistema implementado, implica tener componentes y circuitos extra en el sistema. El filtrado digital puede ser buena opción, sin embargo, tanto en el filtrado digital como en el analógico, las interferencias que tengan la misma frecuencia que las señales acústicas no pueden ser eliminadas por completo.

Una opción más es hacer un filtrado espacial de las señales, es decir, captar solo las señales acústicas de interés sin captar las posibles interferencias, recibiendo sólo las señales que provengan de una dirección establecida y eliminando todas las señales que provengan de cualquier otra dirección, con lo cual se evita la contaminación de la señal de interés.

Antecedentes

Actualmente existen varios métodos para la formación de haz y diferentes algoritmos para hacer la estimación de la dirección de arribo de las señales.

En cuanto a la detección y estimación de dirección de arribo de señales sobre un arreglo de sensores, se están desarrollando diferentes métodos utilizando criterios de máxima verosimilitud [2] y la cota de Cramér-Rao [3], estos últimos desarrollados por un grupo de ingenieros en la UCLA. Diferentes métodos de formación de haz han sido propuestos para mejorar el desempeño de un formador de haz básico como el de retraso y suma, algunos métodos nuevos utilizan otro tipo de técnicas como la mínima varianza [4], desarrollado en la universidad de Oslo Noruega, o el basado en la descomposición Wavelet [5], del Ocean Acoustic Group del MIT, y los formadores de haz de corrimiento de fase y jerárquicos descritos en [6].

Alcances

El tener un formador de haz de buena calidad y desempeño permite realizar diversos procesos y obtener información acerca de las señales acústicas en estudio. El primer parámetro que se puede obtener de una señal de sonido específica es la dirección de arribo (DOA) y, después, a partir de este parámetro obtener más información.

¹ DSP: Procesador digital de señales.

A partir de la dirección de arribo se puede calcular la distancia que hay entre la fuente de sonido y el arreglo de sensores, o determinar si la fuente se está moviendo. Si este es el caso se puede seguir al objeto, o calcular a qué velocidad se está moviendo.

Todo lo anterior se puede hacer para el caso de una sola fuente de sonido, aunque si el sonido que arriba a los sensores es una mezcla de señales, el formador de haz es una herramienta que se puede utilizar para seleccionar solo una de esas señales.

En este trabajo se pretende hacer un filtrado espacial de señales acústicas incidentes en un arreglo de micrófonos, para detectar la señal de interés que provenga de algún objeto con el fin de detectar al mismo.

Para lograr los objetivos se requiere:

- Detectar la dirección de arribo de una fuente de sonido de interés.
- Implementar un formador de haz de muy buena calidad, utilizando diversos algoritmos, para lograr el punto anterior.
- Distinción de la fuente de interés, descartando claramente las demás fuentes de sonido que puedan estar presentes en el mismo espacio de interés, mediante el uso del formador de haz.
- Acondicionar la señal de interés para su procesamiento.
- Desarrollar un sistema funcional implementado en un DSP.

Descripción de Capítulos

- El *Capítulo 2* describe los temas de formador de haz y arreglo de sensores de manera teórica.
- En el *Capítulo 3* se describen las técnicas y métodos de formador de haz y sus características.
- El *Capítulo 4* presenta el diseño e implementación del sistema, simulaciones y definición de parámetros a utilizar.
- El *Capítulo 5* muestra las pruebas realizadas al sistema a fin de evaluar su desempeño y se presentan los resultados obtenidos en la implementación del sistema.
- El *Capítulo 6* muestra las conclusiones obtenidas a partir de la implementación y funcionamiento del sistema.

Capítulo 2

Formador de Haz y Arreglo de Sensores



En este capítulo se presenta la teoría básica de las señales acústicas, su comportamiento en campo lejano y campo cercano y su propagación, debido a que estas señales son las de interés en el desarrollo de este trabajo.

También se presenta la diferencia entre los sensores activos y pasivos, el comportamiento de un arreglo de dos sensores, el análisis de los arreglos de sensores lineales y planares, el patrón direccional para diferentes valores en los parámetros que lo definen y, por último, el estudio del desempeño de dichos arreglos.

2.1 Ondas Acústicas

Dependiendo del tipo y origen de las diversas señales, éstas tienen características particulares que deben ser tomadas en cuenta para su manejo, proceso y análisis.

El sonido es el resultado de variaciones de presión u oscilaciones en un medio elástico, como aire o agua, generadas por una superficie vibrante o un flujo turbulento de fluido. El sonido se propaga en forma de ondas longitudinales, involucrando una sucesión de compresiones y rarefacciones en el medio elástico. Cuando una onda de sonido se propaga en el aire las oscilaciones en presión son por encima y por debajo de la presión atmosférica del ambiente [7].

Las ondas acústicas que consisten de un solo tono puro son caracterizadas por:

- Amplitud, que se describe como el valor máximo de presión.
- Longitud de onda (λ), que es la distancia que viaja la onda de presión durante un ciclo.
- Frecuencia (f), que es el número de ciclos de variaciones de presión en el medio por unidad de tiempo, o simplemente el número de ciclos por segundo y está expresado en Hertz (Hz).
- Periodo (T), que es el tiempo que le toma a un ciclo de la onda en pasar un punto fijo, está relacionado a la frecuencia por $T = \frac{1}{f}$.

2.1.1 Ondas Esféricas

La onda más simple no plana es una onda esférica. Consideremos una sola fuente de ondas acústicas: una esfera pulsante radialmente de diámetro mucho más pequeño que la longitud de onda de las ondas radiadas, tal fuente radía sonido uniformemente en todas direcciones. Un tubo largo de diámetro pequeño conduciendo sonido de una bocina también radiará ondas aproximadamente esféricas desde su extremo abierto. Ahora localicemos la fuente en un espacio libre o en una cámara libre de eco, tal que las ondas no son reflejadas de nuevo al campo del sonido radiado; bajo estas condiciones la energía se distribuye uniformemente sobre una superficie esférica completa a cualquier radio r a partir de la fuente. Entonces la intensidad ϑ ($watts/m^2$) estará relacionada con la potencia de la fuente W ($watts$) por [8]

$$\vartheta = W / 4\pi r^2 \quad (2.1)$$

A una distancia radial r más grande que la longitud de onda o el diámetro de la fuente a ($r \gg a$) el frente de onda es esencialmente plano en distancias comparables con la longitud de onda. En este límite el sonido radía con la impedancia característica de las ondas planas $\rho_0 v [\Omega]$, y la intensidad es proporcional al cuadrado de la presión del sonido. El pico de presión se expresa como: $P = [\vartheta 2\rho_0 v]^{1/2}$ y con el uso de la ecuación (2.1) para la presión de sonido instantánea

$$p = \left(\rho_0 v W / 2\pi \right)^{1/2} \frac{e^{j(\omega t - kr)}}{r} \quad (2.2)$$

Con el uso de $W = P_a^2 S / 2\rho_0 v$ donde $S = 4\pi a^2$ es la superficie radiada de la fuente, la ecuación (2.2) se puede reescribir como

$$p = P_a a \frac{e^{j(\omega t - kr)}}{r} \quad (2.3)$$

La amplitud de la presión P_a está dada por la ecuación (2.2) sólo si W es la potencia realmente radiada o la efectiva [8].

2.1.2 Campo Lejano

El campo lejano en las ondas acústicas se refiere al caso cuando las ondas de sonido están lo suficientemente alejadas de la fuente que las produce tal que estas se pueden considerar como ondas planas.

Examinemos ahora el campo de radiación alrededor de dos fuentes colocadas separadas a una distancia d , vibrando exactamente en fase y con amplitudes iguales. La presión del sonido de cada fuente está dada por la ecuación (2.3) con el mismo valor de P_a para cada una. En la figura 2.1 se muestra un punto de campo B a una distancia r_1 de la fuente S_1 y r_2 de S_2 [8].

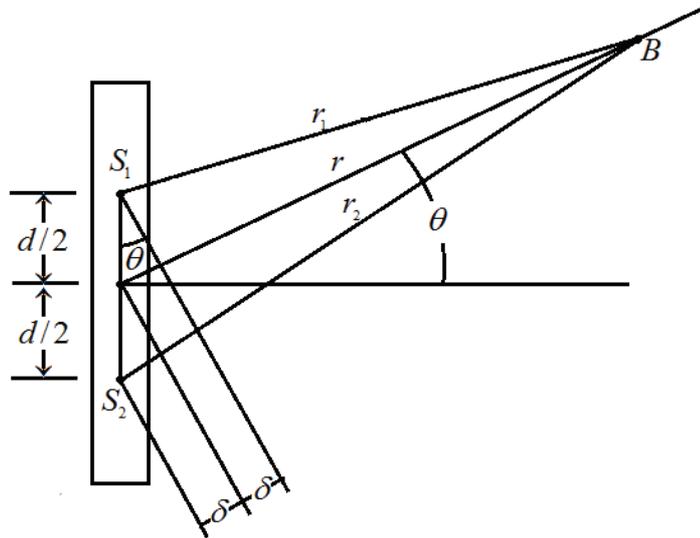


Figura 2.1 Campo lejano de dos fuentes simples [8].

La posición de B es definida por la distancia r desde el punto medio entre las fuentes y el ángulo θ entre r y la normal a la línea $S_1 - S_2$. Ahora consideremos solo el campo lejano,

restringimos r a valores mucho más grandes que d . Esto simplifica la geometría y nos permite hacer las aproximaciones $r_2 - r_1 = r - r_1 = \delta = (d/2) \text{sen}\theta$. La presión combinada en B se expresa por

$$p_B = \frac{P_a a e^{j(\omega t - kr)}}{r} \left[e^{jk\delta} + e^{-jk\delta} \right]$$

$$= \frac{2P_a a e^{j(\omega t - kr)}}{r} \cos \left[\frac{kd}{2} \text{sen}\theta \right] \quad (r \gg d)$$

donde $k = \omega/v = 2\pi/\lambda$

En esta aproximación se supone $r_1 = r_2 = r$ en el denominador, debido a que las pequeñas diferencias en distancia tienen un efecto despreciable en las amplitudes de presión.

La función de directividad dipolar $\cos[(\pi d/\lambda) \text{sen}\theta]$ se gráfica en la figura 2.2 para varios casos.

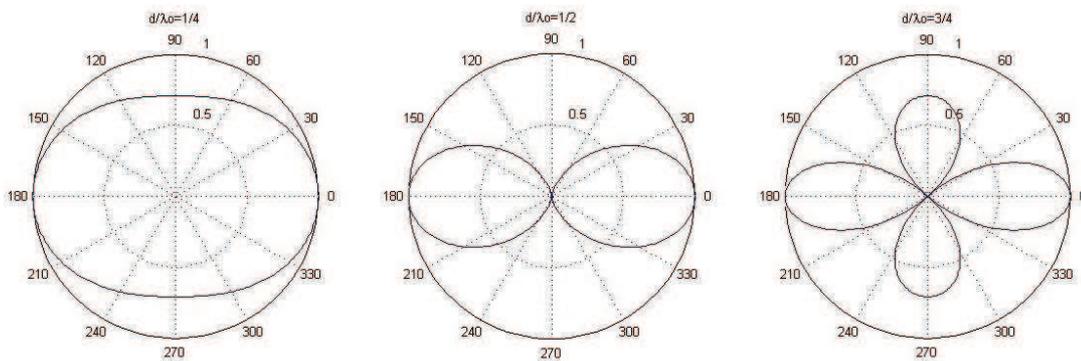


Figura 2.2 Funciones de directividad para fuentes Simples.

- Si $d/\lambda = 0$, el patrón de radiación es un círculo y el patrón de dipolo es indistinguible para una fuente simple. Las dos amplitudes de las fuentes se suman (estas fuentes están en fase) y dan un valor doble de presión en el campo (máximos en 0° y 180° , mínimos en 90° y 270°).
- En $d/\lambda = 1/4$, el patrón de radiación es desplazado a 90° del eje, y a $d/\lambda = 1/2$ la presión se hace cero en $\pm 90^\circ$. En estas gráficas el vector radial es proporcional a la presión de sonido en el campo lejano, y la coordenada angular corresponde a ángulos reales en el espacio. El patrón de radiación completo es una forma tridimensional, simétrica alrededor de la línea a través de la fuente.
- A $d/\lambda = 3/4$, existen lóbulos laterales en 90° y 270° , pero estos son menores en presión que el lóbulo principal en el eje dipolo, máximos en 0° y 180° . Las direcciones de cero presión en este caso son alrededor de 42° del eje.

El ancho del lóbulo principal, o simplemente el ancho del haz, se especifica generalmente por el ángulo θ entre el eje y el primer cero. El primer cero en la ecuación (2.4) ocurre cuando $(kd/2)\text{sen}\theta = \pi/2$, y tenemos que [8]

$$\theta = \text{sen}^{-1}(\lambda/2d) \text{ Ancho de medio haz para el dipolo.}$$

2.1.2.1 Principio de Huyghens

El principio de Huyghens establece que cualquier fenómeno ondulatorio puede ser analizado por la suma de algunas contribuciones de fuentes simples, propiamente seleccionadas en fase y amplitud para representar la situación física. Cada “onda de Huyghens” puede ser expresada con la forma de la ecuación (2.3).

Podemos tomar un radiador de cualquier tamaño y forma, dividimos su superficie en un número de áreas cada una lo suficientemente pequeña para satisfacer la suposición de fuente simple, y sumamos sus contribuciones transmitidas a cualquier punto en el espacio. En un sentido, el dipolo es la primera aproximación a una fuente lineal de longitud finita [8].

2.1.3 Campo Cercano

Ahora analicemos el campo cercano suponiendo que una onda plana puede formarse de un número infinito de ondas Huyghens, éstas se combinan en fase para formar un frente de onda plano normal a la dirección de propagación, pero se cancela en otras direcciones. La eliminación de una o más ondas de Huyghens de un frente de onda plano original, por un número de líneas paralelas opacas (obstáculos), resultará en una cancelación imperfecta en ciertos ángulos. La figura 2.3 muestra que un arreglo regular de dichas líneas producirá frentes planos de fase igual en ciertos ángulos específicos definidos por $\text{sen}\theta = nd/\lambda$ donde d es la distancia entre los centros de las líneas de bloqueo y $\pm n = 0, 1, 2, 3, \dots$. También se nota que el sistema de la figura 2.3 corresponde a una rejilla de difracción óptica [8].

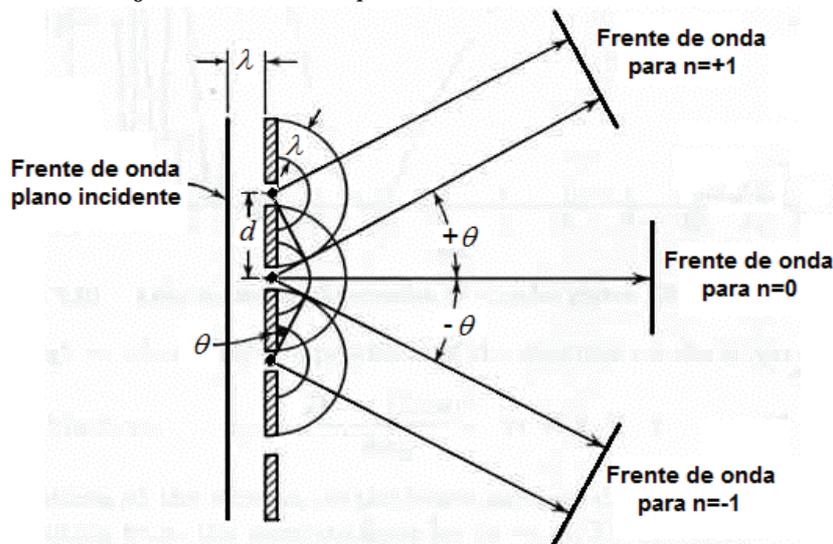


Figura 2.3 Campo de interferencia de una rejilla de difracción [8].

Las mismas consideraciones llevan a efectos de difracción similares siempre que una onda plana esté limitada en direcciones normales a la dirección de propagación, en la frontera las ondas de Huyghens se vuelven manifiestas debido a que las componentes angulares no están completamente canceladas por las ondas de Huyghens vecinas.

Consideremos ahora una onda plana limitada por dos bordes paralelos opacos, esto es equivalente a una fuente de forma de cilindro regular, cuya longitud es mucho mayor que su ancho ($l \gg D$). Un plano seccional (x, y) paralelo a D y normal a l se representa en la figura 2.4 [8].

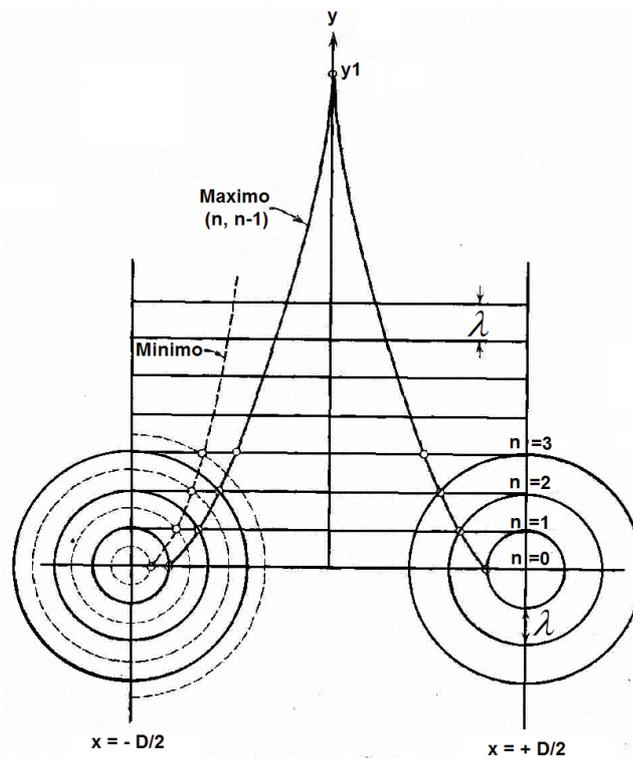


Figura 2.4 Construcción grafica del campo cercano [8].

Aproximamos la radiación del cilindro como un tren de ondas planas limitadas en $x = \pm D/2$ y dos conjuntos de ondas de Huyghens cilíndricas originadas en las fronteras $x = +D/2$ y $x = -D/2$. La posición instantánea de los máximos de presión de la onda plana y los dos conjuntos de ondas cilíndricas están indicadas con líneas y círculos separados a una distancia λ . En las intercepciones de los círculos y líneas continuas las ondas planas y las ondas de Huyghens están en fase y las presiones de sonido se suman. Existe otro conjunto de intercepciones donde la onda cilíndrica y la onda plana tienen fases opuestas y por lo tanto se cancelan. Las posiciones de los puntos de suma (máximos) y de cancelación (mínimos) se representan por parábolas que tienen foco en $x = \pm D/2$ y $y = 0$. En la figura 2.4 las parábolas se representan por ambas condiciones; línea sólida para un máximo y línea punteada para un mínimo.

Se observa que el primer conjunto de máximos resulta de la intersección del n -ésimo círculo con la $(n-1)$ -ésima línea, el segundo conjunto por la combinación $[n, (n-2)]$, etc. La parábola

$[n, (n-1)]$ se obtiene combinando las ecuaciones de los círculos de Huyghens $(x \pm D/2)^2 + y^2 = (n\lambda)^2$ y las líneas rectas $y = (n-1)\lambda$, que resulta en

$$(x \pm D/2)^2 = 2y\lambda + \lambda^2$$

Si hacemos que x tienda a cero obtenemos la posición del máximo de interferencia que está localizado justo en el eje del haz, que, en este caso, $[n, (n-1)]$, es el primer máximo encontrado si la fuente de cilindro se aproxima al infinito [8].

2.1.4 Propagación de ondas acústicas

Las ondas de sonido se propagan a través de los fluidos no viscosos como ondas longitudinales. Las moléculas en el fluido se mueven hacia adelante y atrás en la dirección de propagación, produciendo zonas de compresión y expansión. Una ecuación de onda generalizada para ondas acústicas es algo compleja ya que depende de las propiedades del fluido, sin embargo, suponiendo un fluido ideal con viscosidad cero, la ecuación de onda se puede expresar como [9]

$$\nabla^2 x(t, \mathbf{r}) - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} x(t, \mathbf{r}) = 0$$

Donde $x(t, \mathbf{r})$ es una función representando la presión del sonido en un punto en el tiempo y espacio,

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

∇^2 es el operador Laplaciano, v la velocidad de propagación, la cual, depende de la presión y de la densidad del fluido, y es aproximadamente de 340ms^{-1} en el aire a temperatura ambiente.

La solución de la ecuación diferencial de onda se puede obtener usando el método de separación de variables. La solución para una onda plana monocromática está dada por:

$$x(t, \mathbf{r}) = Ae^{j(\omega t - \mathbf{k} \cdot \mathbf{r})}$$

Donde A es la amplitud de onda, $\omega = 2\pi f$ es la frecuencia en radianes por segundo, y el vector número de onda \mathbf{k} , que indica la velocidad y dirección de propagación de la onda y está dado por:

$$\mathbf{k} = \frac{2\pi}{\lambda} [\text{sen}\theta \cos\phi \quad \text{sen}\theta \text{sen}\phi \quad \cos\theta]$$

La longitud de onda λ está relacionada con v por $\lambda = v/f$.

Alternativamente la solución para una onda esférica se puede obtener como

$$x(t, \mathbf{r}) = -\frac{A}{4\pi r} e^{j(\omega t - kr)}$$

donde $r = |\mathbf{r}|$ es la distancia radial a la fuente y k es el número de onda escalar, dado por $2\pi/\lambda$.

La solución de onda esférica muestra que la amplitud de una señal decae con una tasa proporcional a la distancia a la fuente. Aunque las ondas de sonido son típicamente esféricas por naturaleza, éstas pueden ser consideradas como ondas planas a una distancia suficiente de la fuente [9].

2.2 Arreglo de Sensores

Un arreglo de sensores es un conjunto de los mismos colocados a una distancia específica entre ellos. Las características de los elementos del arreglo y el acomodo de los elementos usado para formar el arreglo tienen un impacto directo en el desempeño del arreglo [10].

2.2.1. Señales en sensores activos y pasivos

Los dispositivos con sensores activos realizan la transmisión y recepción, tal como los sistemas de radar y sonar generando un pulso de una señal conocida (o un tren de pulsos) que se propaga a través de un medio de transmisión y es reflejado por un blanco u objetivo de regreso al emisor. Durante la mayoría del intervalo de tiempo de sensado la señal deseada está ausente, sin embargo, se tiene una idea de la estructura de la señal y la dirección de arribo así que la señal puede ser reconocida fácilmente cuando esté presente.

En el caso de los sistemas de sensores pasivos, la señal deseada es generada por el blanco u objetivo mismo (o evento de interés) y en muchos casos puede estar presente en casi todo el intervalo de sensado. El problema básico de los sistemas pasivos es distinguir la señal objetivo deseada del ruido de fondo. Sin embargo, en contraste con los dispositivos de sensores activos, la dirección de arribo de la señal deseada puede no ser conocida de antemano, y la señal deseada puede ser desconocida. Los métodos más comunes para distinguir la señal deseada de una señal de interferencia es la existencia de una banda de frecuencias dentro de la cual la señal deseada puede estar, en algunos casos el nivel de potencia de la señal deseada puede ser conocida y usada como una característica distintiva [10].

2.2.2 Consideraciones del espaciamiento entre los elementos del arreglo

La posición de los elementos del arreglo determina la resolución del arreglo y los efectos de interferencia llamados glóbulos gratinados. En general, la resolución incrementa conforme la dimensión del arreglo (o la separación entre elementos) incrementa. Una resolución alta del arreglo mejora la relación señal a ruido (SNR) máxima de salida cuando la separación angular entre la señal deseada y las señales no deseadas es pequeña.

La resolución alta también implica mínimos muy pronunciados en el patrón del arreglo, reduciendo la capacidad del arreglo de colocar mínimos amplios en fuentes de interferencia cercanas [10].

2.2.2.1 Arreglo de dos sensores idénticos

Consideremos un par de sensores idénticos separados una distancia d , supóngase, también, una señal $x(t)$ que incide en los dos sensores, en un plano que contiene a los dos elementos y la fuente de la señal, desde una dirección θ con respecto a la normal del arreglo, como se muestra en la figura 2.5.

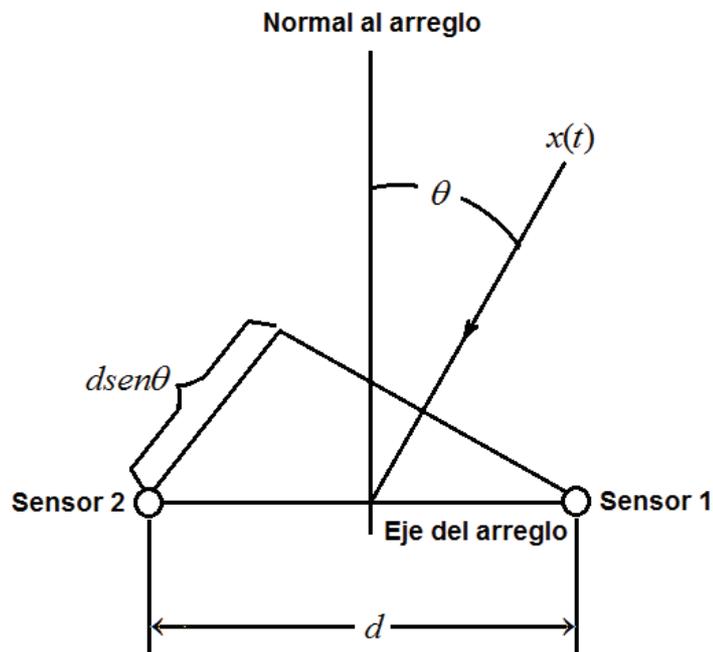


Figura 2.5 Par de sensores omnidireccionales idénticos [10].

En la figura 2.5 se puede observar que el sensor 2 experimenta un retraso de tiempo con respecto al sensor 1 dado por:

$$\tau = \frac{dsen\theta}{v}$$

donde:

v = velocidad de propagación de la onda

Suponiendo que la señal de salida del arreglo $y(t)$ está dada como la suma de las señales de los dos sensores

$$y(t) = x(t) + x(t - \tau)$$

Si $x(t)$ es una señal limitada en banda con una frecuencia central f_0 , entonces el retraso de tiempo τ corresponde a un corrimiento de fase de $2\pi\left(\frac{d}{\lambda_0}\right)\text{sen}\theta$ radianes, donde λ_0 es la longitud de onda correspondiente a la frecuencia central

$$\lambda_0 = \frac{v}{f_0}$$

La respuesta total del arreglo puede ser deducida simplemente considerando la suma fasorial de las contribuciones de las señales de los dos sensores, tal que

$$y(t) = \sum_{i=1}^2 x(t) e^{j(i-1)\psi}$$

donde

$$\psi = 2\pi\left(\frac{d}{\lambda_0}\right)\text{sen}\theta$$

El patrón direccional del arreglo, definido como la sensibilidad relativa de respuesta a señales de frecuencia específica provenientes de varias direcciones, se encuentra considerando sólo el término

$$A(\theta) = \sum_{i=1}^2 e^{j(i-1)\psi}$$

El patrón direccional normalizado en decibeles para los dos elementos está dado por

$$G(\theta) = 10\log_{10} \left\{ \frac{|A(\theta)|^2}{4} \right\} (\text{decibeles}) \quad (2.4)$$

En la figura 2.6 se muestra $G(\theta)$ de la ecuación (2.4) graficada para $\frac{d}{\lambda_0} = 0.5, 1.0$ y 1.5 [10].

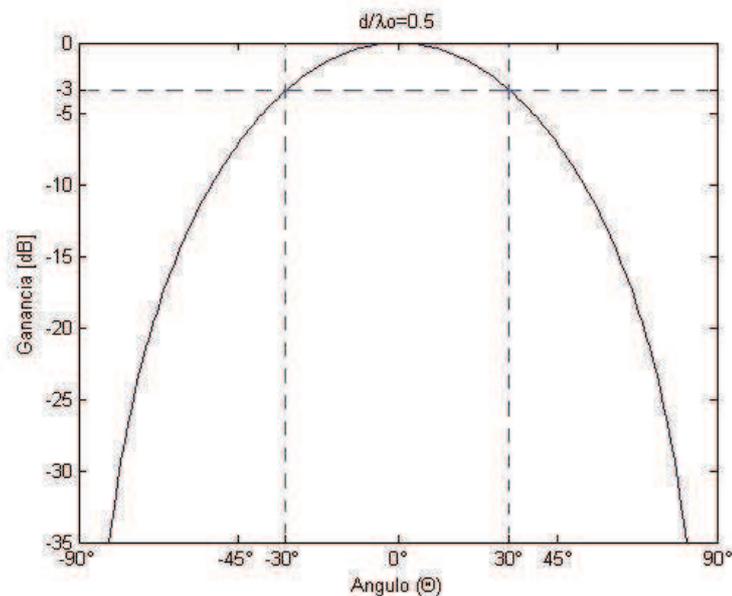


Figura 2.6 (a) Patrón de haz del arreglo de dos sensores, $d/\lambda_0 = 0.5$.

De la figura 2.6 (a) se observa que para $d/\lambda_0 = 0.5$ existe un lóbulo principal (o haz principal) teniendo un ancho de banda de 60° con ganancia de $3dB$ y mínimos en $\theta = \pm 90^\circ$. Los mínimos en $\theta = \pm 90^\circ$ ocurren debido a que, en la dirección de arriba, el frente de onda de la señal debe viajar exactamente $\lambda_0/2$ entre dos sensores, que corresponde a un desplazamiento de fase de 180° entre las señales que arriban a los dos sensores y entonces resulta en la cancelación exacta de la suma fasorial resultante. Si el espaciado entre elementos es menor a $0.5\lambda_0$, entonces la cancelación exacta en $\theta = \pm 90^\circ$ no ocurre, y en el límite cuando el espaciado se acerca a cero (ignorando los efectos mutuos de acoplamiento) el patrón direccional se vuelve el patrón omnidireccional de un sólo elemento. Existe una diferencia muy pequeña entre el patrón direccional de un sólo elemento y el de dos elementos colocados muy cercanos (no más lejos que $\lambda/4$), consecuentemente los arreglos que emplean muchos elementos colocados muy cerca son considerados “ineficientes” si es deseado usar el menor número de arreglos posibles para un nivel específico de lóbulos laterales y ancho de haz.

Si el espaciado entre elementos aumenta a más de $0.5\lambda_0$, los dos mínimos del patrón se mueven de $\theta = \pm 90^\circ$, ocurriendo en $\theta = \pm 30^\circ$ cuando $d = \lambda_0$, como se muestra en la figura 2.6 (b), estos mínimos en $\theta = \pm 30^\circ$ ocurren debido a que en el ángulo de arriba la diferencia de fase entre los dos sensores es nuevamente 180° , y resulta una cancelación exacta en la suma fasorial. Existen dos lóbulos laterales en $\theta = \pm 90^\circ$ que tienen una amplitud igual a la del lóbulo principal en $\theta = 0^\circ$ también aparecen debido a que la diferencia de fase entre los dos sensores es 360° , los dos fasores se alinean exactamente y la respuesta del arreglo es la misma que para un ángulo de arriba muy amplio [10].

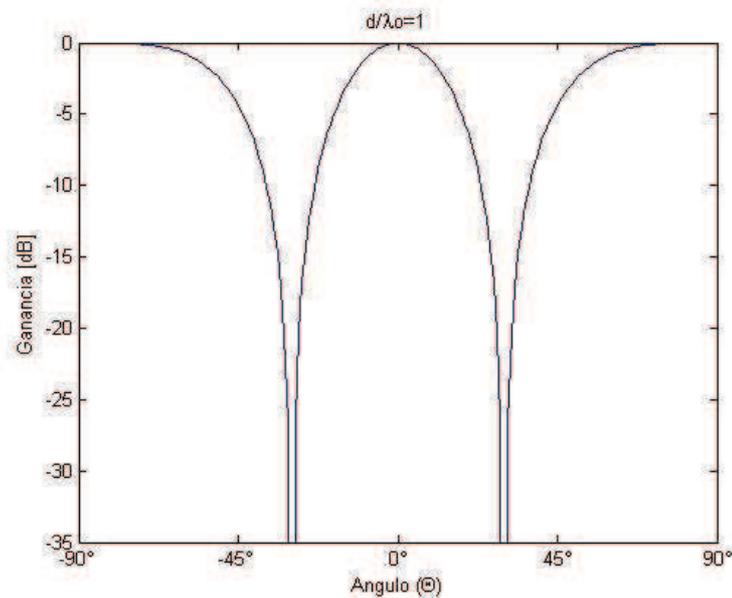


Figura 2.6 (b) Patrón de haz del arreglo de dos sensores, $d / \lambda_0 = 1$.

Conforme el espaciado entre elementos aumenta a $1.5\lambda_0$, el ancho del lóbulo principal decrece aún más, mejorando la resolución, los dos mínimos se mueven más y dos nuevos mínimos aparecen en $\theta = \pm 90^\circ$, como se muestra en la figura 2.6 (c). Incrementos mayores en el espaciado entre elementos resultan en la aparición de más mínimos de patrón y lóbulos laterales y un decremento mayor en el ancho del lóbulo principal.

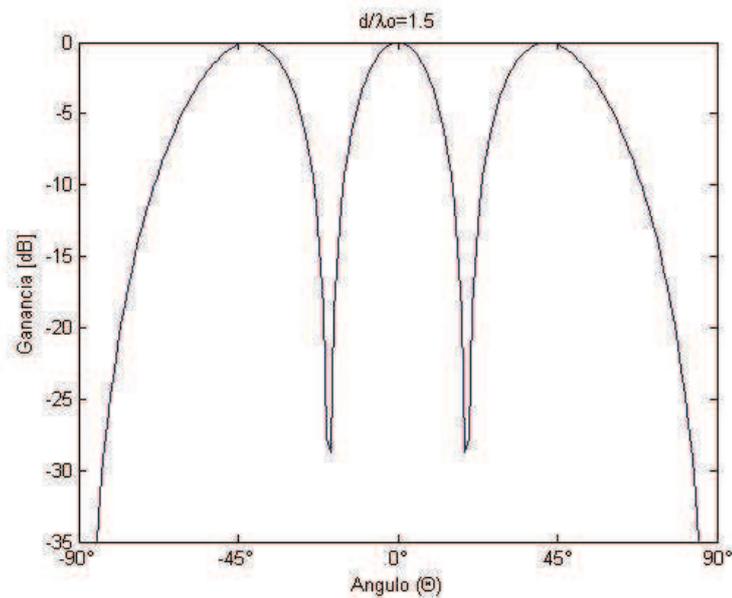


Figura 2.6 (c) Patrón de haz del arreglo de dos sensores, $d / \lambda_0 = 1.5$.

2.2.3 Arreglos Lineales

Para el arreglo lineal de N sensores, igualmente separados, de la figura 2.7, la respuesta total del arreglo se calcula considerando la suma fasorial de las contribuciones de las señales de cada elemento del arreglo, tal que

$$y(t) = \sum_{i=1}^N x(t) e^{j(i-1)\psi}$$

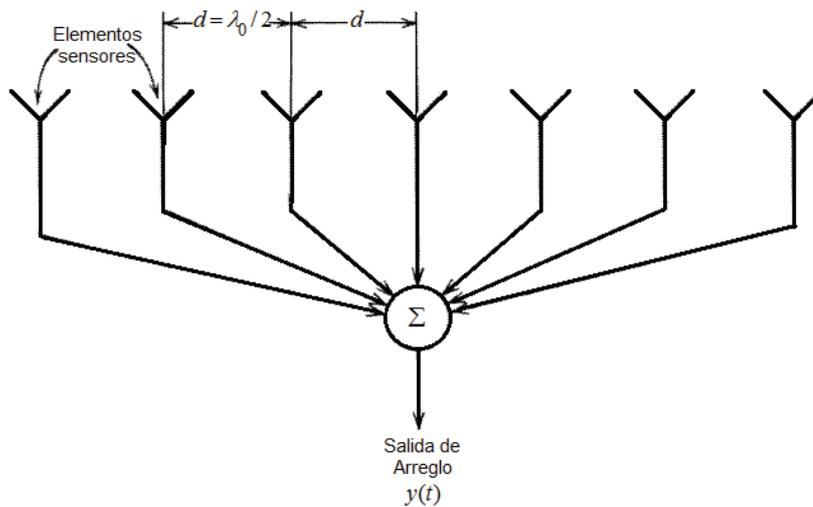


Figura 2.7 Arreglo lineal [10].

El patrón direccional en un plano conteniendo al arreglo puede ser encontrado considerando el factor del arreglo

$$A(\theta) = \sum_{i=1}^N e^{j(i-1)\psi} \quad (2.5)$$

El patrón direccional normalizado está dado por

$$G(\theta) = 10 \log_{10} \left\{ \frac{|A(\theta)|^2}{N^2} \right\} \quad (2.6)$$

La suma fasorial de la ecuación (2.5) es máxima cuando $\sin\theta = 0$ o $\theta = k2\pi$, cuando todos los fasores se alinean. Si la relación (d/λ_0) es lo suficientemente grande, entonces para N elementos los fasores estarán igualmente separados resultando en una suma igual a cero siempre que

$$2\pi \left(\frac{d}{\lambda_0} \right) \text{sen}\theta = 2\pi/N$$

ésto ocurre en una dirección θ_1 dada por

$$\text{sen}\theta_1 = \frac{1}{N} \left(\frac{\lambda_0}{d} \right) \quad (2.7)$$

Suponiendo que $L = (N-1)d$ es la longitud del arreglo, la ecuación (2.7) puede ser reescrita como:

$$\theta_1 = \arcsen \left(\frac{\lambda_0}{L+d} \right)$$

Manteniendo el espacio entre elementos en $(d/\lambda_0) = 0.5$ e incrementando el número de elementos omnidireccionales idénticos, el patrón direccional normalizado puede encontrarse con la ecuación (2.6). Las figuras 2.8 y 2.9 muestran los resultados para tres y cuatro elementos respectivamente [10].

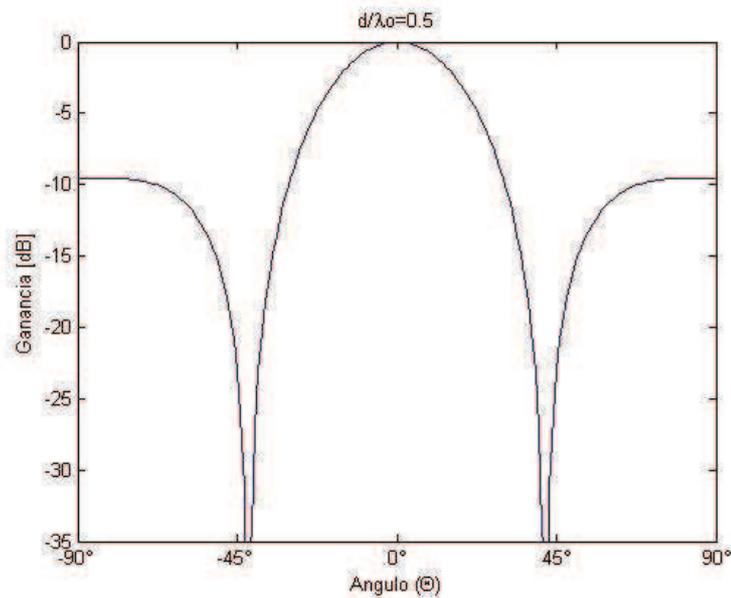


Figura 2.8 Patrón de haz para un arreglo lineal de tres sensores, $(d/\lambda_0) = 0.5$.

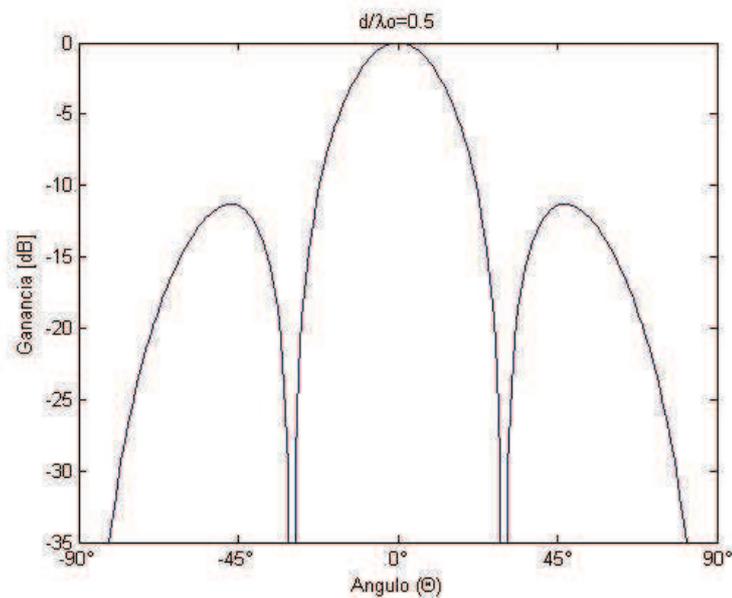


Figura 2.9 Patrón de haz para un arreglo lineal de cuatro sensores, $(d/\lambda_0) = 0.5$.

En las figuras 2.8 y 2.9 se observa que al aumentar el número de elementos el ancho del lóbulo principal disminuye, y el número de lóbulos laterales y mínimos aumenta.

Para ilustrar como el espacio entre elementos afecta el patrón direccional en un arreglo de siete elementos, la figura 2.10 de a) a d) muestra el patrón direccional en el plano angular para valores de (d/λ_0) en un intervalo de 0.1 a 1.0.

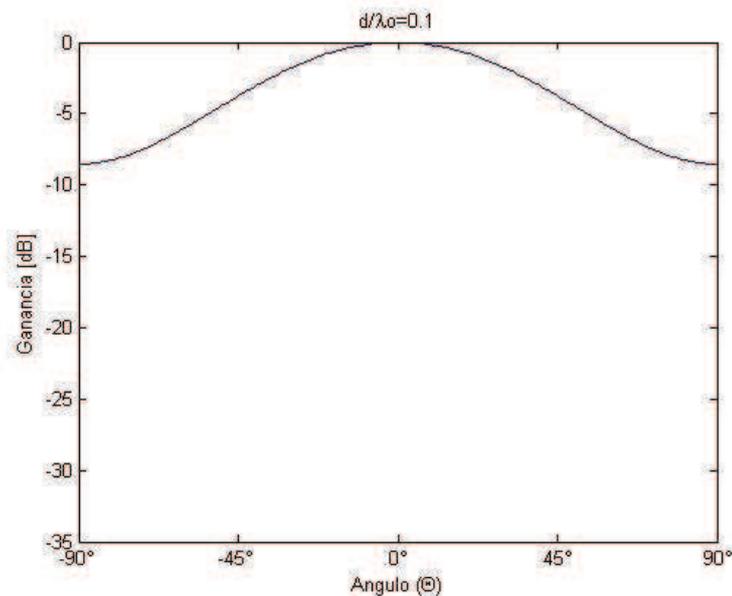


Figura 2.10 (a) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 0.1$.

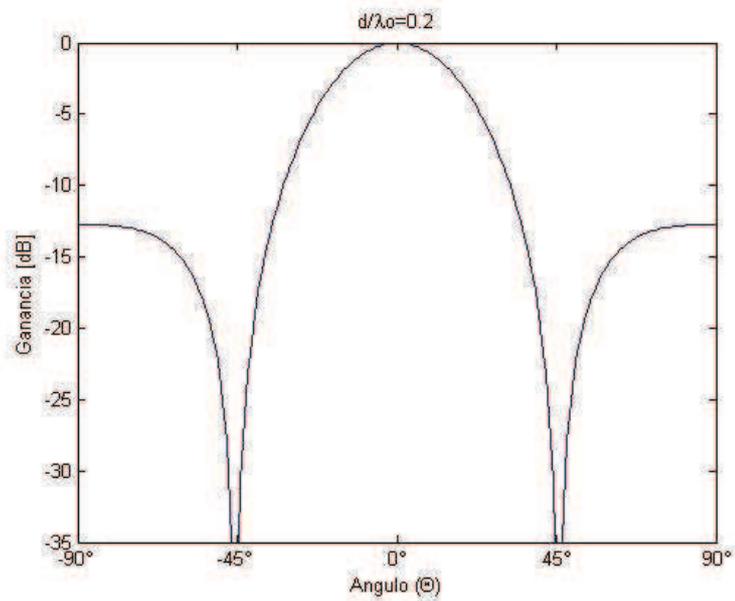


Figura 2.10 (b) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 0.2$.

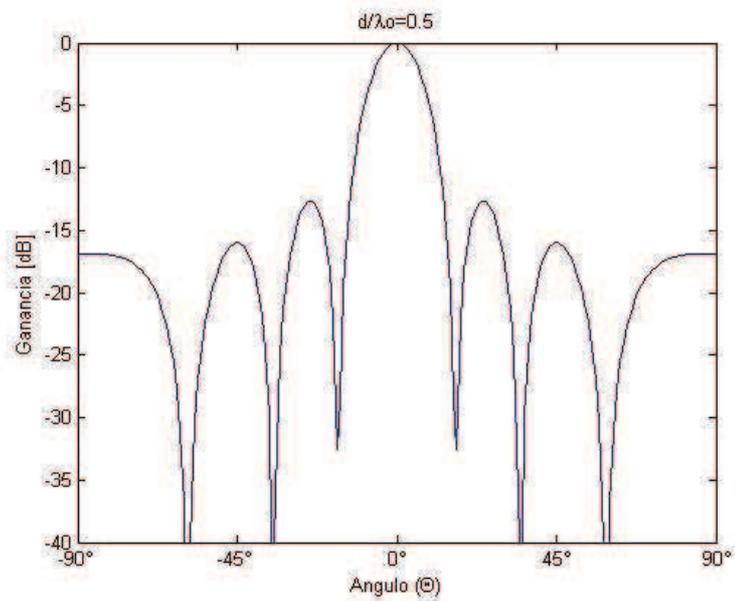


Figura 2.10 (c) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 0.5$.

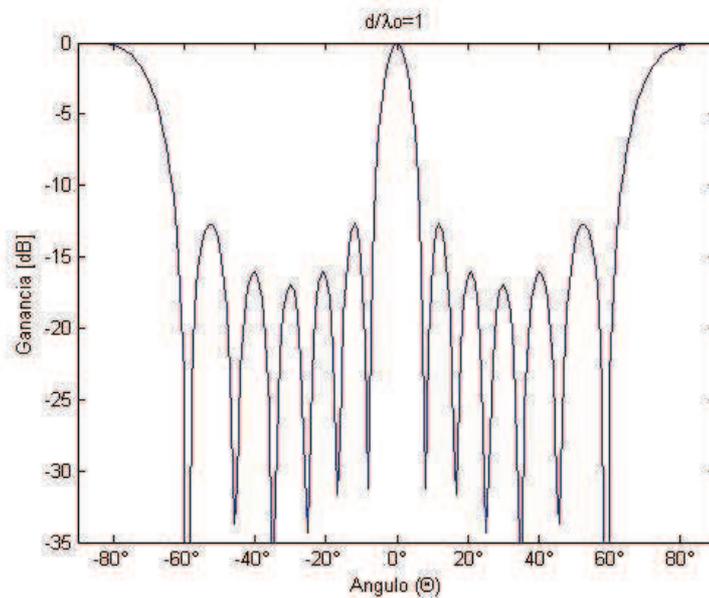


Figura 2.10 (d) Patrón direccional de un arreglo lineal de siete sensores, $(d/\lambda_0) = 1$.

Siempre que (d/λ_0) es menor que $\frac{1}{7}$, el patrón de haz no tiene mínimos exactos, como se observa en la figura 2.10 (a) que el mínimo de -8.5 dB que ocurre en $\theta = \pm 90^\circ$ cuando $(d/\lambda_0) = 0.1$.

Si el espacio entre elementos se incrementa más allá de $\frac{1}{7}$, los mínimos y lóbulos laterales (y lóbulos gratinados) empiezan a aparecer, con más mínimos y lóbulos apareciendo conforme (d/λ_0) aumenta y se produce un patrón de interferencia. Cuando $(d/\lambda_0) = 1$ (figura 2.10 (d)) los lóbulos en los extremos en $\theta = \pm 90^\circ$ tienen una ganancia igual a la del lóbulo principal dado que los siete fasores de las señales están exactamente alineados y se suman coherentemente.

La figura 2.11 muestra el cambio en el patrón de haz conforme (d/λ_0) varía con un número de siete sensores fijos.

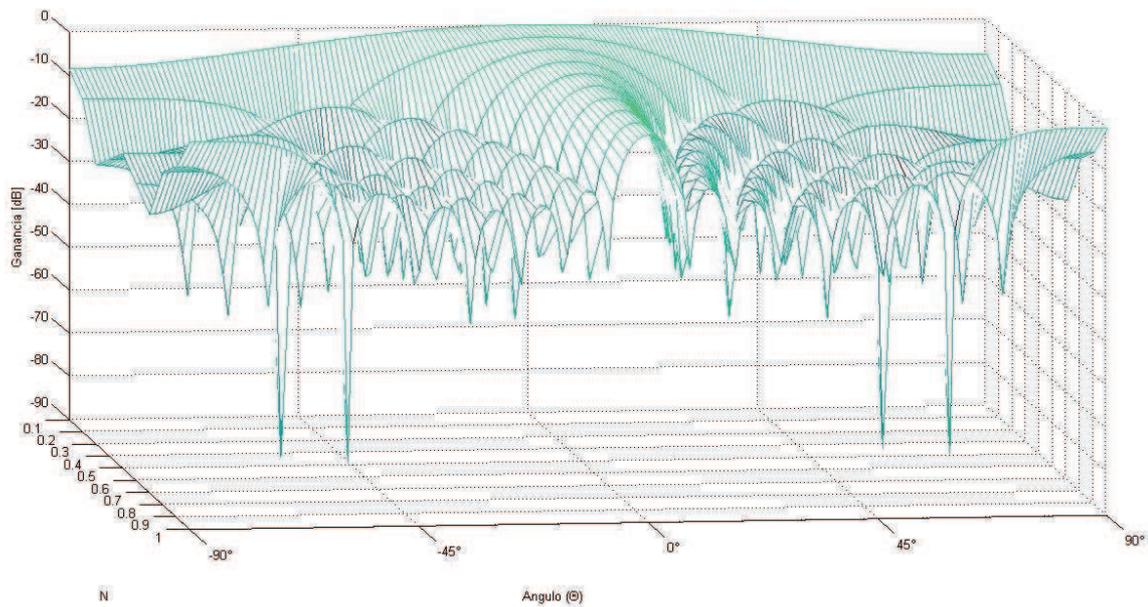


Figura 2.11 Arreglo de sensores $N=7$ y $d/\lambda_0 = 0.1 \rightarrow 1$.

La figura 2.12 muestra la variación en el patrón de haz al tener (d/λ_0) fijo y variar el número de sensores de 1 a 10.

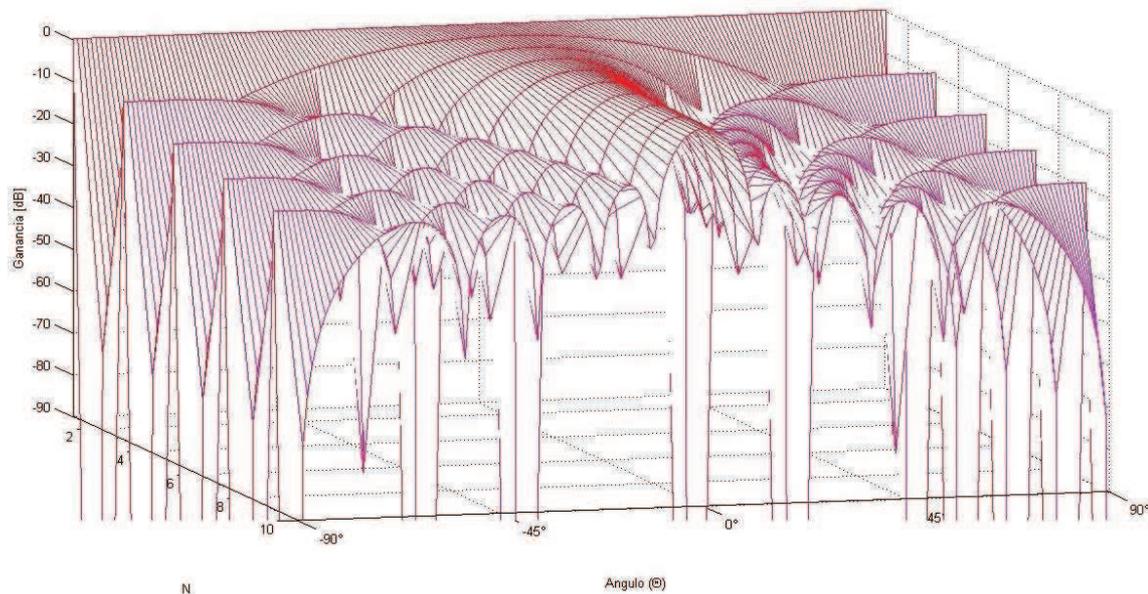


Figura 2.12 Arreglo de sensores $d/\lambda_0 = 0.5$, $N=1 \rightarrow 10$.

De las figuras anteriores se puede observar que el patrón de haz se modifica tanto al variar el número de sensores como la relación (d/λ_0) haciendo que el lóbulo lateral sea mas estrecho y que aparezcan más lóbulos laterales que en este caso son aproximadamente función de N .

Suponiendo un arreglo lineal como el de la figura 2.13, en el cual un corrimiento de fase (equivalente a un retraso de tiempo) de valor δ es insertado en el segundo elemento del arreglo, un corrimiento de fase de 2δ en el tercer elemento, y un corrimiento de fase de $(n-1)\delta$ en cada n -ésimo elemento siguiente [10].

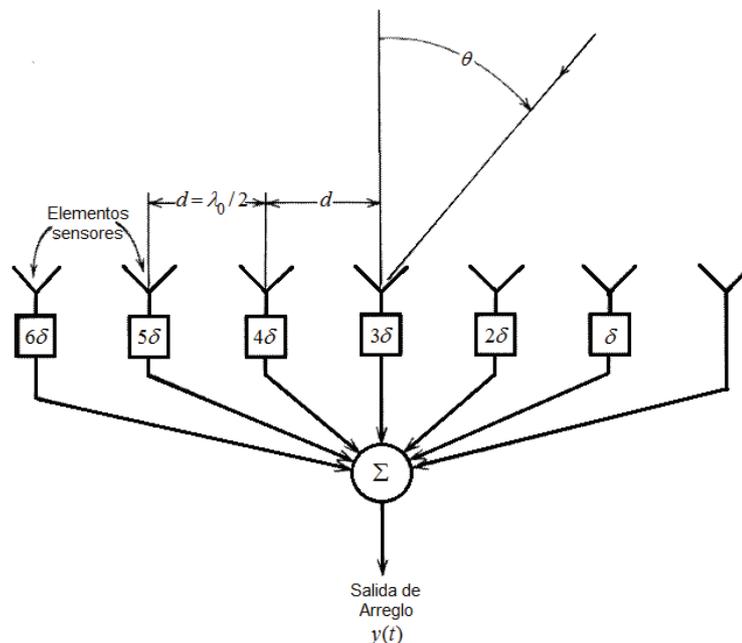


Figura 2.13 Arreglo lineal desplazado con elementos corridos en fase [10].

La inserción de esta secuencia de corrimientos de fase tiene el efecto de correr el lóbulo principal (o haz principal) dado por

$$\theta = \arcsen \left[\frac{1}{2\pi} \left(\frac{\lambda_0}{d} \right) \delta \right]$$

Así que el patrón direccional total ha sido “desplazado” por la inserción de la secuencia de corrimientos de fase. Este efecto se muestra en la figura 2.14 para un $\delta = 15^\circ$.

Se observa entonces que el patrón direccional puede ser desplazado hacia cualquier dirección angular y no solo en la dirección 0° , esto sin afectar su forma, lo cual es la base del filtrado espacial.

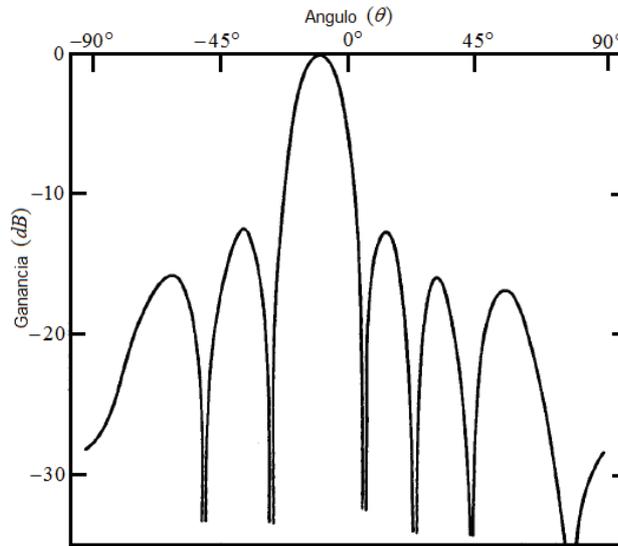


Figura 2.14 Patrón direccional de un arreglo lineal de siete elementos con $\delta = 15^\circ$ [10].

2.2.4 Arreglos Planares

Consideremos el arreglo de sensores discretos, planar, de forma rectangular, colocado en el plano $x-y$ que se muestra en la figura 2.15, donde el origen coordenado es seleccionado en el elemento central. Con N_x elementos en cada columna paralela al eje x , separadas un espaciamiento uniforme d_x , y N_y elementos en cada renglón paralela al eje y , con un espaciamiento uniforme d_y , el arreglo entero tiene $N_x \times N_y$ elementos [10].

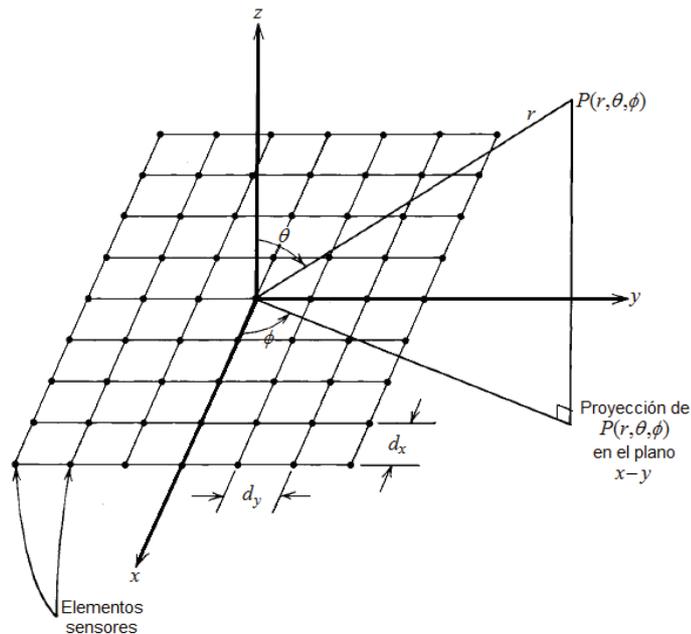


Figura 2.15 Arreglo planar de forma rectangular [10].

La suma fasorial de las contribuciones de las señales de cada elemento del arreglo en una sola columna es la misma que para el caso de un arreglo lineal y está dada por

$$y(t) = \sum_{i=1}^{N_x} x(t) e^{j(i-1)\psi_x}$$

donde

$$\psi_x = 2\pi \left(\frac{d_x}{\lambda_0} \right) \text{sen}\theta \cos\phi \text{ y } \psi_y = 2\pi \left(\frac{d_y}{\lambda_0} \right) \text{sen}\theta \text{sen}\phi$$

Se observa que la salida depende del ángulo azimuth proyectado ϕ y el ángulo de elevación θ . La suma fasorial total de las contribuciones de las señales de todos los elementos del arreglo está dada por

$$y(t) = \sum_{i=1}^{N_x} \sum_{k=1}^{N_y} x(t) e^{j(i-1)\psi_x} e^{j(k-1)\psi_y}$$

Consecuentemente, el patrón direccional del arreglo se encuentra considerando el término

$$A(\theta, \phi) = \sum_{i=1}^{N_x} \sum_{k=1}^{N_y} e^{j(i-1)\psi_x} e^{j(k-1)\psi_y} = A_x(\theta, \phi) A_y(\theta, \phi) \quad (2.8)$$

donde

$$A_x(\theta, \phi) = \sum_{i=1}^{N_x} e^{j(i-1)\psi_x}$$

$$A_y(\theta, \phi) = \sum_{k=1}^{N_y} e^{j(k-1)\psi_y}$$

De la ecuación (2.8) se puede observar que el patrón de haz del arreglo planar puede ser considerado como el producto de los factores de dos arreglos lineales, uno a lo largo del eje x , y otro a lo largo del eje y . Consecuentemente el requerimiento del espaciamiento de los elementos d_x y d_y es el mismo que en los arreglos lineales con el fin de evitar la formación de lóbulos gratinados.

Como en los arreglos lineales, los factores de arreglo de los renglones y columnas $A_x(\theta, \phi)$ y $A_y(\theta, \phi)$ del arreglo rectangular pueden ser desplazados arbitrariamente introduciendo las secuencias de cambio de fase apropiadas a los elementos de los renglones y columnas.

Además de los arreglos planares rectangulares, existen arreglos circulares y elípticos, que son muy útiles cuando se desea tener simetría angular en una operación bidimensional [10].

2.2.5 Desempeño del Arreglo

Se ha encontrado que un arreglo de sensores puede producir una resolución y sensibilidad mucho mejores que un solo sensor y ofrece la posibilidad de mejorar el desempeño de la recepción a través de la supresión simultánea de interferencias indeseables preservando una señal deseada. Más aun, en la selección de una configuración de arreglo, es necesario tener presente los siguientes factores críticos:

- Capacidad de resolución.
- Cobertura angular.
- Número de elementos del arreglo.
- Nivel de lóbulos laterales.
- Capacidad de cómputo del sistema utilizado.

Si el número de elementos del arreglo es pequeño, obtener un mínimo en el patrón que suprime una señal de interferencia puede degradar significativamente la sensibilidad del arreglo a lo largo de la región deseada de cobertura angular. Una forma de tolerar esa degradación de cobertura manteniendo la capacidad de supresión, es la de asegurar una resolución mayor usando un número de elementos mayor; sin embargo, conforme el número de elementos aumenta, el costo y la complejidad del arreglo también incrementan. Por eso existe una relación fundamental entre la capacidad de resolución, el nivel de lóbulos laterales y el número de elementos del arreglo para una región de cobertura deseada específica; esta relación se considera para arreglos lineales y planares [10].

2.3 Resumen

Se presentaron los conceptos de las ondas esféricas en el campo lejano y cercano, así como su representación teórica. También se observó el patrón de directividad para fuentes simples considerando varias condiciones, como diferente número de fuentes y relación entre distancia de separación y longitud de onda.

Los arreglos de sensores pueden tener diferentes configuraciones en cuanto a la posición que tengan los sensores en el arreglo, dentro de estas configuraciones se presentaron las respuestas que tienen un arreglo de dos sensores, un arreglo lineal de varios sensores, y un arreglo planar.

La respuesta del arreglo depende del espacio que hay entre los sensores, siendo éste un parámetro que la define a las señales incidentes captadas.

La respuesta de un arreglo planar puede ser considerada como la suma de las respuestas de varios arreglos lineales.

Capítulo 3

Técnicas y Métodos de Formador de Haz



En este capítulo se presentan diferentes tipos y métodos de formador de haz, describiendo los formadores de haz fijos y adaptables; así como sus características y esquemas generales. Se presentan las ecuaciones y procesos involucrados en su implementación, mencionando sus ventajas y desventajas en comparación con los demás métodos.

3.1 Fundamentos y Tipos de Formador de Haz

El Beamforming o formador de haz es una técnica que ha sido utilizada por muchas décadas en telecomunicaciones, radar y en aplicaciones de arreglos de antenas; sin embargo, recientemente ha sido adoptado para aplicaciones acústicas de banda ancha con el fin de mejorar la voz.

Un formador de haz es un conjunto de sensores posicionados en un arreglo cuyas salidas pueden ser dirigidas u orientadas electrónicamente. La orientación electrónica usa técnicas de procesamiento de señales que explotan la información espacial en las señales del sensor, requiriendo operaciones de filtrado espacio-temporales que pueden o no ser adaptables.

Un esquema básico de un formador de haz digital se muestra en la figura 3.1, donde las fuentes de señales emiten ondas esféricas que eventualmente llegan al arreglo de sensores, que comprende micrófonos lineales para extraer información espacial o temporal de las ondas. La información espacial obtenida de las ondas esféricas es usada por el formador de haz para selección y separación espacial [1].

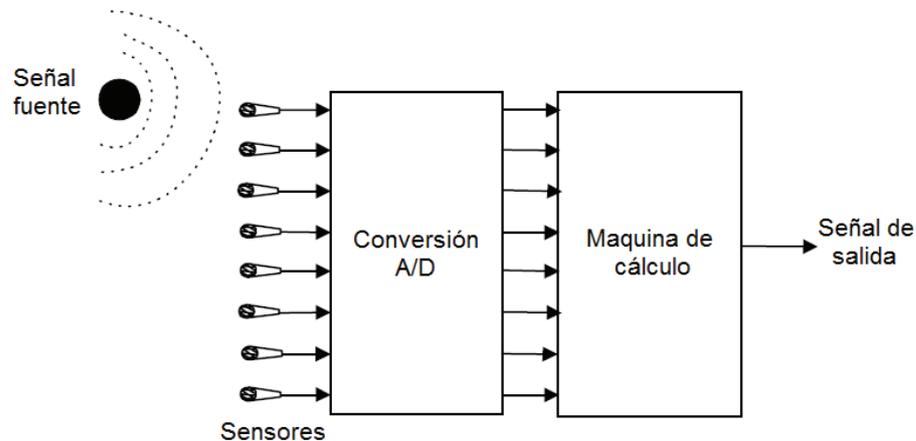


Figura 3.1. Formador de haz digital básico [1].

El formador de haz usa la fase de las ondas de las señales recibidas por el arreglo de sensores. Las señales recibidas en el arreglo de sensores son enviadas a un sistema de proceso, el cual las digitaliza y las envía a un conjunto de filtros digitales. Debido a esta lógica de proceso, el formador de haz se puede diseñar para que realice una selección angular o espacial de las señales u ondas que arriban.

Capítulo 3. Técnicas y Métodos de Formador de Haz.

El formador de haz es la técnica más común para selección espacial, siendo muy útil para supresión de distorsiones y de reverberación. Los formadores de haz acústicos deben manejar el ancho de banda correspondiente al audio y de la voz [1].

Las técnicas de formador de haz pueden ser clasificadas como independientes o dependientes de los datos. Los formadores de haz independientes de los datos, o fijos, son llamados así porque sus parámetros son fijos durante la operación. Por otro lado, los formadores de haz dependientes de los datos o adaptables, actualizan continuamente sus parámetros basados en las señales recibidas. Existen diferentes métodos de formador de haz y a continuación se presentan sus principales características [9].

Técnica	Tipo	Ventajas	Desventajas
Retardo y suma	Fijo	Simplicidad	Bajo desempeño en frecuencia Banda angosta
Subarreglo retardo y suma	Fijo	Banda ancha	Bajo desempeño en frecuencia
Superdirectividad	Fijo	Ganancia de arreglo optimizada	Asume ruido difuso
Superdirectividad de campo cercano	Fijo	Ganancia de arreglo optimizada Fuentes de campo cercano Buen desempeño a bajas frecuencias	Asume ruido difuso Asume ruido en el campo cercano
Cancelador generalizado de lóbulos laterales	Adaptable	Se adapta a condiciones de ruido Minimiza la potencia del ruido de salida Restricción fuerte en la señal	Bajo desempeño en frecuencia Puede distorsionar
AMNOR	Adaptable	Se adapta a condiciones de ruido Minimiza la potencia del ruido de salida Restricción suave en la señal Nivel de distorsión controlado	Bajo desempeño en frecuencia Complejidad de detección de silencio Algo de distorsión
Post filtrado	Adaptable	Se adapta a condiciones de ruido Mejora la salida del formador de haz	Puede distorsionar la señal

Tabla 3.1. Métodos de formador de haz y características.

3.2 Formadores de haz fijos

Los formadores de haz fijo mantienen sus parámetros independientemente de los datos y las operaciones que se realicen con éstos. Existen varios métodos fijos de formador de haz que se clasifican de acuerdo a las operaciones que forman parte de ellos.

3.2.1 Formador de haz de retardo y suma

La técnica más simple de un formador de haz en un arreglo de micrófonos, es el formador de haz de retardo y suma, aplicando pesos de fase, que son valores dependientes de la fase de la señal de arribo, a los sensores o canales de entrada; variando estos pesos se puede mover el lóbulo principal del patrón de dirección a una dirección deseada. Considerando los pesos de fase φ_n [9]

$$\varphi_n = \frac{-2\pi(n-1)d \cos \phi' f}{c}$$

obtenemos el patrón direccional

$$D(f, \phi) = \sum_{n=1}^N e^{j \frac{2\pi f (n-1) d (\cos \phi - \cos \phi')}{c}} \quad (3.1)$$

y el lóbulo principal del patrón direccional se moverá a la dirección $\phi = \phi'$.

El retraso en fase negativo en el dominio de la frecuencia puede ser implementado aplicando retardos de tiempo en las entradas de los sensores, donde el retraso del n -ésimo sensor está dado por

$$\tau_n = \frac{(n-1)d \cos \phi'}{c}$$

que es el tiempo que toma la onda plana en viajar entre el sensor de referencia y el n -ésimo sensor.

En el formador de haz de retardo y suma, las entradas de los sensores en el dominio del tiempo primero son retardadas τ_n segundos y después sumadas para tener una sola salida del arreglo. Usualmente, a cada canal se le da un peso de amplitud igual en la suma, así el patrón tiene ganancia unitaria en la dirección deseada. Esto lleva a los pesos complejos de canal [9]

$$w_n(f) = \frac{1}{N} e^{j \frac{-2\pi f}{c} (n-1) d \cos \phi'}$$

expresando la salida del arreglo como la suma de los canales con los pesos complejos obtenemos

$$y(f) = \frac{1}{N} \sum_{n=1}^N x_n(f) e^{j \frac{-2\pi f}{c} (n-1) d \cos \phi'}$$

equivalentemente a una suma de señales x_n retardadas en el dominio del tiempo tenemos

$$y(t) = \sum_{n=1}^N x_n(t - \tau_n)$$

3.2.2 Formador de haz de filtrado y suma

El formador de haz de retardo y suma pertenece a una clase más general conocida como formadores de haz de filtrado y suma, en los cuales la amplitud y los pesos de fase son dependientes de la frecuencia. En la práctica, la mayoría de los formadores de haz son una clase de formador de haz de filtrado y suma. La salida de un formador de haz de filtrado y suma es [9]

$$y(f) = \sum_{n=1}^N w_n(f) x_n(f)$$

Algunas veces es conveniente usar álgebra matricial para simplificar la notación cuando se describen las técnicas de arreglos de micrófonos. La ecuación anterior puede ser reescrita usando la notación matricial como

$$\mathbf{y}(f) = \mathbf{w}(f)^T \mathbf{x}(f)$$

donde el vector de pesos $\mathbf{w}(f)$ y el vector de datos $\mathbf{x}(f)$, se definen como

$$\mathbf{w}(f) = [w_1(f) \cdots w_n(f) \cdots w_N(f)]^T$$

$$\mathbf{x}(f) = [x_1(f) \cdots x_n(f) \cdots x_N(f)]^T$$

Donde $[\cdot]^T$ denota la transpuesta [9].

La figura 3.2 muestra la estructura de un formador de haz de filtrado y suma.

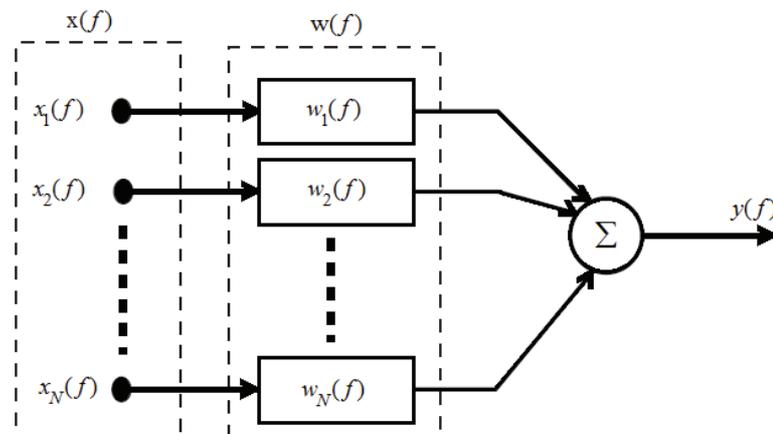


Figura 3.2. Estructura de un formador de haz de filtrado y suma [9].

3.2.3 Formador de haz de subarreglo

De la ecuación del patrón direccional de un arreglo de sensores uniformemente espaciados, ecuación (3.1), se observa que las características de la respuesta del arreglo dependen de la frecuencia de interés, el espacio entre elementos y el número de elementos del arreglo. La dependencia en la frecuencia de operación significa que las características de la respuesta (ancho de haz, nivel de lóbulos laterales) solo permanecerán constantes para señales de banda angosta. Sin embargo, la voz, es una señal de banda ancha, lo que significa que un diseño con un arreglo lineal sencillo es inadecuado si se desea un patrón de haz invariante con la frecuencia [9].

Un método simple para cubrir señales de banda ancha es implementar el arreglo como una serie de subarreglos lineares con espaciado uniforme, estos subarreglos son diseñados para obtener características de respuesta deseadas para un intervalo de frecuencia dado. A medida que la frecuencia aumenta, se requiere una longitud de arreglo más pequeña para mantener un ancho de haz constante. Además, para asegurar que el nivel de lóbulos laterales permanece igual a lo largo de diferentes bandas de frecuencias, el número de elementos en cada subarreglo debe mantenerse igual. Los subarreglos son generalmente implementados de forma combinada, para que cualquiera de los sensores pueda ser usado en más de un subarreglo. Cada subarreglo es restringido a un intervalo de frecuencia diferente aplicando filtros pasa banda, y la salida total del arreglo está formada por la combinación de las salidas de los subarreglos limitados en banda.

La estructura de un formador de subarreglo se muestra en la figura 3.3.

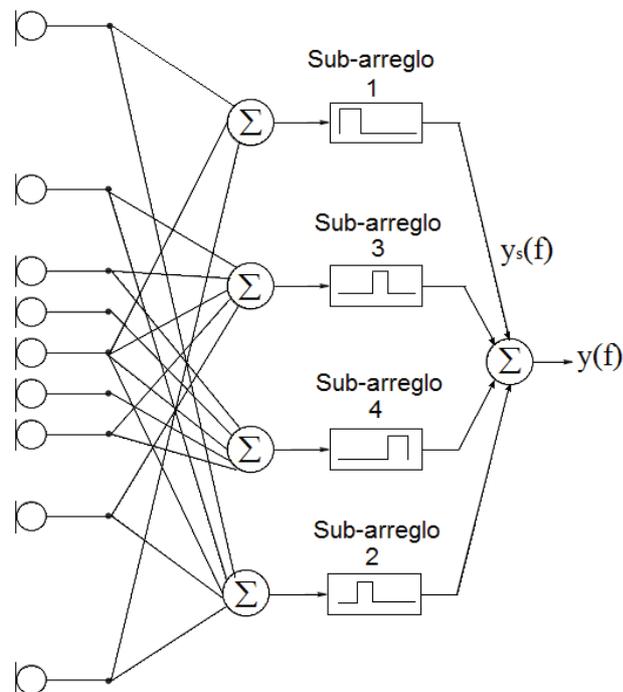


Figura 3.3. Estructura de subarreglo combinado [9].

Para un formador de haz de subarreglo para banda ancha, las señales de los sensores o canales del formador de haz son filtrados con filtros pasa banda, entre la frecuencia más alta y más baja especificadas para cada sub banda. A la salida de cada filtro de canal tenemos

$$v_{s,i} = w_{s,i}(f) x_i(f)$$

donde $x_i(f)$ es la entrada al canal i del arreglo, y el sub índice s representa el índice del arreglo. La salida del subarreglo s está dada por la suma de los canales

$$y_s(f) = \sum_{i=1}^N v_{s,i}(f)$$

donde existen N micrófonos en el arreglo. La suma en cada subarreglo se calcula hasta N por simplicidad de notación, aunque en la práctica solo los canales que pertenecen a cada subarreglo son usados.

La salida total del arreglo es calculada como

$$y(f) = \sum_{s=1}^S y_s(f)$$

Donde existen S subarreglos.

3.3 Formadores de haz adaptables

Los formadores de haz adaptables van cambiando sus parámetros en cada iteración dependiendo de los valores de los datos de entrada. Existen diferentes estructuras de algoritmos adaptables y se clasifican dependiendo del efecto que producen en la salida total del sistema.

3.3.1 Cancelador de lóbulos laterales generalizado (GSC)

Las técnicas de formador de haz dependientes de los datos tratan de filtrar adaptablemente las señales recibidas con el fin de dejar pasar la señal que viene de la dirección deseada mientras rechaza ruido proveniente de otras direcciones.

La mayoría de las técnicas adaptables recaen en la minimización del error cuadrático medio entre una señal de referencia, que está altamente correlacionada con la señal deseada, y la señal de salida. Sin embargo, el algoritmo de la media de los mínimos cuadrados (LMS) puede degradar la señal deseada ya que busca únicamente minimizar el error cuadrático medio y no fija condiciones acerca de la distorsión de la señal deseada. La técnica de formador de haz adaptable más famosa, que toma en cuenta esta limitación, es conocida como el algoritmo de Frost. Este algoritmo toma el proceso de la estimación del filtro como un problema de minimización de la media de los mínimos cuadrados con restricciones. La solución minimiza el error cuadrático medio mientras mantiene una función de transferencia específica para la señal deseada. Esta restricción es normalmente diseñada para asegurar que la respuesta a la señal deseada tenga ganancia constante y fase lineal. El algoritmo de Frost pertenece a una clase de formadores de haz conocidos como formadores de haz linealmente restringidos con varianza mínima (LCMV) [9].

Tal vez la técnica comúnmente usada de formadores de haz LCMV es el cancelador de lóbulos laterales (GSC). El GSC es una estructura de formador de haz que puede ser usada para implementar una variedad de procesadores adaptables de arreglos linealmente restringidos, incluyendo el algoritmo de Frost. Como se muestra en la figura 3.4, el GSC separa al formador de haz adaptable en dos caminos principales de proceso. El primero implementa un formador de haz fijo estándar, con restricciones en la señal deseada. El segundo camino es la porción adaptable que provee un conjunto de filtros que minimizan la potencia en la salida. La señal deseada es eliminada de este segundo camino por una matriz de bloqueo B, asegurando que es la potencia del ruido la que es minimizada [9].

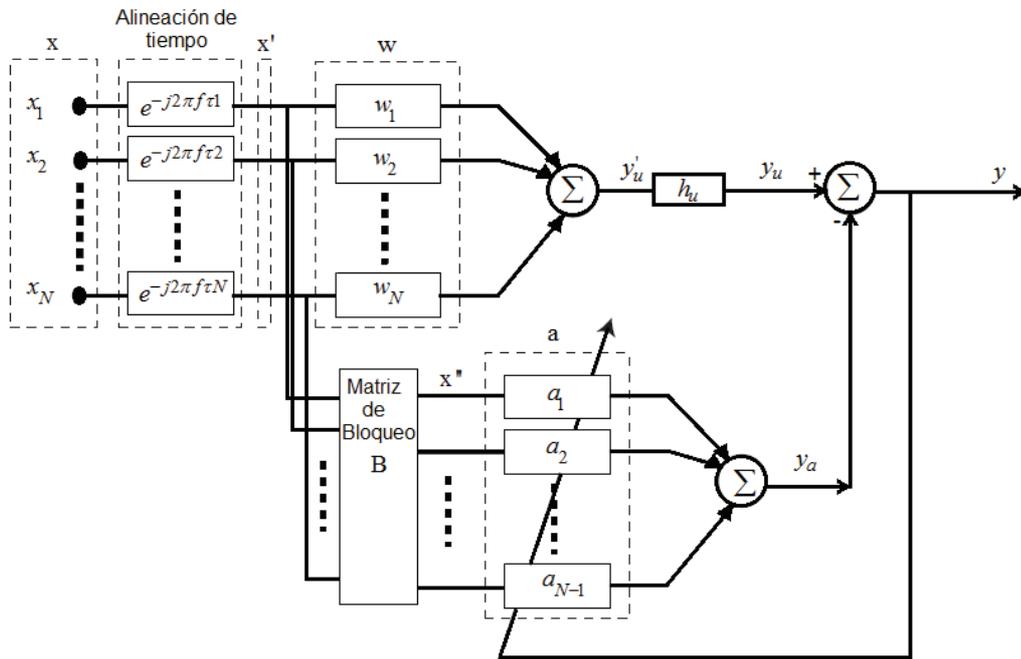


Figura 3.4. Estructura del cancelador de lóbulos laterales generalizado [9].

Examinando la estructura superior, las entradas primero son alineadas y después pasan a través de un formador de haz de filtrado y suma para tener una señal de formador de haz fijo $y'_u(f)$ de la forma

$$y'_u(f) = \mathbf{w}(f)^T \mathbf{x}'(f)$$

donde

$$\mathbf{w}(f) = [w_1(f) \cdots w_n(f) \cdots w_N(f)]^T$$

son los pesos de amplitud fijos para cada uno de los N canales y

$$\mathbf{x}'(f) = [x'_1(f) \cdots x'_n(f) \cdots x'_N(f)]^T$$

son las señales de entrada alineadas en el tiempo.

La salida del formador de haz fijo es filtrada por el filtro de restricción h_u que asegura una ganancia y fase específica para la señal deseada. La salida del camino superior está dada por

$$y_u(f) = h_u(f) y'_u(f)$$

El camino inferior de la estructura es la porción adaptable, la cual consiste de dos partes principales, la primera de ellas es la matriz de bloqueo, B , cuyo propósito es remover la señal deseada en este camino. Como la señal deseada es común a todos los canales de entrada alineados en el tiempo, el bloqueo ocurre si los renglones de la matriz de bloqueo suman cero. Si \mathbf{x}'' denota las señales a la salida de la matriz de bloqueo, entonces

$$\mathbf{x}''(f) = B\mathbf{x}'(f)$$

Donde cada renglón de la matriz de bloqueo suma cero, y los renglones deberían ser linealmente independientes. Dado que \mathbf{x}' puede tener máximo $N-1$ componentes linealmente independientes, el número de renglones debe ser $N-1$ o menos. La matriz de bloque estándar de Griffiths-Jim es

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \ddots & \cdots & \vdots \\ 0 & \cdots & 0 & 1 & -1 & 0 \\ 0 & \cdots & 0 & 0 & 1 & -1 \end{bmatrix}$$

Después de la aplicación de la matriz de bloqueo, \mathbf{x}'' es filtrada adaptablemente y sumada para obtener la salida del camino inferior $y_a(f)$. Si denotamos a los filtros adaptables del camino inferior como \mathbf{a} , entonces tenemos

$$y_a(f) = \mathbf{a}(f)^T \mathbf{x}''(f)$$

Debido a la matriz de bloqueo, la salida del camino inferior solo contiene señales de ruido. La salida total del sistema es calculado como la diferencia entre las salidas del camino superior e inferior como

$$y(f) = y_u(f) - y_a(f)$$

Debido a que el camino superior contiene la estimación de la señal deseada restringida y el camino inferior solo contiene términos de ruido e interferencia, el proceso de encontrar el conjunto de coeficientes de los filtros \mathbf{a} que minimizan la potencia de la salida y es equivalente a encontrar la solución de formador de haz linealmente restringido con mínima varianza. Como la

señal está restringida en el camino superior, el algoritmo LMS sin restricción puede ser usado para adaptar los coeficientes del filtro del camino inferior [9]

$$\mathbf{a}_{k+1}(f) = \mathbf{a}_k(f) + \mu y_k(f) \mathbf{x}_k''(f) \quad (3.2)$$

donde μ es el tamaño de paso y k es el número de muestra.

El algoritmo GSC es una estructura flexible debido a la separación del formador en una porción fija y una adaptable, y es el formador de haz adaptable más utilizado. En la práctica, el GSC puede causar un grado de distorsión en la señal deseada, debido a un fenómeno conocido como fuga de señal. La fuga de señal ocurre cuando la matriz de bloqueo falla en remover toda la señal deseada del camino cancelador de ruido inferior. Esto puede ser particularmente problemático para señales de banda ancha, como la voz, ya que es difícil asegurar la cancelación perfecta de la señal a través de un intervalo amplio de frecuencia [9].

3.3.2 AMNOR

A pesar de que los algoritmos LCMV son teóricamente poderosos, usualmente encuentran un número de problemas en la práctica, debido a la alta restricción de un valor permisible para la función de transferencia de la señal deseada. Las técnicas LCMV pueden fallar para reducir suficientemente el nivel de ruido debido a las restricciones en la elección de los filtros. La evaluación del sistema auditivo humano muestra que un cierto nivel de distorsión en la señal deseada puede ser tolerado y en algunas situaciones puede ser permisible, y hasta deseable, para permitir cierta distorsión de señal con el fin de alcanzar una mejor reducción de ruido.

Una técnica que incorpora una restricción "suave" es el llamado sistema AMNOR (sistema adaptable de arreglo de micrófonos para reducción de ruido) fue propuesto por Kaneda. En lugar de permitir solo una respuesta para la señal deseada, el sistema adopta una restricción suave que permite una clase de respuestas cuya degradación es menor que un nivel permisible preestablecido [9]. La figura 3.5 muestra la estructura del sistema AMNOR.

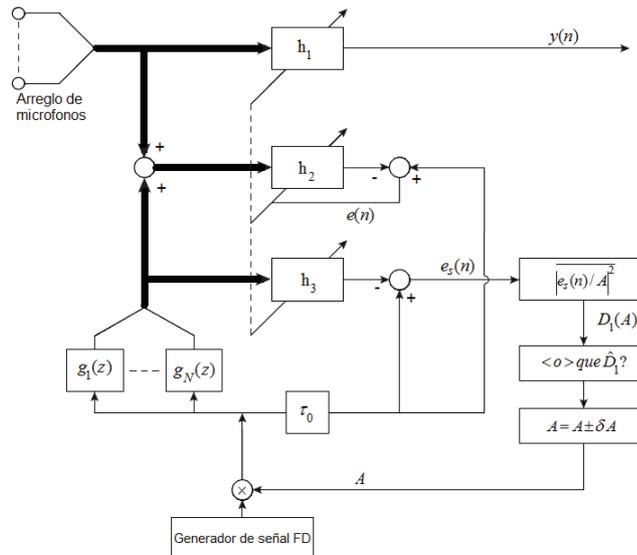


Figura 3.5. Estructura del sistema AMNOR [9].

El sistema está compuesto por tres bloques de filtros adaptables: h_1 , h_2 , y h_3 .

El bloque de filtro h_1 contiene los filtros de formador de haz que son aplicados a la entrada multicanal para tener la salida del sistema $y(n)$. La respuesta al impulso del camino acústico entre la fuente y el elemento del arreglo i es modelado por el filtro $g_i(z)$. La función de transferencia del formador de haz a la respuesta deseada es

$$F(z) = \sum_{i=1}^N h_{1,i}(z) g_i(z)$$

El segundo conjunto de filtros h_2 es actualizado para satisfacer el criterio de la potencia mínima de ruido en la salida, para un cierto nivel de degradación en la señal deseada. La adaptación ocurre durante periodos de solo ruido, cuando una señal deseada ficticia es introducida al sistema. La señal deseada ficticia es una señal de ruido blanco con potencia unitaria que es magnificada por un factor de amplitud variable A . La señal deseada ficticia es filtrada por las respuestas al impulso del camino acústico $g_i(z)$ con el fin de simular la presencia de una señal deseada conocida durante los periodos de sólo ruido.

El error cuadrático medio en la salida está relacionado con la degradación de la señal deseada D_1 , la potencia del ruido de la salida D_2 y la amplitud de la señal deseada ficticia A , de acuerdo a

$$\overline{|e(n)|^n} = A^2 \cdot D_1 + D_2$$

D_1 y D_2 son funciones monótonicamente decreciente y creciente de A , respectivamente. El nivel de degradación y la potencia del ruido de salida pueden ser ajustados variando el parámetro A .

El tercer conjunto de filtros h_3 es usado para estimar la degradación de respuesta D_1 , con el fin de adaptar la amplitud de la señal deseada ficticia para alcanzar los niveles deseados de degradación y potencia de ruido de salida.

La técnica AMNOR tiene las limitaciones de requerir de la detección precisa de voz/silencio y el conocimiento de las respuestas al impulso de los caminos acústicos entre la fuente y cada micrófono. Debido a los filtros fijos durante los periodos de voz, la técnica implícitamente asume características de ruido que varían lentamente. En la práctica, los caminos acústicos son modelados usando retardos de tiempo simples como en el formador de haz de retardo y suma [9].

3.3.3 Post-filtrado

En la práctica, el formador de haz básico de filtrado y suma muy pocas veces exhibe el nivel de desempeño que la teoría promete y un mejoramiento posterior es deseable. Un método para mejorar el desempeño del sistema es agregar un filtro a la salida del sistema.

La incorporación de un post filtro con un formador de haz permite el uso de la información obtenida en el filtrado espacial para permitir un filtrado de frecuencia efectivo en la señal. Usando el mejoramiento tanto en el dominio espacial como en el de la frecuencia, el uso de información acerca de la señal es maximizado, donde esta información es sólo la dirección de arribo de la señal [9].

La figura 3.6 muestra la estructura de un formador de haz con post filtrado.

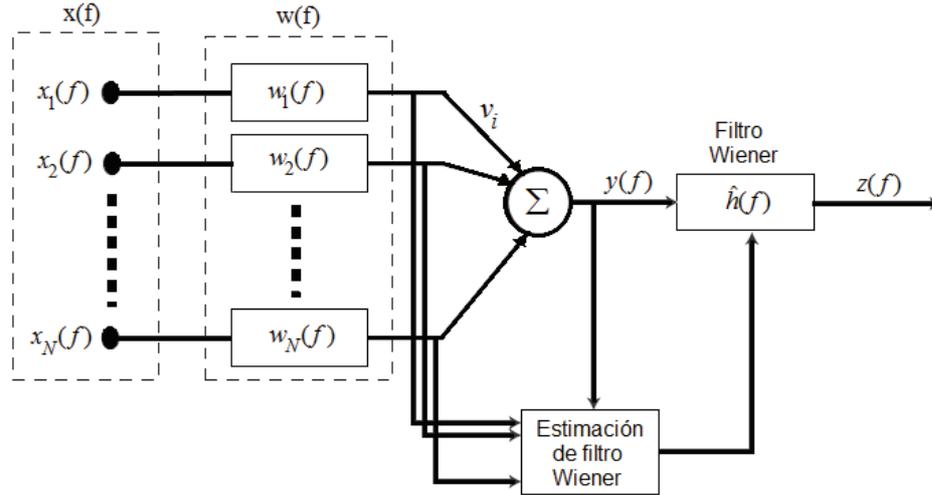


Figura 3.6. Formador de haz de filtrado y suma con post filtro [9].

A la salida de los filtros de cada sensor o canal tenemos las entradas de canal alineadas en el tiempo

$$v_i(f) = w_i(f) x_i(f)$$

Estas señales contienen una versión alineada de la señal deseada más una componente de ruido

$$v_i(f) = s(f) + n_i(f)$$

Donde s es la señal deseada y n_i es el ruido en el micrófono i .

La expresión general de un filtro Wiener para un arreglo de micrófonos está dado por

$$h_{opt}(f) = \frac{\Phi_{ss}(f)}{\Phi_{ss}(f) + \Phi_{\bar{nn}}(f)}$$

donde $\Phi_{ss}(f)$ y $\Phi_{\bar{nn}}(f)$ son respectivamente la densidad auto espectral de la señal deseada $s(f)$ y el ruido en la salida del formador de haz $\bar{n}(f)$.

Un problema común con los filtros Wiener es la estimación de las densidades auto espectrales de la señal y del ruido. La aproximación multicanal provee una solución interesante a este problema bajo las suposiciones que [9]:

1. La señal que arriba a cada micrófono puede ser modelada como la suma de la señal deseada más ruido.
2. El ruido $n_i(n)$ y la señal deseada $s(n)$ no están correlacionados.
3. La densidad espectral de potencia del ruido es la misma en cada micrófono $\Phi_{n_i n_i}(f) = \Phi_{nn}(f), i=1, \dots, N$.
4. El ruido no está correlacionado entre diferentes micrófonos $\Phi_{n_i n_j}(f) = 0, i \neq j$.
5. Las señales de entrada son restablecidas con perfecto alineamiento de fase con $s(n)$.

Tenemos que

$$\Phi_{v_i v_i}(f) = \Phi_{ss}(f) + \Phi_{nn}(f)$$

y

$$\Phi_{v_i v_j}(f) = \Phi_{ss}(f)$$

y promediando estas densidades espectrales, podemos estimar la ecuación del filtro Wiener como

$$\hat{h}(f) = \frac{\sum_{i=1}^N |w_i(f)|^2}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N w_i(f) w_j^*(f)} \frac{\Re \left\{ \sum_{i=1}^{N-1} \sum_{j=i+1}^N \hat{\Phi}_{v_i v_j}(f) \right\}}{\sum_{i=1}^N \hat{\Phi}_{v_i v_i}(f)}$$

El operador real $\Re\{\cdot\}$ es usado debido a que $\Phi_{ss}(f)$ debe ser necesariamente real. La salida total del sistema es

$$z(f) = \hat{h}(f) y(f)$$

donde $y(f)$ es la salida del formador de haz.

La efectividad de dicho post filtro depende de la efectividad del formador de haz, si el formador de haz es efectivo, el post filtro mejorará aún más la salida del sistema. Sin embargo, en el caso de que el formador de haz no sea efectivo, el post filtro, estando intrínsecamente ligado al desempeño del formador de haz, será similarmente inefectivo [9].

3.4 Formador de haz con descomposición Wavelet

En el análisis de señales no estacionarias usando la transformada de Fourier (FT) o la transformada corta de Fourier (STFT), los resultados no son satisfactorios; para obtener mejores resultados se puede usar un análisis con wavelets. Una ventaja del análisis con wavelets es el realizar análisis locales. El análisis con wavelets es capaz de revelar aspectos de las señales que otras técnicas no pueden hacer, tales como tendencias, puntos de ruptura, discontinuidades, etc. En comparación con la STFT, el análisis con wavelets hace posible realizar un análisis multiresolución, que presenta una resolución tiempo–frecuencia fija. Usando una solución llamada análisis multiresolución (MRA) es posible analizar señales a diferentes resoluciones en frecuencia, este cambio de resolución se muestra en la figura 3.7 [11].

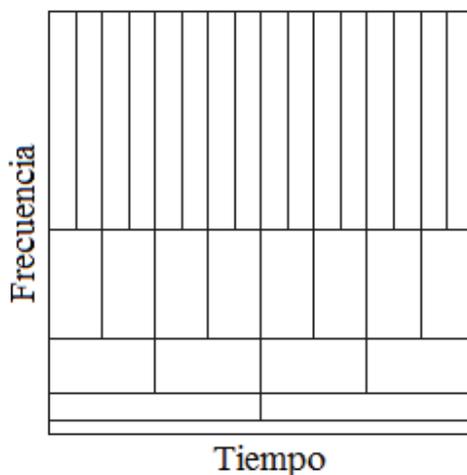


Figura 3.7 Plano multiresolución tiempo-frecuencia [11].

De acuerdo a la figura 3.7, se asume que las frecuencias bajas están presentes durante toda la señal, mientras que las frecuencias altas aparecen de tiempo en tiempo, como un pico corto, lo cual es regularmente el caso en aplicaciones prácticas. El análisis con wavelets calcula la correlación entre la señal en consideración y una función wavelet $\psi(t)$. La similitud entre la señal y la función wavelet de análisis es computada separadamente para diferentes intervalos de tiempo, resultando en una representación bidimensional. La función wavelet de análisis $\psi(t)$ es referida también como la wavelet madre [11].

En comparación con la transformada de Fourier, la función de análisis de la transformada wavelet puede ser escogida con más libertad, sin la necesidad del uso de formas senoidales. Una función wavelet $\psi(t)$ es una onda pequeña, que debe ser oscilatoria, en cierta manera, para discriminar entre diferentes frecuencias. La función wavelet contiene la forma de análisis y la ventana. La figura 3.8 muestra un ejemplo de una posible wavelet conocida como la wavelet Morlet [11].

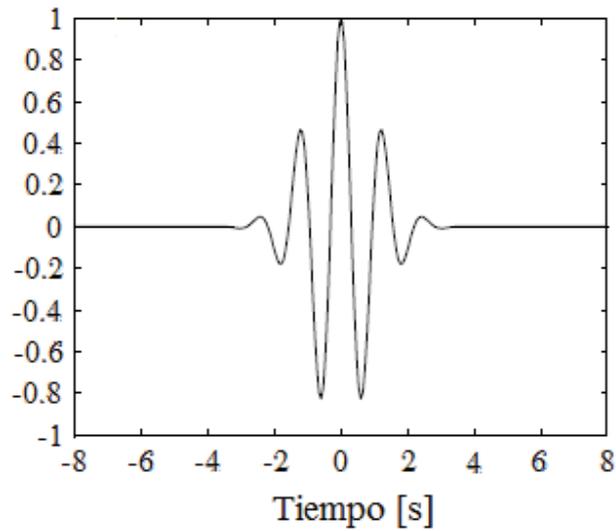


Figura 3.8 Wavelet Morlet [11].

Las wavelets y los bancos de filtros son comúnmente usados en el procesamiento de señales muestreadas. A partir de la función wavelet, se genera un conjunto de bases ortonormales mediante traslación y dilatación inicial, permite una representación multiresolución de una señal, descomponiéndola en esta base de escala de tiempo. En el problema de los arreglos, las observaciones de datos involucran procesos temporales y espaciales. En cualquier instante de tiempo el campo de señal (campo de onda acústica, campo de onda electromagnética, etc.) es muestreado por sensores individuales en posiciones de espacio discretas en cada sensor, la señal que llega es muestreada en puntos discretos de tiempo. Existe una similitud esencial entre el proceso de la señal en el dominio espacial y el proceso de la señal en el dominio temporal: muestreo espacial reemplazando muestreo temporal y espectro direccional reemplazando espectro de frecuencia. Es entonces natural extender el concepto de wavelets aplicado al aspecto espacial del procesamiento de señales en arreglos de micrófonos. Muchos métodos de proceso de arreglos de micrófonos actuales son implementados con base a un muestreo espacial fijo y uniforme sin importar la dirección de la fuente de señal [5].

3.4.1 Representación Multiresolución

Suponiendo s como una variable espacial continua. La función de escalamiento $\psi(s)$ y la función wavelet $w(s)$ satisfacen

$$\text{Ecuación de dilatación } \psi(s) = 2 \sum_{k=0}^K H_L[k] \psi(2s-k)$$

$$\text{Ecuación wavelet } w(s) = 2 \sum_{k=0}^K H_H[k] \psi(2s-k)$$

H_L y H_H son los coeficientes paso bajas y paso altas espaciales, respectivamente, y $H_L[0] + \dots + H_L[K] = 1$.

La traslación y dilatación de la función wavelet es descrita:

$$\{w_k^j(s)\} = \{2^{j/2} w(2^j s - k), k, j \text{ enteros}\}$$

y forman un conjunto de bases ortogonales en el dominio espacial L^2 . El nivel de escala está dado por j y el intervalo de muestreo espacial normalizado en ese nivel es 2^{-j} . Entonces, una función espacial, $x(s)$, puede ser analizada y sintetizada mediante la descomposición y reconstrucción wavelet

$$\text{Síntesis de una función } x(s) = \sum_{j,k} b_k^j w_k^j(s)$$

$$\text{Análisis de función } b_k^j = \int x(s) w_k^j(s) ds$$

donde $b_k^j = \int x(s) w_k^j(s) ds$ es el conjunto de coeficientes de la descomposición wavelet en el nivel j . Un banco de filtros paso bajas y paso altas puede ser usado para implementar el procedimiento anterior: filtros espaciales antialias y submuestreo para el análisis y sobremuestreo y filtros espaciales para la síntesis.

La transformada wavelet puede ser realizada eficientemente con el banco de filtros mostrado en la figura 3.9.

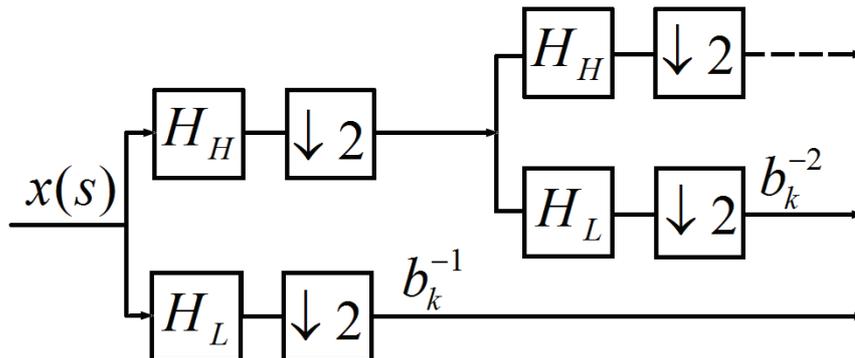


Figura 3.9 Banco de filtros para descomposición Wavelet [5].

3.4.2 Aproximación Wavelet

Las señales de un arreglo de micrófonos pueden ser descompuestas aplicando la transformación wavelet y cada subbanda muestrea el campo de señal recibido dentro de los intervalos angulares correspondientes sin ambigüedad. Entonces es natural implementar un formador de haz en cada sub banda de la descomposición wavelet. La figura 3.10 muestra un formador de haz modificado operando en la primera sub banda [5].

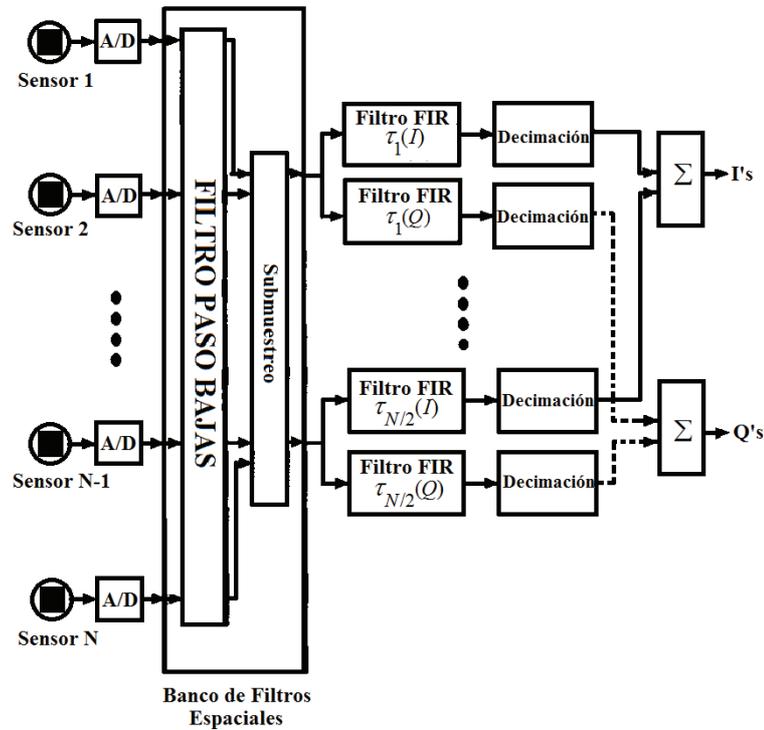


Figura 3.10 Diagrama de bloques de un formador de haz modificado con descomposición Wavelet [5].

La eficiencia computacional hace al esquema wavelet atractivo, usando una representación multiresolución, un esfuerzo computacional extra es necesario para el proceso de wavelets, pero es linealmente proporcional al número de sensores. En contraste, debido a la dimensión reducida de los datos de la sub banda, la carga computacional para la operación del formador de haz se reduce significativamente haciendo el proceso total más eficiente [5].

3.5 Resumen

En este capítulo se mostraron algunas técnicas y métodos de formador de haz que se están estudiando actualmente.

Dentro de las técnicas de formador de haz se encuentran la del retardo y suma, en la cual las señales provenientes de los sensores se retrasan y suman para obtener la respuesta del arreglo.

El formador de haz de máxima varianza da distintos pesos a cada una de las señales del arreglo para obtener una salida de mínima varianza. Otra forma de retrasar las señales en el formador de haz es hacer un corrimiento de fase en las señales. Finalmente se puede hacer una descomposición por medio de wavelets de las señales y a cada descomposición se le aplica alguna técnica de formador de haz.

Capítulo 4

Diseño e Implementación del Sistema



En este capítulo se describe el diseño del sistema, desde su esquema fundamental, sus elementos físicos, el arreglo de micrófonos, el amplificador y la tarjeta del DSP, hasta los algoritmos y parámetros necesarios para su implementación.

Se muestra el cálculo de los parámetros físicos del arreglo de micrófonos, los criterios de selección para el arreglo óptimo entre varias opciones posibles y las simulaciones basadas en el arreglo seleccionado; así como diferentes métodos de formador de haz para obtener el más adecuado para la aplicación.

Asimismo, se muestra la implementación del sistema en la tarjeta de DSP seleccionada y su integración con los elementos del sistema que se construyeron.

4.1 Diseño General del Sistema

En la implementación de un sistema para filtrar espacialmente señales acústicas en tiempo real, primero se deben definir las partes y etapas que lo conformarán; la figura 4.1 muestra el esquema básico del sistema a implementar.

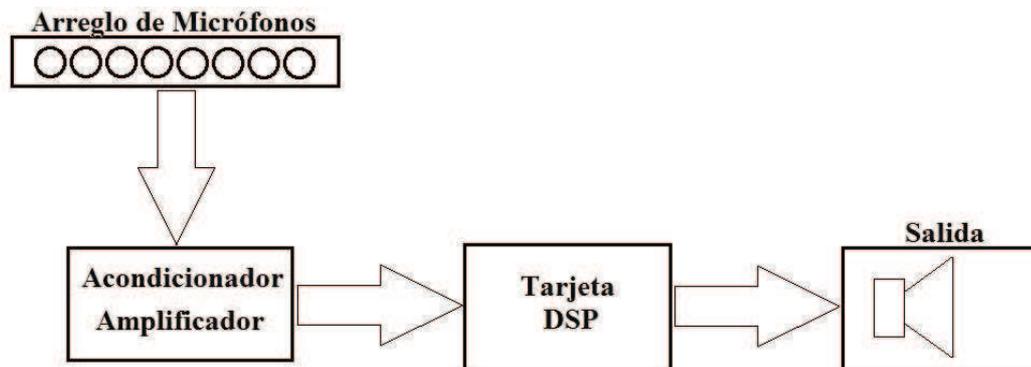


Figura 4.1 Esquema básico del sistema.

El arreglo de micrófonos que se observa en la figura 4.1 debe captar las señales acústicas para procesarlas y tener características específicas para el tipo de señales que capta, tales como el número de elementos y la distancia entre ellos, las cuales se presentan mas adelante.

El bloque amplificador y acondicionador recibe las señales captadas por los micrófonos y las filtra, amplifica y les suma un voltaje de offset para que puedan ser recibidas por la tarjeta DSP, la cual hará todo el procesamiento de estas señales mediante algoritmos específicos diseñados para este sistema y su propósito de filtrar espacialmente las señales en los micrófonos.

4.2 Diseño del Arreglo de Micrófonos

Para implementar el sistema de filtrado espacial se requiere tener un arreglo de micrófonos diseñado especialmente para el tipo de señales que se estarán utilizando; esto es, señales de banda ancha en el intervalo de frecuencias de la voz, por lo que se manejarán estas señales por medio de sub bandas de frecuencia dentro del intervalo de frecuencias total de las señales.

Se proponen cuatro bandas espaciadas en octavas, es decir, las frecuencias que delimitan a cada banda son el doble de las frecuencias que delimitan a la banda anterior, abarcando el intervalo de frecuencias completo de la voz; para este trabajo se considerará un intervalo de frecuencias de 200 Hz a 4800 Hz. Las bandas se definen de la siguiente manera:

- Banda 1. 200 Hz – 600 Hz $BW = 400$ Hz
- Banda 2. 400 Hz – 1200 Hz $BW = 800$ Hz
- Banda 3. 800 Hz – 2400 Hz $BW = 1600$ Hz
- Banda 4. 1600 Hz – 4800 Hz $BW = 3200$ Hz

donde $BW =$ ancho de banda

Como se puede observar, existe cierto traslape entre las bandas, lo cual es deseable para tener cubiertas todas las frecuencias es su totalidad y no dejar zonas de frecuencia sin cubrir en los límites de las bandas.

Con base en esta distribución de bandas se debe hacer el cálculo de cuántos micrófonos necesita cada banda para procesar las señales adecuadamente teniendo en cuenta las siguientes consideraciones:

- El espacio dentro del laboratorio donde se harán las pruebas limita la distancia de la fuente de sonido al arreglo de micrófonos a 4 metros.
- Las señales se consideran que llegan al arreglo de micrófonos en el campo lejano, es decir, se consideran ondas planas.
- La frecuencia de muestreo para cada banda debe cumplir con el teorema de Nyquist.
- Como se observa en la figura 2.9(c), el valor máximo de la relación entre la distancia entre micrófonos y la longitud de onda debe ser $d \leq \lambda_{\max}/2$ para evitar la aparición de lóbulos gratinados.
- La velocidad de propagación de una onda acústica en el aire es $v = 340 \text{ m/seg}$ a temperatura ambiente.

Procedimiento de diseño

A continuación se muestran los pasos que se siguieron para el diseño del arreglo de micrófonos.

Primero. Se calculó el número de sensores que requeriría un arreglo para cada banda definida, y después se construyó un solo arreglo que combina los cuatro arreglos resultantes del cálculo.

Segundo. Se determinó la distancia entre micrófonos para cada uno de los arreglos que corresponden a las cuatro bandas que tenemos,

$$d_l = 0.5 \times \lambda_{l\max} = \frac{\lambda_{l\max}}{2} \quad (4.1)$$

donde $l=1,2,3,4$ indica el número de la banda.

La longitud de onda de las frecuencias máximas se calcula como

$$\lambda_{l\max} = \frac{v}{f_{l\max}} \quad (4.2)$$

sustituyendo (4.2) en (4.1)

$$d_l = \frac{v}{2 f_{l\max}} \quad (4.3)$$

por lo que las distancias entre micrófonos por banda quedan

$$d_1 = 0.28[m]$$

$$d_2 = 0.14[m]$$

$$d_3 = 0.07[m]$$

$$d_4 = 0.035[m]$$

Tercero. Suponiendo ondas en el campo lejano y que llegan con un frente de onda plano a los micrófonos, se utiliza la expresión para la distancia de campo lejano dada por [12]

$$r > \frac{d_{ltotal}^2 f_{sbl}}{v} \quad (4.4)$$

donde d_{ltotal}^2 es la longitud total del arreglo de micrófonos de la banda, f_{sbl} es la frecuencia de muestreo de cada banda correspondiente y r es la distancia de la fuente de sonido al centro del arreglo de micrófonos que en este caso es 4 m . La longitud total de cada arreglo está dada por

$$d_{ltotal} = (n_l - 1) d_l \quad (4.5)$$

donde n_l es el número de micrófonos del arreglo de cada banda.

Cuarto. Debemos determinar las frecuencias de muestreo de cada banda de acuerdo al teorema de Nyquist, en este caso la frecuencia de muestreo debe ser al menos el doble de la frecuencia más alta de cada banda; sin embargo, esta frecuencia no es necesariamente la más óptima para la implementación del sistema. Por lo anterior se evaluaron valores distintos de frecuencia de muestreo para cada banda, a fin de encontrar el valor óptimo, y como los anchos de banda están espaciados en octavas, las frecuencias de muestreo también deben estar dadas en términos de octavas.

A partir de (4.4) y (4.5) se puede determinar el número de micrófonos por cada banda como

$$n_l = \frac{\sqrt{rv/f_{sl}}}{d_l} + 1$$

Se calculó el número de micrófonos para cada banda, usando diferentes valores de frecuencia de muestreo, y los resultados se presentan en la figura 4.2, en la que se observa el número de micrófonos necesarios para cada banda a diferentes frecuencias de muestreo espaciadas en octavas.

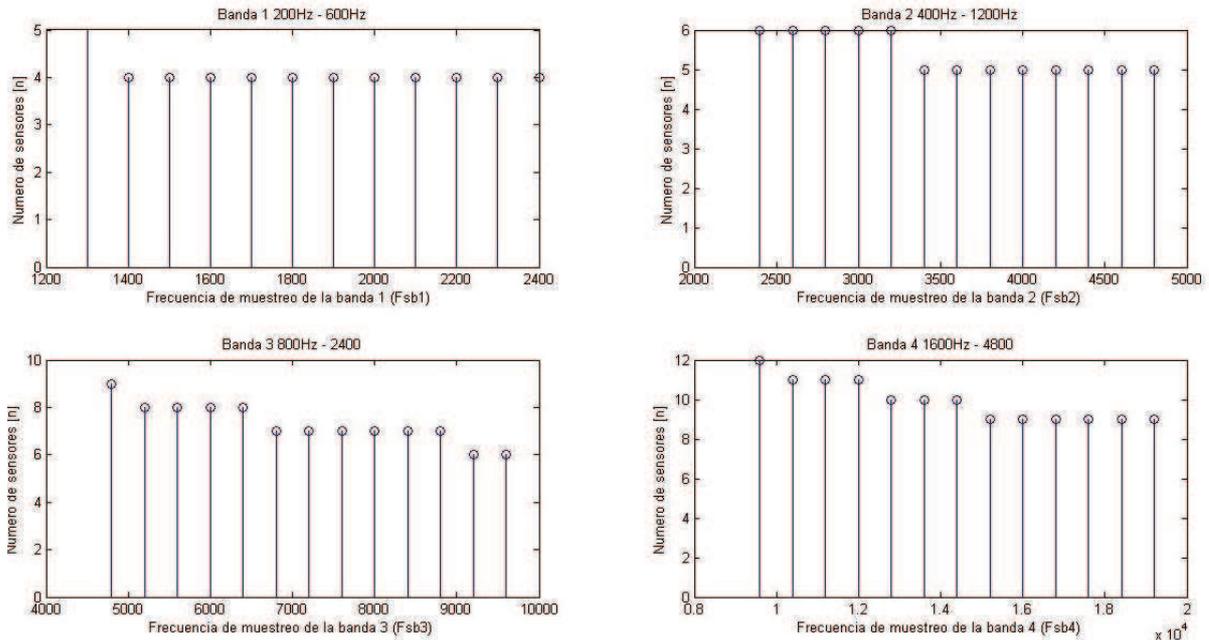


Figura 4.2 Número de micrófonos para las cuatro bandas a diferentes frecuencias.

Quinto. Una vez calculado el número óptimo de micrófonos y la distancia entre ellos, para cada banda a diferentes frecuencias de muestreo, se deben escoger las cuatro frecuencias de muestreo más convenientes para el sistema, manteniendo siempre la relación de octavas entre ellas. Para esto se diseñó un programa que toma las frecuencias de muestreo por octavas, el número de micrófonos por banda y la distancia entre ellos y genera un solo arreglo simétrico por conjunto de frecuencias de muestreo. En total, el programa genera 13 arreglos, de los cuales algunos son iguales entre sí pero están definidos para diferentes conjuntos de frecuencias. El primer criterio de discriminación de los arreglos es el número total de micrófonos que contiene el mismo, el cual está limitado por el número de canales de conversión A/D que tiene el DSP en el que se implementará el sistema, que en este caso es de 16 canales.

Sexto. Una vez que se eliminan los arreglos que exceden el número de micrófonos permitidos, quedan sólo dos arreglos óptimos diferentes para escoger, cada uno de 15 elementos, de los cuales se escoge el que tiene la menor frecuencia de muestreo para la banda 4, ya que es la frecuencia mas alta a la que operará el sistema y por lo tanto es la que limitará el número de operaciones disponibles entre muestra y muestra que reciba el sistema; por lo tanto, una frecuencia menor permite tener un número mayor de operaciones disponibles, por lo que se seleccionó el arreglo de la figura 4.3.

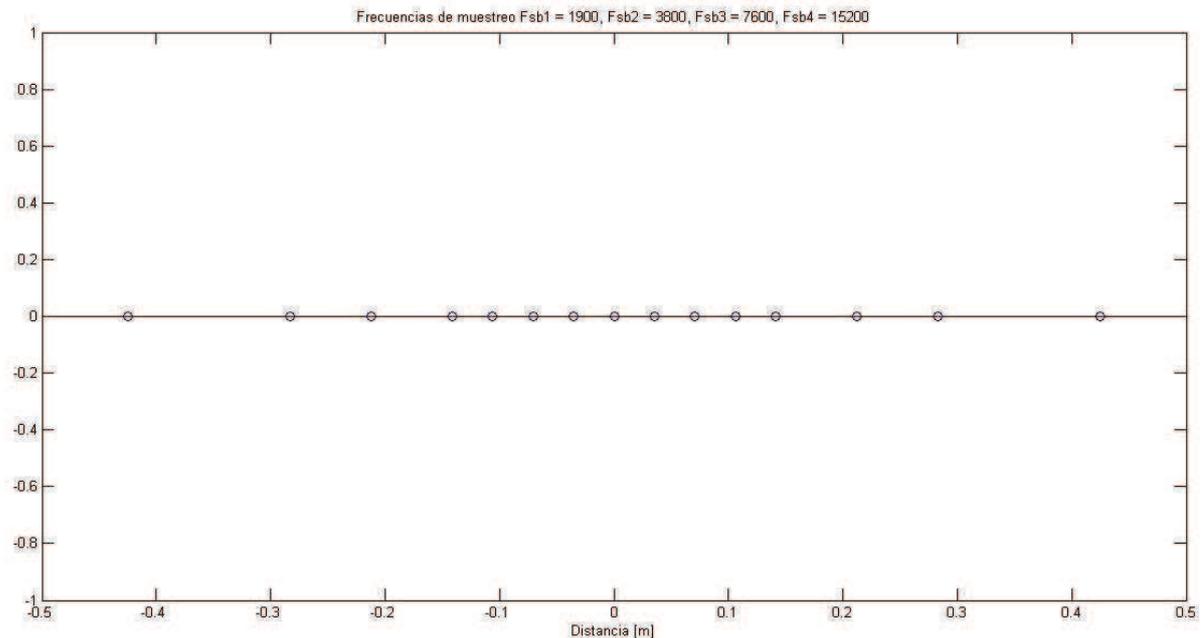


Figura 4.3 Arreglo seleccionado con 15 micrófonos.

Como se observa en la figura 4.3 las frecuencias óptimas de muestreo por banda para el arreglo son

$$f_{sb1} = 1900 [Hz]$$

$$f_{sb2} = 3800 [Hz]$$

$$f_{sb3} = 7600 [Hz]$$

$$f_{sb4} = 15200 [Hz]$$

y el número de micrófonos por banda es

$$n_1 = 4$$

$$n_2 = 5$$

$$n_3 = 7$$

$$n_4 = 9$$

Se diseñó un solo arreglo que contiene los cuatro arreglos necesarios para las diferentes bandas, es decir, los arreglos están combinados y los arreglos individuales se formarán mediante el programa a implementar, el cual tomará las señales de los micrófonos que correspondan al arreglo individual de cada banda. La figura 4.4 muestra qué micrófonos se utilizarán para cada banda.

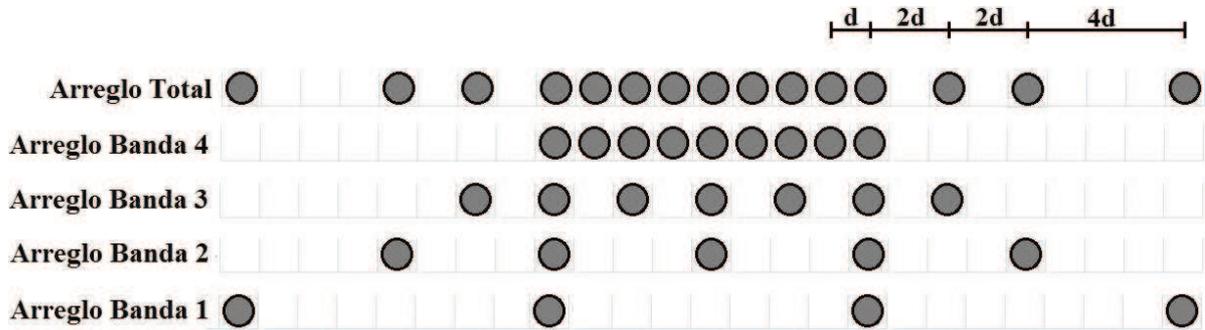


Figura 4.4 Distribución de micrófonos en el arreglo total y por banda.

donde d se calculó con la ecuación (4.3) y $d = 0.035[m]$.

4.3 Determinación del Método a Usar

Una vez determinadas las características del arreglo, distancias y número de elementos, se hicieron simulaciones de métodos de formador de haz teniendo como parámetros las características del arreglo mencionado, las bandas y frecuencias definidas anteriormente. Se simuló el sistema completo desde las señales recibidas en los micrófonos hasta los algoritmos de formador de haz que se implementarán en el DSP (figura 4.1).

El sistema divide el análisis de las señales en cuatro bandas de frecuencias definidas en la sección anterior. El sistema debe separar las señales de acuerdo a su contenido frecuencial y se implementaron cuatro filtro paso banda de tipo FIR con las siguientes características:

- Filtro 1. $N = 50$ $f_{C1} = 200Hz$ $f_{C2} = 600Hz$
- Filtro 2. $N = 50$ $f_{C1} = 400Hz$ $f_{C2} = 1200Hz$
- Filtro 3. $N = 50$ $f_{C1} = 800Hz$ $f_{C2} = 2400Hz$
- Filtro 4. $N = 50$ $f_{C1} = 1600Hz$ $f_{C2} = 4800Hz$

Se generaron señales desde 200 Hz hasta 4800 Hz con un incremento de 100 Hz y posiciones angulares desde -90° a 90° con incrementos de 0.2° , las cuales se dividen en bandas de frecuencia por medio de la aplicación de los filtros descritos anteriormente. Una vez que las señales se separan en cuatro bandas se implementa un algoritmo formador de haz para cada banda, primero usando un formador de haz fijo para realizar las simulaciones, usando ventanas espaciales de Hamming como los pesos para las señales de cada micrófono, teniendo ventanas de distinta longitud para cada banda, estas ventanas espaciales se muestran en la figura 4.5.

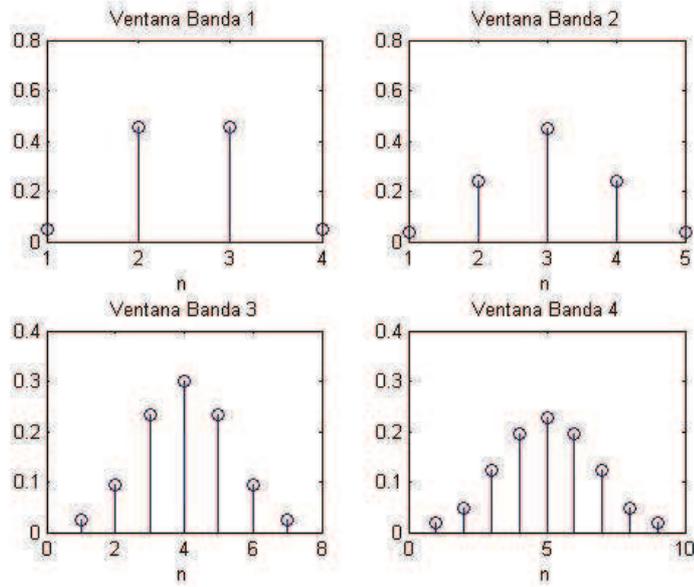


Figura 4.5 Ventanas.

Se generó también una matriz de dirección conformada por los retrasos en muestras que deben hacerse en las señales de entrada para tener el haz orientado en esa dirección. Los retrasos por cada banda se calculan como

$$\delta_{ln} = \frac{f_{sl} d_l n \text{sen}(\theta)}{v}$$

donde θ es el ángulo al que se quiere dirigir el haz, n es el número del micrófono donde se calcula el retraso, d_l es la distancia entre micrófonos de cada banda, f_{sl} es la frecuencia de muestreo de cada banda y v es la velocidad de propagación de las ondas acústicas.

Considerando las bandas divididas en octavas, se tienen las siguientes relaciones:

$$d_1 = 2d_2 \tag{4.6a}$$

$$d_2 = 2d_3 \tag{4.6b}$$

$$d_3 = 2d_4 \tag{4.6c}$$

$$f_{s2} = 2 f_{s1} \tag{4.7a}$$

$$f_{s3} = 2 f_{s2} \tag{4.7b}$$

$$f_{s4} = 2 f_{s3} \tag{4.7c}$$

Los retrasos en la banda 1 se calculan como

$$\delta_{1n} = \frac{f_{s1} d_1 n \text{sen}(\theta)}{v}$$

Los retrasos en la banda 2 se calculan como

$$\delta_{2n} = \frac{f_{s2} d_2 n \text{sen}(\theta)}{v} \quad (4.8)$$

Sustituyendo (4.6a) y (4.7a) en (4.8) tenemos

$$\delta_{2n} = \frac{f_{s2} d_2 n \text{sen}(\theta)}{v} = 2 f_{s1} \frac{d_1 n \text{sen}(\theta)}{2v} = \frac{f_{s1} d_1 n \text{sen}(\theta)}{v}$$

entonces:

$$\delta_{2n} = \delta_{1n}$$

Si se hace lo mismo con las demás relaciones llegamos a la relación de retardos

$$\delta_{1n} = \delta_{2n} = \delta_{3n} = \delta_{4n}$$

Construyendo una matriz de dirección (ecuación (4.9)) para la banda 4 puede ser utilizada para las otras bandas, se escoge la matriz de la banda 4 ya que al ser la banda con mayor número de micrófonos su matriz de dirección contiene a las bandas restantes. Los cálculos de los retrasos nos dan como resultados números reales, sin embargo necesitamos números enteros que representen los retrasos como muestras de la señal, ya que se están manejando señales digitales por lo que los retrasos se deben redondear a solo números enteros. En la matriz de dirección las columnas representan los ángulos a los que se quiere dirigir el formador de haz y los renglones representan los retardos por micrófono. La matriz de dirección queda como la ecuación (4.9), donde la variación de los renglones representa una variación en el formador de haz, y las variaciones en las columnas los retardos.

		Retardo								
Ángulo	-90	13	11	9	8	6	5	3	2	0
	-85	12	11	9	8	6	5	3	2	0
	-80	12	11	9	8	6	5	3	2	0
	-75	12	11	9	8	6	5	3	2	0
	-70	12	10	9	7	6	4	3	1	0
	-65	11	10	9	7	6	4	3	1	0
	-60	11	9	8	7	5	4	3	1	0
	-55	10	9	8	6	5	4	3	1	0
	-50	10	8	7	6	5	4	2	1	0
	-45	9	8	7	6	4	3	2	1	0
	-40	8	7	6	5	4	3	2	1	0
	-35	7	6	5	4	4	3	2	1	0
	-30	6	5	5	4	3	2	2	1	0
	-25	5	5	4	3	3	2	1	1	0
	-20	4	4	3	3	2	2	1	1	0
	-15	3	3	2	2	2	1	1	0	0
	-10	2	2	2	1	1	1	1	0	0
	-5	1	1	1	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	1	1	1	1	1
	10	0	0	1	1	1	1	2	2	2
	15	0	0	1	1	2	2	2	3	3
	20	0	1	1	2	2	3	3	4	4
	25	0	1	1	2	3	3	4	5	5
	30	0	1	2	2	3	4	5	5	6
	35	0	1	2	3	4	4	5	6	7
	40	0	1	2	3	4	5	6	7	8
45	0	1	2	3	4	6	7	8	9	
50	0	1	2	4	5	6	7	8	10	
55	0	1	3	4	5	6	8	9	10	
60	0	1	3	4	5	7	8	9	11	
65	0	1	3	4	6	7	9	10	11	
70	0	1	3	4	6	7	9	10	12	
75	0	2	3	5	6	8	9	11	12	
80	0	2	3	5	6	8	9	11	12	
85	0	2	3	5	6	8	9	11	12	
90	0	2	3	5	6	8	9	11	13	

(4.9)

Matriz de dirección.

Una gráfica de la matriz de dirección en tres dimensiones se muestra figura 4.6

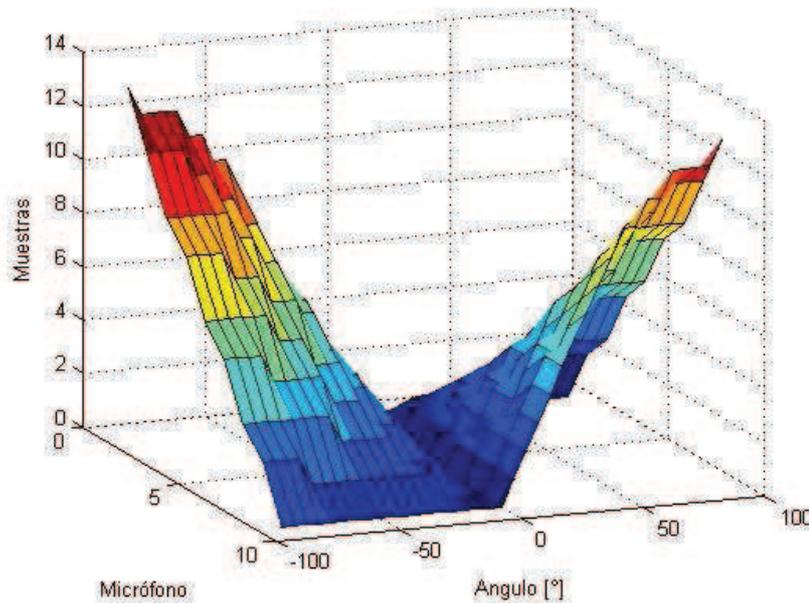


Figura 4.6 Matriz de dirección.

De la figura 4.6 se puede apreciar la simetría de la gráfica de la matriz de dirección, lo cual se puede asociar a la forma del patrón direccional que se tendrá, por lo que se puede asumir que dicho patrón también será simétrico.

Teniendo la matriz de dirección, se puede direccionar el formador de haz hacia un cierto ángulo en específico, del cual proviene la señal que se quiere captar y omitir las señales provenientes de los ángulos restantes.

Las señales del programa de simulación están generadas a una frecuencia de muestreo igual a la mayor frecuencia de muestreo determinada para las bandas, es decir 15200 Hz. Además sabiendo que cada banda tiene una frecuencia de muestreo diferente, se debe hacer un ajuste en las señales para que cumplan con esta característica, ya que de ella depende el que se pueda considerar que la señal está en el campo lejano. Las relaciones de (4.7) se pueden describir como (4.10a - 4.10c).

$$f_{s4} = 8 f_{s1} \quad (4.10a)$$

$$f_{s4} = 4 f_{s2} \quad (4.10b)$$

$$f_{s4} = 2 f_{s3} \quad (4.10c)$$

lo cual es evidente al saber que las bandas fueron establecidas como octavas y al ser la frecuencia de muestreo mas alta múltiplo de las otras frecuencias de muestreo, se puede obtener las frecuencias de muestro de cada banda a partir de la frecuencia de muestreo mas alta que será la frecuencia de muestreo del sistema. Submuestreando las señales a una tasa determinada por la

relación entre las frecuencias de muestreo dada por (4.11), se obtienen las frecuencias de muestreo de las bandas con menos micrófonos.

$$R_{sn} = \frac{f_{sn}}{f_{s4}} \quad (4.11)$$

Calculando esta tasa para cada banda se tiene que

$$R_{s1} = \frac{f_{s1}}{f_{s4}} = \frac{1}{8} \quad (4.12a)$$

$$R_{s2} = \frac{f_{s2}}{f_{s4}} = \frac{1}{4} \quad (4.12b)$$

$$R_{s3} = \frac{f_{s3}}{f_{s4}} = \frac{1}{2} \quad (4.12c)$$

$$R_{s4} = \frac{f_{s4}}{f_{s4}} = 1 \quad (4.12d)$$

Una vez que se han submuestreado las señales con la tasa correspondiente se aplica el formador de haz fijo y las salidas que se obtienen se deben volver a ajustar al número de muestras original, dado que al submuestrear el tamaño de las señales disminuyó con la misma tasa calculada en (4.12), para realizar el sobremuestreo se hace un llenado con ceros y después se filtran las señales con el mismo filtro paso banda que se utilizó para separarlas en bandas con el fin de recuperar el tamaño de la ventana original de las señales. Una vez recuperado el tamaño de ventana original se calcula la potencia de la señales de salida en cada posición angular y en cada frecuencia indicada en los barridos determinados para obtener así el patrón de comportamiento del sistema.

La figura 4.7 muestra un diagrama a bloques general del sistema.

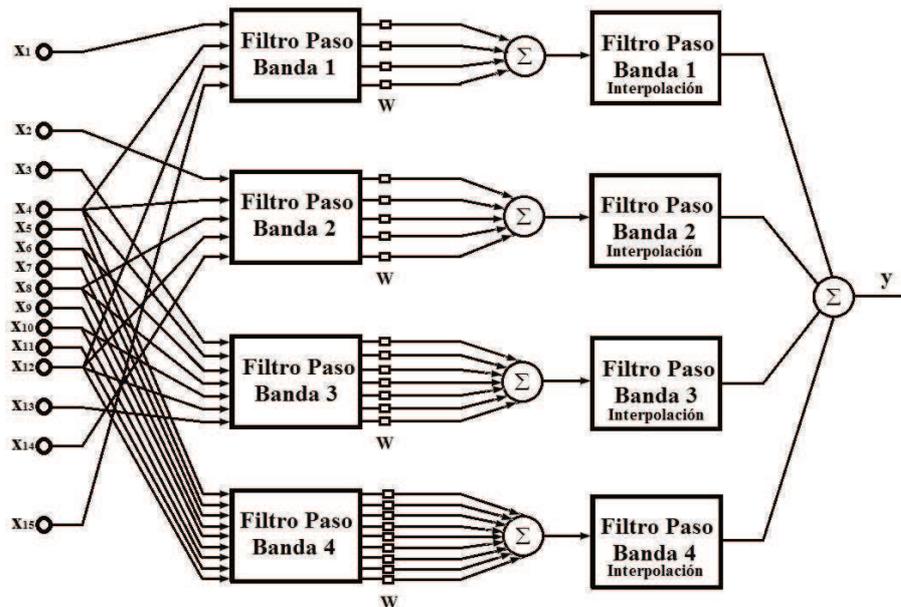


Figura 4.7 Diagrama a bloques del sistema formador de haz fijo multibanda.

Las figuras 4.8 a 4.14 muestran los resultados de la simulación, para un ángulo de 0° y la aplicación de un formador de haz fijo, para las cuatro bandas utilizadas.

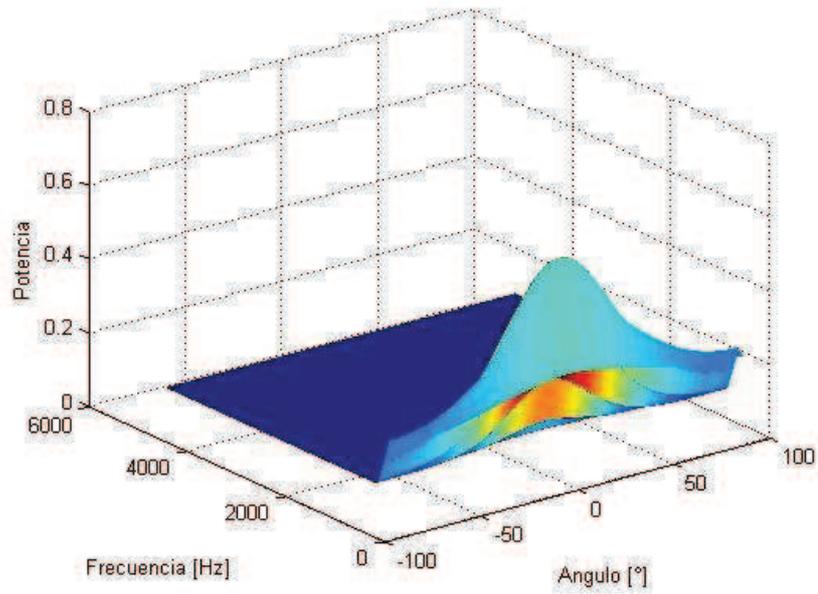


Figura 4.8 Banda 1, ángulo 0° y formador de haz fijo.

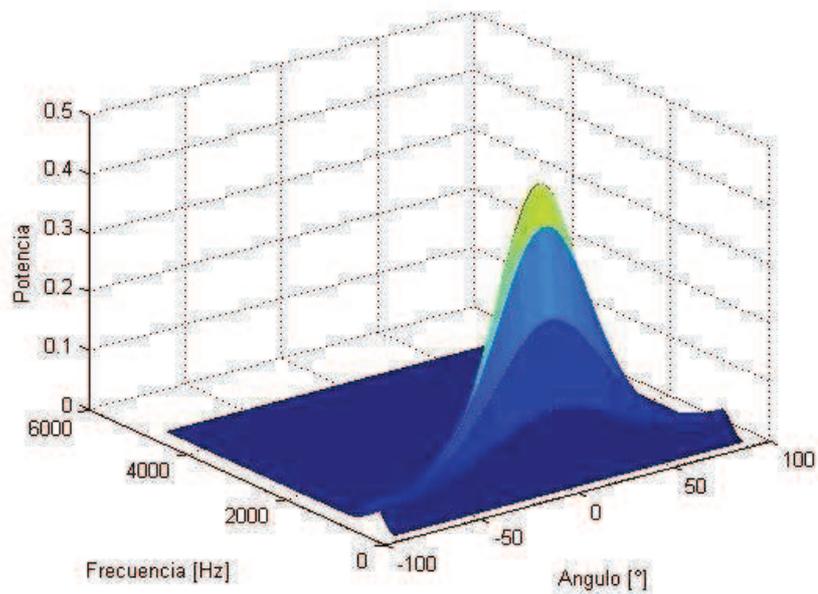


Figura 4.9 Banda 2, ángulo 0° y formador de haz fijo.

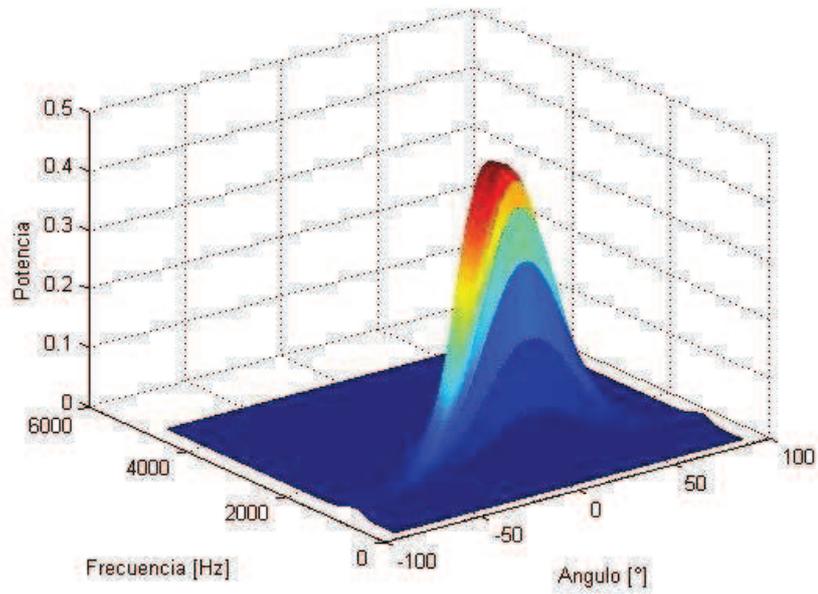


Figura 4.10 Banda 3, ángulo 0° y formador de haz fijo.

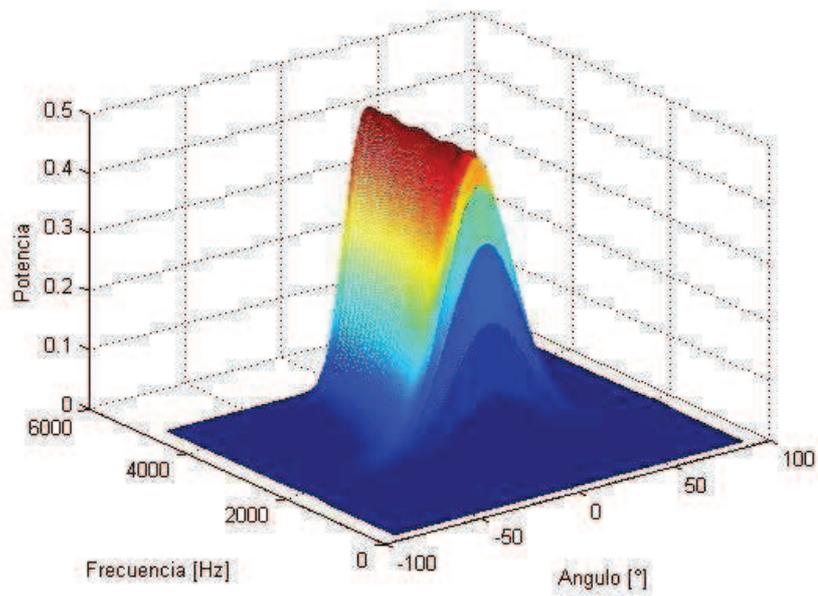


Figura 4.11 Banda 4, ángulo 0° y formador de haz fijo.

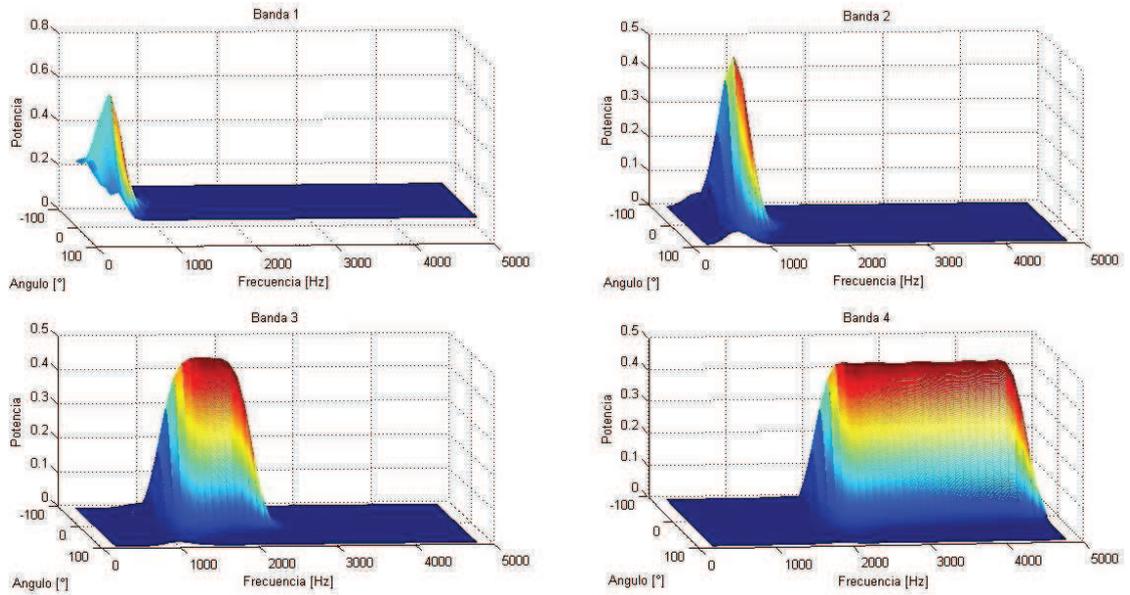


Figura 4.12 Bandas 1 a 4, ángulo 0° y formador de haz fijo.

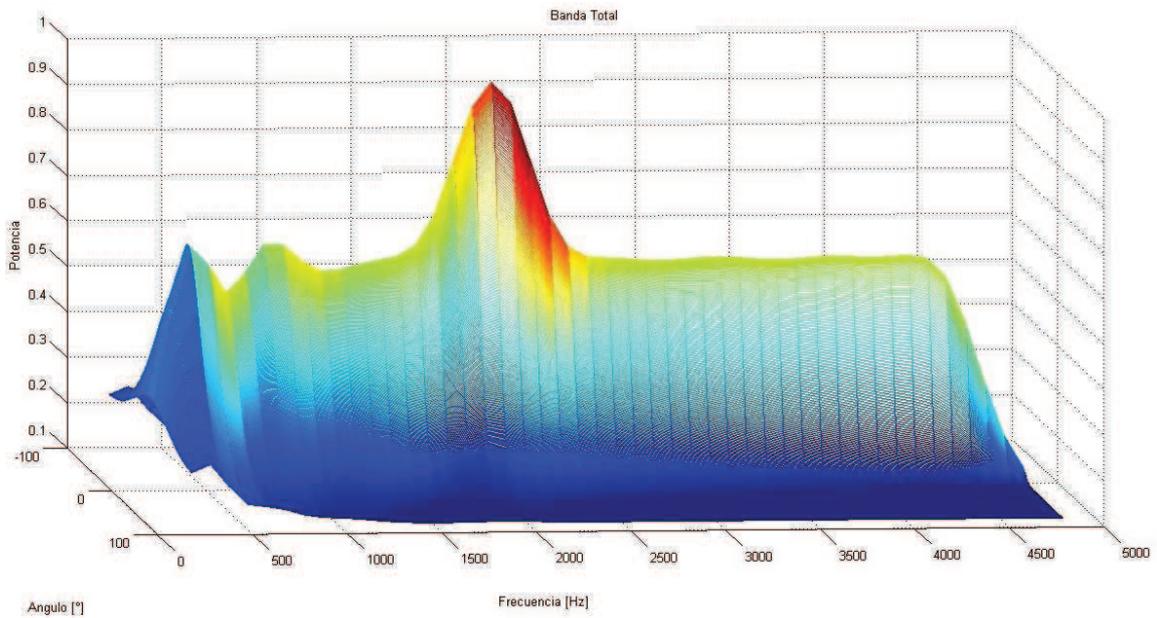


Figura 4.13 Banda total, ángulo 0° y formador de haz fijo.

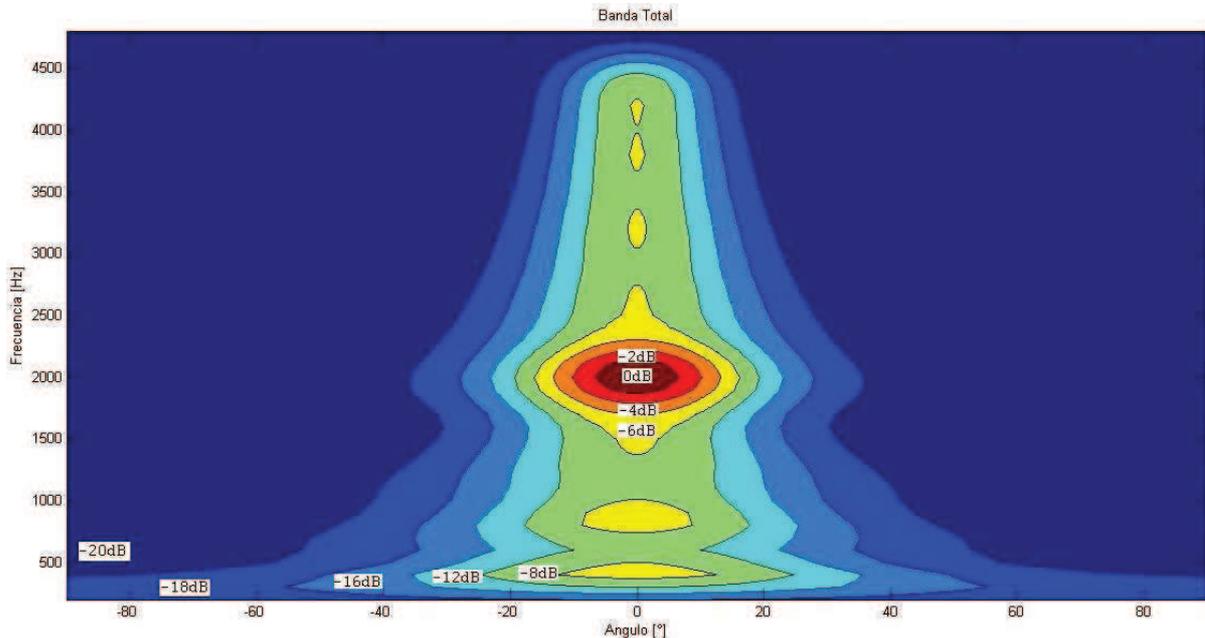


Figura 4.14 Formador de haz fijo multibanda.

En las figuras 4.8 a 4.12 se puede observar que el formador de haz está dirigido hacia el ángulo 0° que se especificó y que separa el intervalo de frecuencias en las cuatro bandas propuestas, sin embargo, a pesar de estar centrado en el ángulo establecido, también deja pasar señales provenientes de ángulos cercanos lo que hace que el grosor del lóbulo principal del patrón direccional sea aproximadamente de unos 50° para una ganancia que se puede considerar significativa (de 70% a 100% de la ganancia máxima). Una vez teniendo la simulación del sistema funcionando correctamente, se realizaron simulaciones para ángulos entre -90° y 90° con un incremento de 5° , observándose comportamientos similares. En la figura 4.13 se muestra el patrón direccional de la banda de frecuencias completa y se observa un pico alrededor de la frecuencia de 2000 Hz debido a que es ahí donde se presenta un empalme de las bandas 3 y 4 y se presenta una potencia mayor, ya que estas frecuencias son procesadas por los formadores de haz de ambas bandas.

La figura 4.14 muestra el contorno del patrón direccional total y se observa que en las frecuencias bajas el lóbulo central es más ancho y se va haciendo más delgado conforme la frecuencia aumenta.

El programa se modificó para utilizar un formador de haz adaptable de tipo GSC, como se muestra en la figura 4.15, en cada banda de frecuencias, y para hacer un barrido angular de -90° a 90° con un incremento de 0.5° . El algoritmo GSC hace una estimación adaptable de señales de interferencia para luego restárselas a la señal deseada que contiene las interferencias, por tanto se debe obtener la señal deseada u óptima al ángulo propuesto, lo cual conducirá a un mejor formador de haz. Este algoritmo fue explicado en el Capítulo 3.

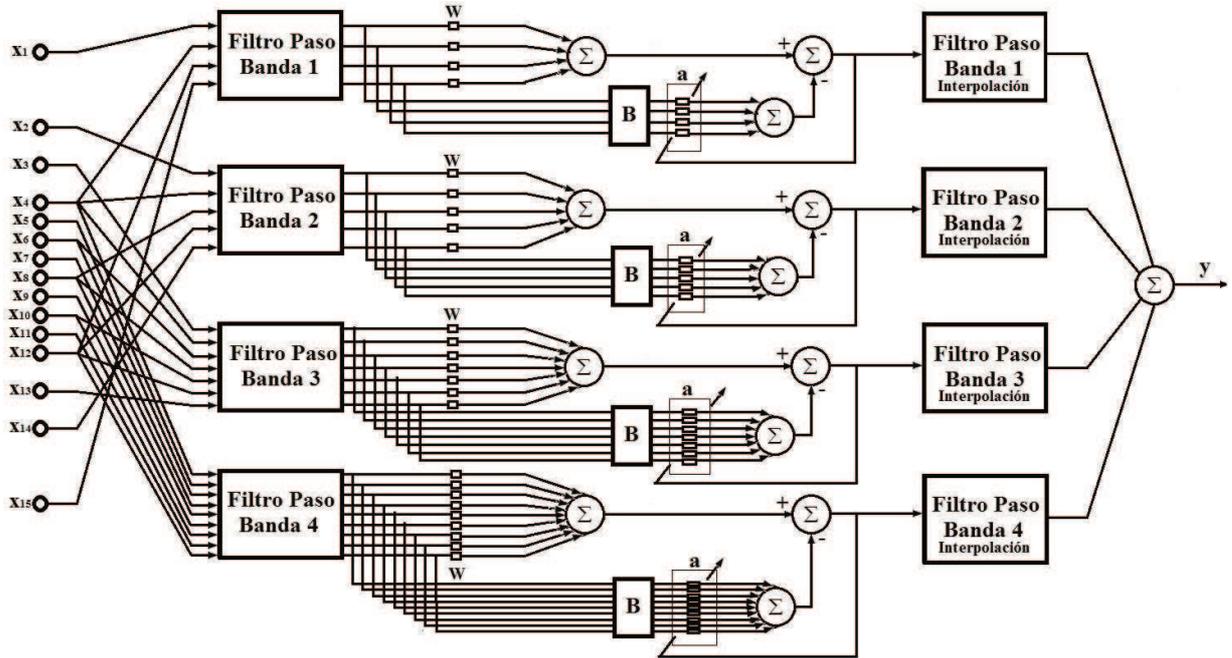


Figura 4.15. Diagrama a bloques del sistema con algoritmo GSC.

Las figuras 4.16 a 4.22 muestran el resultado de la simulación para un ángulo de 0° y la aplicación de un formador de haz con el algoritmo GSC.

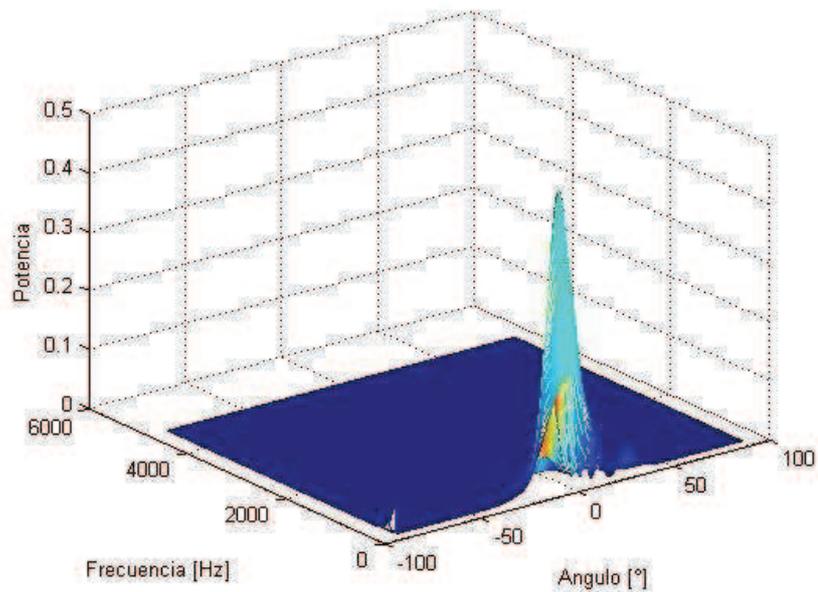


Figura 4.16 Banda 1, ángulo 0° y GSC.

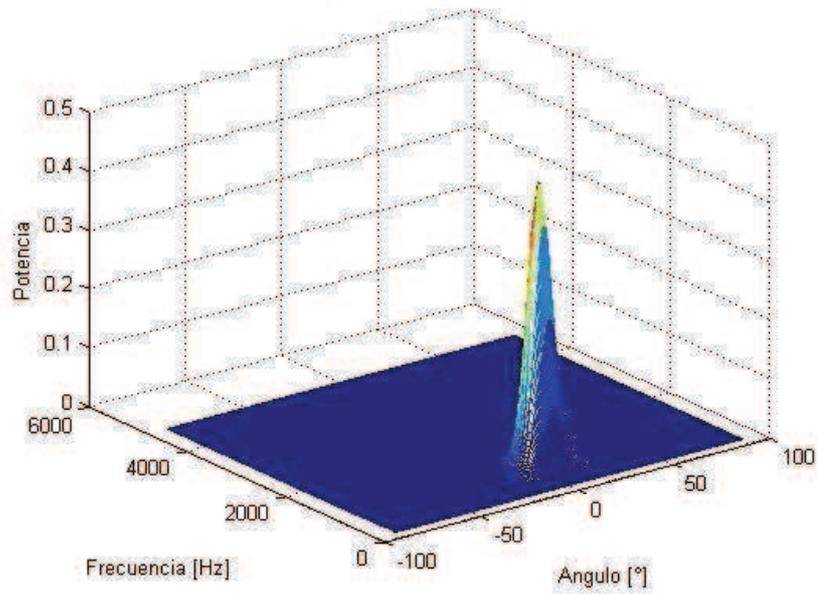


Figura 4.17 Banda 2, ángulo 0° y GSC.

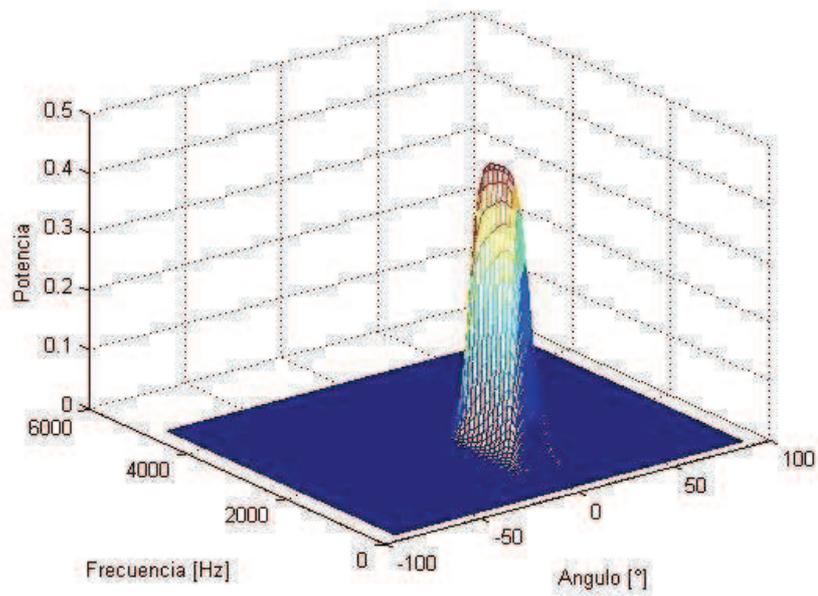


Figura 4.18 Banda 3, ángulo 0° y GSC.

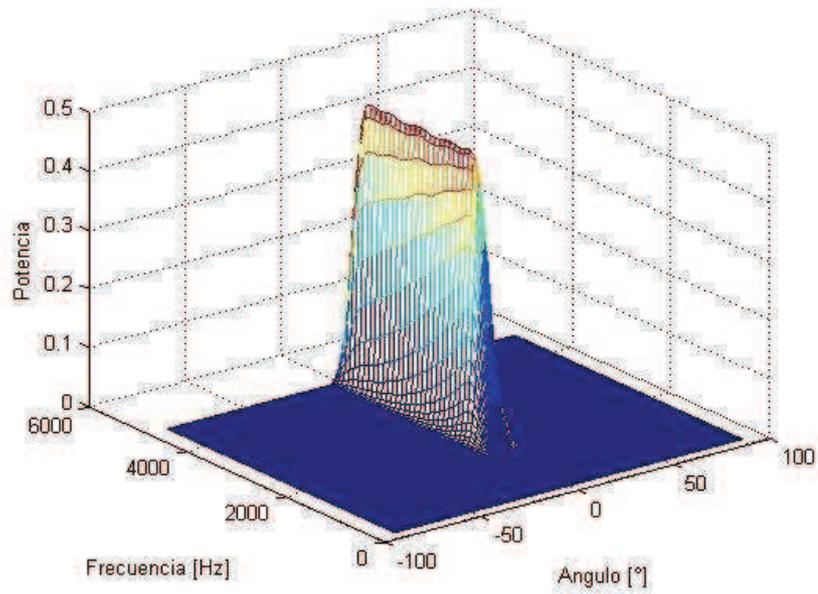


Figura 4.19 Banda 4, ángulo 0° y GSC.

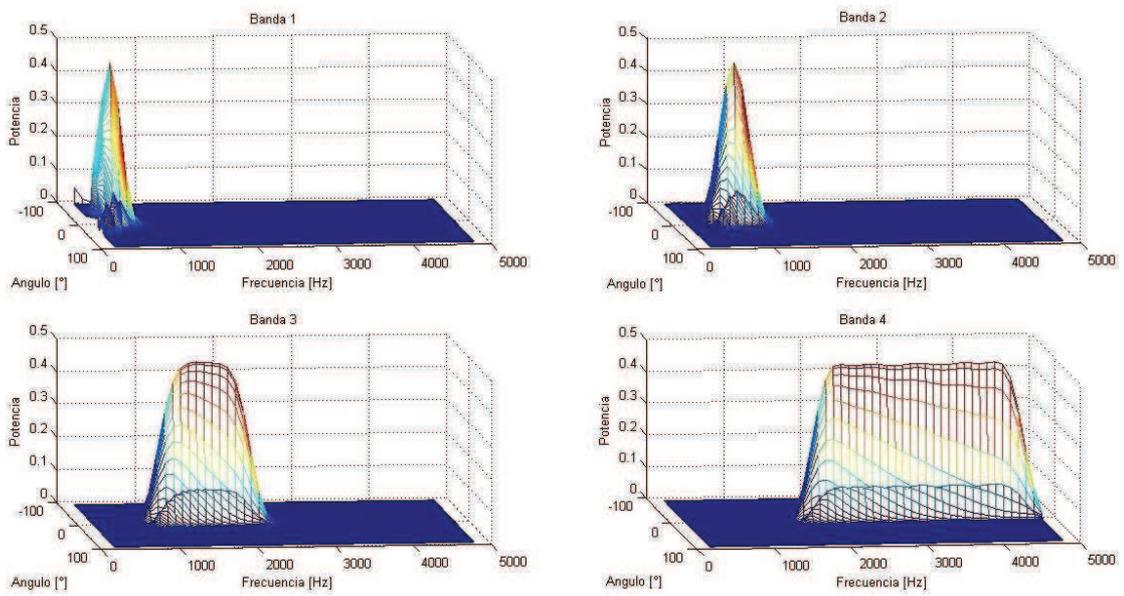


Figura 4.20 Bandas 1 a 4, ángulo 0° y GSC.

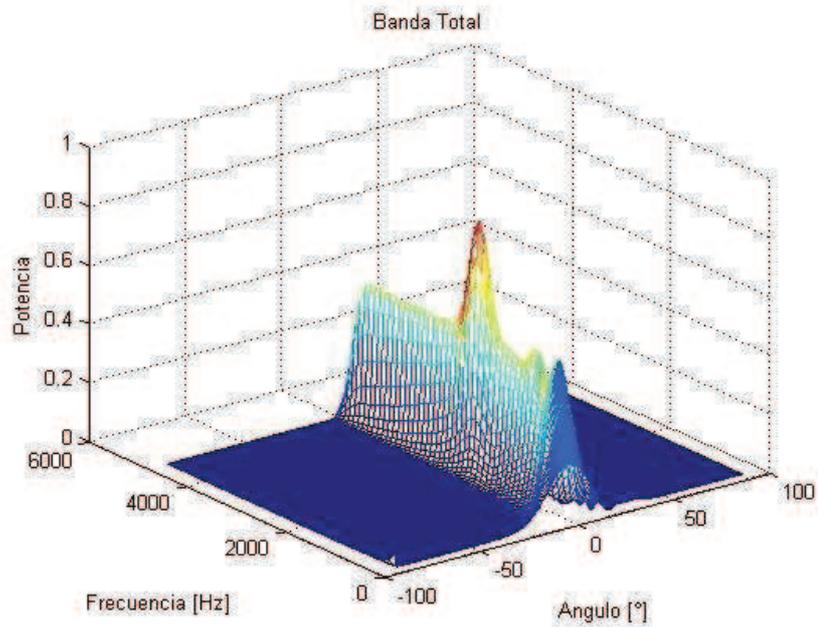


Figura 4.21 Banda total, ángulo 0° y GSC.

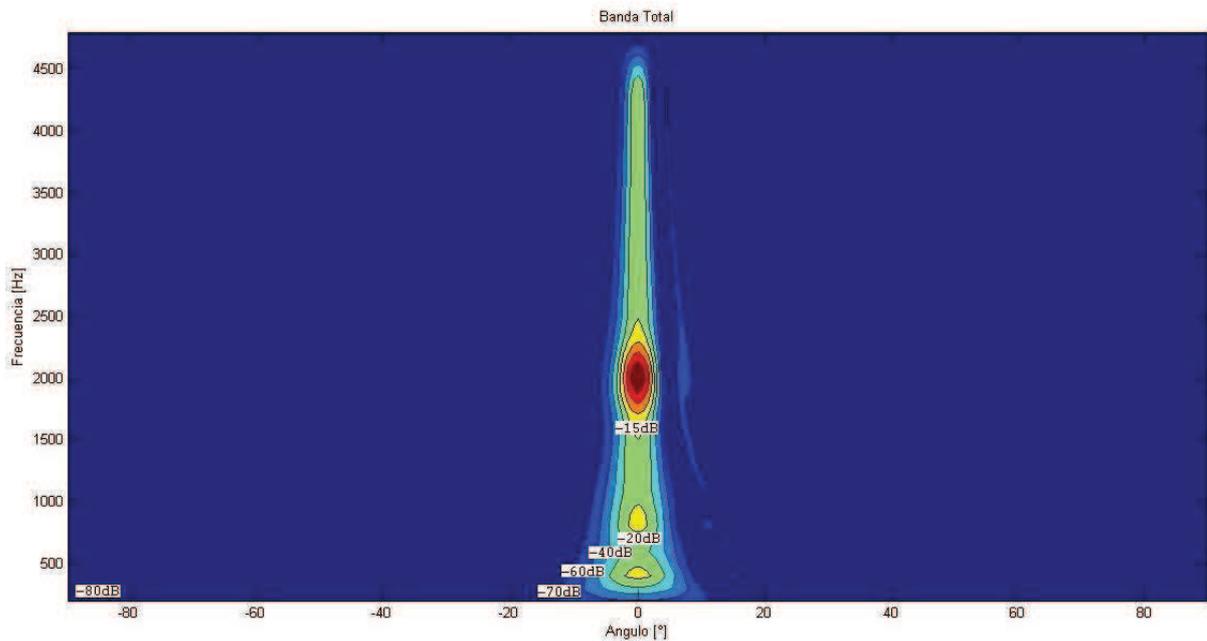


Figura 4.22 Banda total contorno con GSC.

En las figuras 4.16 a 4.20 se observa la mejora en el resultado de la simulación en comparación al formador de haz fijo, el sistema se centra en el ángulo determinado y separa el intervalo de frecuencia en las cuatro bandas; sin embargo, el grosor del lóbulo principal disminuye considerablemente, a unos 10° o menos, para ganancias arriba del 70% de la máxima, y su altura respecto a los demás puntos también aumenta, esto indica que solo está dejando pasar las señales

que provienen del ángulo determinado y las que provienen de ángulos muy cercanos. Esto es la base del filtrado espacial de señales acústicas. En la figura 4.21 se muestra el patrón direccional de la banda de frecuencias completa y se observa un pico alrededor de la frecuencia de 2000 Hz debido a que es ahí donde se presenta un traslape de las bandas 3 y 4 y se presenta un a potencia mayor ya que estas frecuencias son procesadas por los formadores de haz de ambas bandas. La figura 4.22 muestra el contorno del patrón direccional total y se observa que en las frecuencias bajas el lóbulo central es más ancho y se va haciendo más delgado conforme la frecuencia aumenta y es mucho mas delgado que el obtenido con el formador de haz fijo.

4.4 Implementación del Sistema en un DSP

Una vez teniendo el resultado de las simulaciones, el sistema se implementó en un DSP TMS320F28335 de Texas Instruments, haciendo las modificaciones necesarias en el algoritmo para su funcionamiento con señales y frecuencias de muestreo reales y adaptándolo al ambiente de programación de la tarjeta tomando en cuenta que se está operando señales muestreadas.

La implementación en el DSP se divide en tres partes, las cuales deben ser consideradas para su funcionamiento y son:

- Los algoritmos.
- El hardware.
- El software.

Los algoritmos fueron presentados en el capítulo 3 y simulados en este capítulo, para escoger el de mejor desempeño.

4.4.1 Hardware general del sistema

El hardware del sistema se muestra en la figura 4.23.

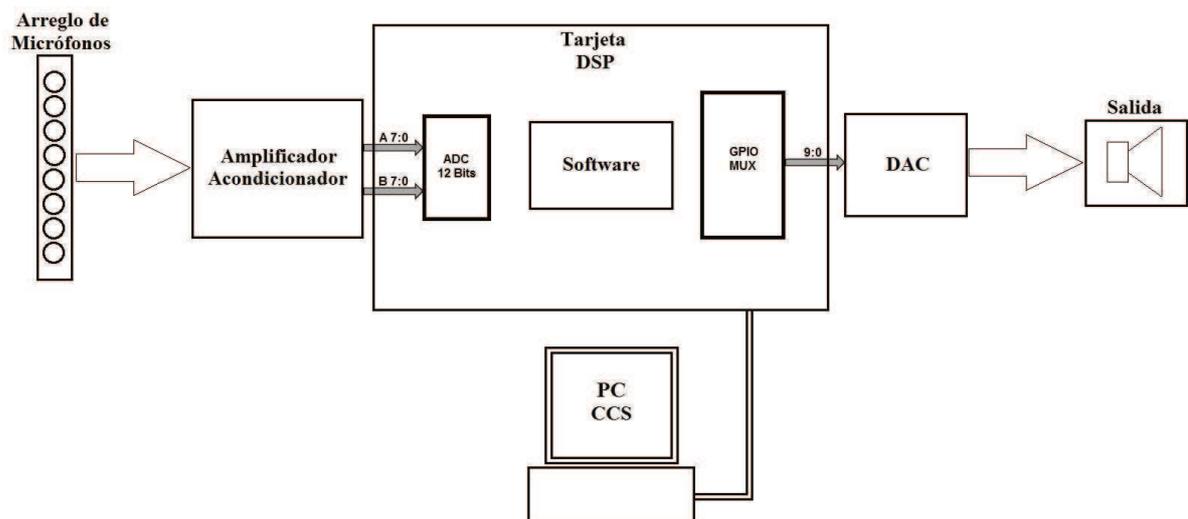


Figura 4.23 Hardware del sistema.

El arreglo de micrófonos

Consta de 15 micrófonos electret, distribuidos en las posiciones descritas en el diseño del arreglo mencionado anteriormente, los cuales son polarizados a través del circuito amplificador y acondicionador.

El circuito amplificador y acondicionador

Se rediseñó a partir del usado en un trabajo de tesis anterior desarrollado, también, en el laboratorio de procesamiento digital de señales [13]. El circuito consta de 16 amplificadores, que también actúan como filtros paso banda, para limitar en banda las señales de los micrófonos y amplificarlas para ser recibidas por el convertidor analógico digital del DSP. La ganancia de los amplificadores es de $A_v = -170$ y las frecuencias de corte del filtro son $f_1 = 15.4\text{Hz}$ y $f_2 = 19.74\text{kHz}$; además, el circuito le suma un voltaje de offset a las señales de $V_{off} = 1.5\text{V}$ para que estas no tengan valores negativos.

La tarjeta DSP es la TMDSDOCK28335

Es la tarjeta de experimentación del TMS320F28335 de Texas Instruments, la cual se programa a través de una PC, mediante una conexión USB y un software IDLE (Interactive DeveLopment Environment), que es un ambiente interactivo de desarrollo llamado Code Composer Studio v4 en el cual se desarrolla el software que forma parte del sistema y el cual se describe posteriormente.

EL DSP TMS320F28335

Tiene un convertidor analógico digital (ADC) que consta de 16 canales de conversión de 12 bits, con una entrada analógica de 0.0V a 3.0V , por lo que es necesario sumar el offset a las señales de los micrófonos como se mencionó anteriormente. Para su configuración es necesario escribir en dos registros de control $ADCTRL1$ y $ADCTRL1$ y cuatro registros de control de selección de secuencia de canal $ADCCHSELSEQ1$, $ADCCHSELSEQ2$, $ADCCHSELSEQ3$ y $ADCCHSELSEQ4$, la configuración se muestra mas adelante en la parte software. EL DSP TMS320F28335 también cuenta con tres timers de 32 bits, de los cuales se utiliza el timer 0 para generar una interrupción dentro de la cual se realizará el procesamiento de los datos provenientes del ADC, la configuración de este timer se presenta mas adelante en la parte de software.

Para visualizar la salida se utilizan los pines de entrada/salida de propósito general (GPIO), los cuales se deben configurar como salidas para poder desplegar los datos de la salida a través de ellos. Una vez que se tuvieron los datos de salida en los pines GPIO, fue necesario implementar una etapa de salida para poder observar estos datos en un osciloscopio o bien escucharlos en unos audífonos. Esta etapa de salida es un convertidor digital analógico (DAC) de 10 bits, el cual toma los datos digitales que se tienen en los pines GPIO y los vuelve señales analógicas para poder oírlas o verlas.

4.4.2 Software del sistema

El software del sistema consta de diferentes procesos mostrados en el diagrama de flujo de la figura 4.24.

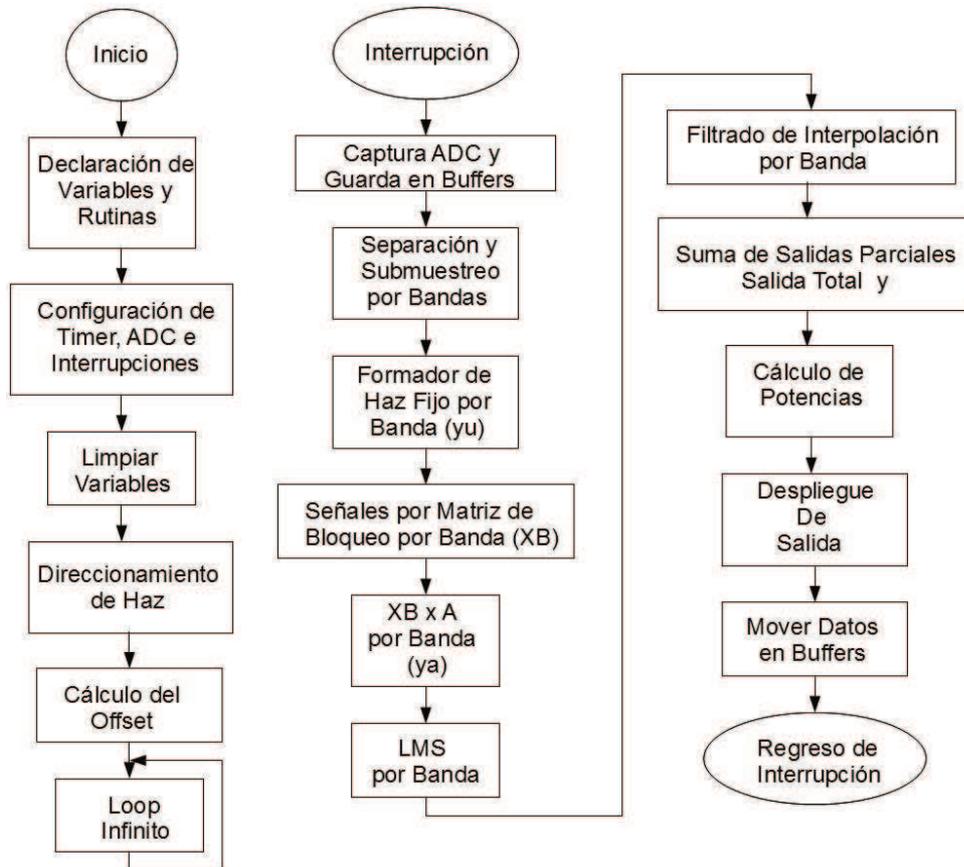


Figura 4.24 Diagrama de flujo del sistema.

Estos procesos se definen a continuación:

Configuración de timer, ADC e interrupciones

En este proceso se configura el timer para que genere una interrupción cada $65\mu\text{seg}$, la cual será la base del sistema ya que esta generará la frecuencia de muestreo de 15384Hz , debido a que el tiempo de muestreo debe ser en microsegundos enteros, y durante esta interrupción se hará el procesamiento de las señales de entrada en los micrófonos. Esta configuración se muestra en el siguiente código:

```

InitCpuTimers(); //Inicializa los Timers del CPU

#if (CPU_FRQ_150MHZ)
//Configura CPU-Timer 0 para interrumpir cada 65 microsegundos:
ConfigCpuTimer(&CpuTimer0, 150, 65);
#endif
#if (CPU_FRQ_100MHZ)
//Configura CPU-Timer 0 para interrumpir cada 65 microsegundos:

```

Capítulo 4. Diseño e Implementación del Sistema.

```
ConfigCpuTimer(&CpuTimer0, 100, 65);
#endif

// Activa Timer e Interrupción
CpuTimer0Regs.TCR.all = 0x4001; //Usar instrucción de solo escritura para TSS
bit=0
```

También se configura el convertidor analógico-digital para que haga conversiones en modo continuo y, en cada interrupción del timer, se lean los datos, se configura para hacer una conversión en modo cascada, es decir, hace la conversión de todos los canales que se habilitan de manera secuencial en el orden en el que se configuran, esta configuración se hizo en orden ascendente del canal 1 al 16, esta configuración se muestra en el siguiente código:

```
InitAdc(); // Inicializa el ADC
AdcRegs.ADCCTRL1.bit.ACQ_PS = ADC_SHCLK; //Tamaño de ventana de adquisición
ADC_SHCLK = 0x1 (ADC_SHCLK+1)*ADCLK
AdcRegs.ADCCTRL3.bit.ADCCLKPS = ADC_CKPS; //Divisor de reloj ADC_CKPS = 0x0
//HSPCLK El reloj que alimenta al ADC
//no se modifica

AdcRegs.ADCCTRL1.bit.SEQ_CASC = 1; //1 Modo Cascada
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x1; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x2; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x3; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x4; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 0x5; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 0x6; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 0x7; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ3.bit.CONV08 = 0x8; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ3.bit.CONV09 = 0x9; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ3.bit.CONV10 = 0xA; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ3.bit.CONV11 = 0xB; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ4.bit.CONV12 = 0xC; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ4.bit.CONV13 = 0xD; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ4.bit.CONV14 = 0xE; //Secuencia de conversión
AdcRegs.ADCCHSELSEQ4.bit.CONV15 = 0xF; //Secuencia de conversión
AdcRegs.ADCCTRL1.bit.CONT_RUN = 1; //Configura modo continuo
AdcRegs.ADCMAXCONV.all = 0xF; //Configura par cada secuencia con 15 SOCs
```

Se configuran también los pines GPIO como salidas, el siguiente código muestra esta configuración.

```
EALLOW;
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0; //LSB
GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 0; //Configura los GPIO como entrada/salida
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO4 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO6 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO7 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO8 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO9 = 0; //MSB
GpioCtrlRegs.GPADIR.bit.GPIO0 = 1; //LSB
```

Capítulo 4. Diseño e Implementación del Sistema.

```
GpioCtrlRegs.GPADIR.bit.GPIO1 = 1;           //Configura los pines GPIOx como salida
GpioCtrlRegs.GPADIR.bit.GPIO2 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO3 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO4 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO5 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO6 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO7 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO8 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO9 = 1;           //MSB
EDIS;
```

Limpiar variables

Este proceso se lleva a cabo mediante una subrutina en lenguaje ensamblador la cual carga un valor de 0 al acumulador A y lo guarda en todas las localidades de memoria de los buffers a utilizar en el algoritmo, asegurando así que todos están inicializados con valor 0. El siguiente código muestra un ejemplo de esta acción:

```
ZAPA                ; Carga cero al acumulador ACC al registro P y a OVC
MOV AL,#0           ; Carga cero a la parte baja del acumulador ACC
MOVL XAR2,#_Buff1   ; XAR2 Apunta a la dirección de Buff1
RPT #63             ; Repite 63+1 veces la siguiente instrucción
| MOV *XAR2++,AL     ; Carga la parte baja de ACC en donde apunta XAR2 y
;pos incrementa
```

Direccionamiento del haz

Este direccionamiento consiste en seleccionar los valores de la matriz de dirección correspondientes a la dirección angular a la que se desea orientar el haz en el algoritmo y colocarlos de manera correcta en los vectores de dirección para cada banda.

Cálculo del offset

Debido a que el convertidor analógico-digital solo acepta voltajes entre 0V y 3V es necesario sumarles un offset a las señales de los micrófonos en el circuito amplificador para que solo tengan valores de voltaje positivos y en el intervalo de conversión antes mencionado. Al operar los datos en el DSP es necesario eliminar este offset para tener los valores reales de las señales, para esto se muestrea 512 veces uno de los canales del convertidor que corresponde al micrófono central del arreglo mientras se tiene un ambiente de silencio ya que se quiere medir solo el offset en el amplificador, éstas muestras se suman, se promedian y se guarda este valor en una localidad de memoria, el siguiente código muestra esta acción:

```
for (i=0; i<512; i++)           //Repite 512 veces
{
    soff = soff + (AdcRegs.ADCRESULT8>>4); //Suma entrada del canal 4 ADC
}
soff = soff >> 9;                 //Divide suma entre 512
off = (int)soff;                 //Guarda valor en 16 bits
```

Interrupción

Después de realizar los procesos descritos el sistema entra en un ciclo infinito en espera de la interrupción del timer. Cuando esta interrupción se presenta se deben realizar todos los procesos que operan sobre las señales en los $65\mu\text{seg}$ disponibles entre cada interrupción.

Captura ADC y guarda en buffers

En este proceso se leen los datos presentes en el convertidor analógico-digital y se guardan en los 15 buffers de memoria y se les resta el offset calculado en un proceso anterior, los buffers son de 51 localidades para la implementación de los filtros FIR de separación de bandas. La señal actual se guarda en la localidad 50 del buffer, es decir, al final del buffer. La figura 4.25 muestra el ordenamiento de los datos en los buffers.

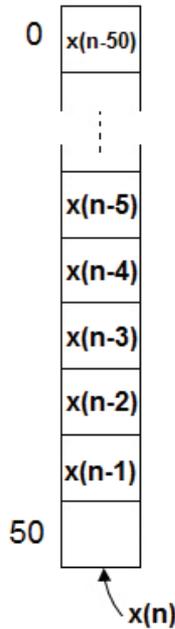


Figura 4.25 Recepción de datos en los buffers.

Separación y submuestreo por bandas

Para hacer la separación y submuestreo por bandas se tienen unas variables que corresponden a los índices de submuestreo por banda, recordando que la banda 4 no se submuestra, la banda 3 se submuestra a la mitad de la frecuencia de muestreo, la banda 2 a un cuarto de la frecuencia de muestreo y la banda 1 a un octavo de la frecuencia de muestreo. Estas variables se incrementan en cada interrupción y dependiendo su valor se hace el filtrado de cada banda o no. Si el índice de la banda 1 tiene un valor de 8 entonces se aplica el filtro FIR sobre los datos guardados en los buffers que corresponden a las señales de los micrófonos contenidos en el arreglo de la banda 1 y el valor resultante se guarda en los buffers de memoria que contienen los datos filtrados por banda y el valor del índice se regresa a 0 para empezar el conteo de interrupciones nuevamente, cuando el índice tiene un valor diferente de 8 el filtro FIR no se aplica y el valor que se guarda en los buffers de memoria es 0, para la banda 2 el valor del índice debe ser 4 para realizar el filtrado y para la banda 3 el valor del índice debe ser 2, la banda 4 se filtra siempre en cada interrupción. Los números de los buffers van de acuerdo a la posición del micrófono al que corresponden en el arreglo, esto se muestra en la figura 4.26.



Figura 4.26 Números de los micrófonos y buffers.

Los buffers que se operan por cada banda son los siguientes:

Banda	Buffers
1	1, 4, 12, 15
2	2, 4, 8, 12, 14
3	3, 4, 6, 8, 10, 12, 13
4	4, 5, 6, 7, 8, 9, 10, 11, 12

Tabla 4.1 Buffers de datos por banda.

Los filtros FIR se implementaron como subrutinas en lenguaje ensamblador mediante la instrucción *MAC* (multiplicación-acumulación) que multiplica y suma el resultado en cada iteración de la instrucción y debido a que los filtros FIR son simétricos en sus coeficientes el orden de los datos en el buffer no afecta la operación de éstos.

La ecuación en diferencias de un filtro FIR esta dada por [14]:

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \quad (4.13)$$

donde

$y(n)$ es la muestra actual de la señal filtrada, $x(n-k)$ son las muestras de la señal a filtrar, b_k son los coeficientes del filtro FIR y M es la longitud del filtro.

El siguiente código muestra la implementación de un filtro FIR a partir de la ecuación (4.13); los coeficientes del filtro están en formato de punto fijo q_{15} y los datos de entrada se manejan en formato q_{12} [15].

```

MOVL XAR2, #_Buff1           ;Apunta XAR2 a Buff1
MOVL XAR7, #_h1             ;Apunta XAR7 a h1
MOVL XAR4, #_fill1         ;Apunta XAR4 a fill1
ADDB XAR4, #5h              ;Mueve XAR4 5 localidades
SPM 0                       ;Product shift mode en 0
ZAPA                        ;Carga cero al acumulador ACC al registro P y a OVC
RPT #50                     ;Repite 50 + 1 veces la siguiente instrucción
| | MAC P, *XAR2++, *XAR7++   ;Multiplica y acumula Buff1 por h1
ADDL ACC, P<<PM             ;Suma ACC + P con corrimiento de acuerdo a PM
MOVH *XAR4, ACC<<1         ;Hace el ajuste del qi=12 al ACC y lo salve en dir
                               ;fill1
    
```

Formador fijo por banda y_u

Si el índice de submuestreo de una banda tiene valor de 0, significa que se ha realizado el filtrado para separar dicha banda, dado este caso se procede a calcular la salida de un formador de haz fijo por cada banda, el cual se obtiene primero tomando las muestras de las señales filtradas con un retraso definido por los valores del vector de dirección por cada banda, multiplicándolas por el valor de las ventanas y sumando los resultados para obtener un solo valor y_u como se muestra en la figura 4.27.

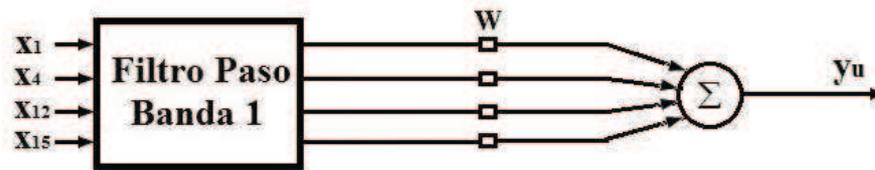


Figura 4.27 Formador de haz fijo.

Las muestras de las señales filtradas con retraso se guardan en variables para su uso en el siguiente proceso, el siguiente código muestra el formador de haz fijo para la banda 1.

```
ret = 5 - dir1[0]; //Calcula el retraso en localidades
x11 = fil11[ret]; //Guarda el valor para usarlo después
aux = (long) fil11[ret] * (long) v1[0]; //Multiplica valor por ventana
ret = 5 - dir1[1]; //Calcula el retraso en localidades
x12 = fil12[ret]; //Guarda el valor para usarlo después
aux = aux + (long) fil12[ret] * (long) v1[1]; //Multiplica valor por ventana y
//suma al valor anterior
ret = 5 - dir1[2]; //Calcula el retraso en localidades
x13 = fil13[ret]; //Guarda el valor para usarlo después
aux = aux + (long) fil13[ret] * (long) v1[2]; //Multiplica valor por ventana y
suma al valor anterior
ret = 5 - dir1[3]; //Calcula el retraso en localidades
x14 = fil14[ret]; //Guarda el valor para usarlo después
aux = aux + (long) fil14[ret] * (long) v1[3]; //Multiplica valor por ventana y
//suma al valor anterior
aux = aux >> 12; //Hace ajuste de qi=12
```

Para las bandas restantes es el mismo proceso pero con un número mayor de micrófonos.

Señales de la matriz de bloqueo por banda (X_B)

Las muestras de la señal filtrada con retraso, que se guardaron en variables en el proceso anterior, se multiplican por la matriz de bloqueo B de cada banda; sin embargo, al ver los valores de estas matrices, se puede deducir que el resultado de esta multiplicación es un vector cuyos valores son la resta de las variables X_B , como se muestra en la figura 4.28.

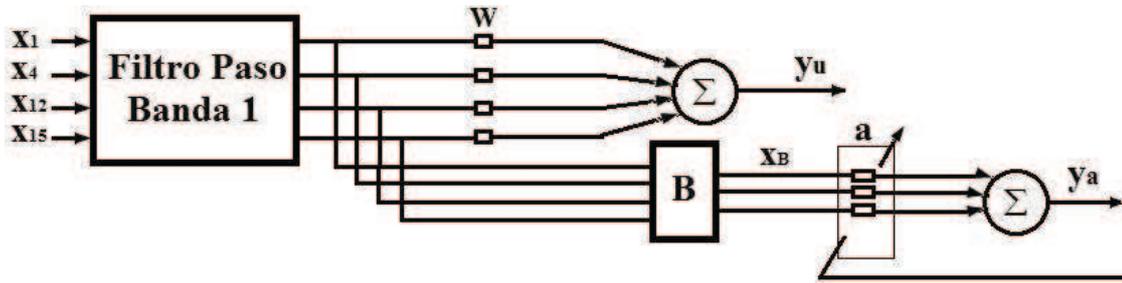


Figura 4.28 Matriz de bloqueo y vectores \mathbf{a} .

El resultado para la banda 1 se puede apreciar en el siguiente código:

```
B1[0] = x11 - x12; //Resta x12 a x11 y lo guarda en B1
B1[1] = x12 - x13; //Resta x13 a x12 y lo guarda en B1
B1[2] = x13 - x14; //Resta x14 a x13 y lo guarda en B1
```

Para las bandas restantes es el mismo proceso pero con un número mayor de variables.

Señales X_B x A por banda ya

Una vez calculados los vectores X_B por banda, éstos se multiplican por los vectores \mathbf{a} de cada banda y se suman los resultados para obtener el valor y_a . Este valor se le resta al valor y_u , como muestra la figura 4.28, para obtener una salida parcial de un formador GSC por banda y se guarda en la localidad 50 de los buffers asignados para estos datos por banda, a los cuales se les aplicará el filtro FIR de interpolación. Si el índice de submuestreo por banda no es igual a 0, a la localidad 50 de estos buffers se le asigna un valor de 0. El siguiente código muestra este proceso para la banda 1.

```
ya = (long) A1[0] * (long) B1[0]; //Multiplica primer valor de A y B
ya = ya + (long) A1[1] * (long) B1[1]; //Mult sig valor de A y B y suma
ya = ya + (long) A1[2] * (long) B1[2]; //Mult sig valor de A y B y suma
ya = ya >> 12; //Ajusta qi
BFb1[50] = (int) aux - (int) ya; //Salida parcial 1 GSC
```

Para las bandas restantes es el mismo proceso pero con un número mayor de variables.

Algoritmo LMS por banda

Con el valor de la salida parcial de un formador GSC por banda se pueden actualizar los valores de los vectores A de cada banda utilizando el algoritmo LMS definido por la ecuación (3.2)

$$\mathbf{a}_{k+1}(f) = \mathbf{a}_k(f) + \mu y_k(f) \mathbf{x}_k^*(f).$$

El siguiente código muestra este proceso para la banda 1.

```
aux = (long) BFb1[50] * (long) B1[0]; //Multiplica salida BFb1 por vector B
aux = aux >> 12; //Ajusta qi
aux2 = (int) aux; //Guarda valor en 16 bits
aux = (long) miu1 * (long) aux2; //Multiplica valor anterior por paso
aux = aux >> 12; //Guarda valor en 16 bits
A1[0] = A1[0] + (int) aux; //Actualiza valor de A1[0]
```

Capítulo 4. Diseño e Implementación del Sistema.

```
aux = (long) BFb1[50] * (long) B1[1]; //Multiplica salida BFb1 por vector B
aux = aux >> 12; //Ajusta qi
aux2 = (int) aux; //Guarda valor en 16 bits
aux = (long) miu1 * (long) aux2; //Multiplica valor anterior por paso
aux = aux >> 12; //Guarda valor en 16 bits
A1[1] = A1[1] + (int) aux; //Actualiza valor de A1[1]

aux = (long) BFb1[50] * (long) B1[2]; //Multiplica salida BFb1 por vector B
aux = aux >> 12; //Ajusta qi
aux2 = (int) aux; //Guarda valor en 16 bits
aux = (long) miu1 * (long) aux2; //Multiplica valor anterior por paso
aux = aux >> 12; //Guarda valor en 16 bits
A1[2] = A1[2] + (int) aux; //Actualiza valor de A1[2]
```

Para las bandas restantes es el mismo proceso pero con un número mayor de variables.

Filtrado de interpolación por banda

Para poder sumar las salidas parciales de los formadores de haz GSC es necesario interpolar los buffers donde están estos datos a través de un llenado de ceros y un filtro FIR ponderando los coeficientes del filtro, dado que se submuestreó la entrada en cada banda, se tendrá de nuevo la misma cantidad de datos de salida en cada banda. La amplitud de los datos interpolados debe ajustarse ya que esta decrece en proporción al índice de submuestreo de cada banda, es decir, si la entrada se submuestreó a la mitad de la frecuencia de muestreo, la amplitud de los datos de salida al interpolar se reduce a la mitad; si se submuestreó a un cuarto de la frecuencia de muestreo, la amplitud de los datos de salida al interpolar se reduce a un cuarto, etc., esta amplitud se ajusta en el filtro de interpolación al hacer los corrimientos para ajustar el formato q_i del resultado y se usan los mismos coeficientes de los filtros paso banda que se utilizaron para separar los datos por bandas de frecuencia. La implementación es similar a la mostrada en la descripción de la etapa de separación y submuestreo por bandas.

Suma de salidas parciales, salida total y

Una vez que se interpolan las salidas parciales de los formadores de haz GSC de cada banda, éstas se suman para obtener la salida total del sistema. El siguiente código muestra este proceso.

```
sal[i] = BInt1 + BInt2; //Suma la salida 1 a la salida 2
sal[i] = sal[i] + BInt3; //Suma el valor anterior a salida 3
sal[i] = sal[i] + BInt4; //Suma el valor anterior a salida 4
```

Cálculo de potencias

Para determinar si el sistema está filtrando correctamente las señales, se deben calcular las potencias de la señal de salida para diferentes posiciones angulares a las que se dirige el algoritmo. Para esto es necesario ir calculando el cuadrado de la salida final en cada interrupción y sumarlo en una variable, cuando se tengan 512 sumas se divide entre 512 para obtener una potencia de referencia y poder compararla con la que resulte en las diferentes posiciones angulares en las que se pruebe el algoritmo, se calcula la potencia de la salida total, así como la potencia de las salidas parciales por banda. La potencia promedio por banda se calcula como

$$P_l = \frac{\sum_{n=1}^{512} y_l^2(n)}{512}$$

donde $l=1,2,3,4$ indica el número de la banda.

Despliegue de salida

Después de realizar todo el procesamiento de las señales se obtiene una salida la cual necesita ser desplegada de alguna manera para comprobar el funcionamiento del sistema, por lo que se hizo una etapa de despliegue, la cual consiste en los pines de salida configurados en el programa principal y un circuito de conversión que incluye un convertidor digital-analógico (DAC). En el programa se debe ajustar la salida para que pueda ser convertida a una señal analógica, al igual que el ADC, el DAC debe tener como entrada señales positivas por lo que a las salida del algoritmo se le debe sumar de nuevo el offset que se calculó y hacerle un corrimiento de 3 bits a la izquierda para que pueda ser desplegado por el DAC ya que este último es de 10 bits de resolución, este valor de la salida con corrimiento se escribe en un registro el cual toma este valor y lo dirige hacia los pines de salida bit por bit.

Mover datos en buffers

Una vez que se realizaron todos los procesos del sistema sobre los datos y se tiene la salida total, se deben mover los datos en los buffers de memoria a una localidad para dejarlos listos en la siguiente interrupción. El siguiente código en lenguaje ensamblador muestra como se mueven los datos en el buffer de datos del micrófono 1.

```
MOVL XAR2,#_Buff1           ;XAR2 Apunta a Buff1
ADDB XAR2,#1h               ;Incrementa XAR2 en 1
RPT #50                     ;Repite 50 + 1 veces
| |MOV *(0:_Buff1),*XAR2++   ;Mueve datos de Buff1 una localidad
                               ;Para efectuar un retardo
```

Regreso de interrupción

Al terminar de procesar los datos se debe regresar de la interrupción para que pueda entrar el sistema de nuevo al ciclo infinito en espera de la próxima interrupción, y se debe habilitar de nuevo la interrupción, de lo contrario el sistema no acepta otra interrupción y se quedara en el ciclo infinito. La instrucción que habilita de nuevo la interrupción se presenta a continuación.

```
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1; //Registra interrupción en PIE
```

Debido a que el lenguaje C genera muchas instrucciones para procesos simples se hicieron subrutinas en lenguaje ensamblador para minimizar el número de instrucciones por proceso, ya que todo el procesamiento de las señales se debe hacer en un intervalo de tiempo fijo correspondiente a un periodo de muestreo y si se hacen todos los procesos en lenguaje C se puede tener un número tan grande de instrucciones que no pueda ser ejecutado en el tiempo disponible.

4.5 Resumen.

En este capítulo se presentó el procedimiento de diseño e implementación del arreglo de micrófonos utilizado en el sistema a partir de los cálculos de los parámetros de frecuencia de muestreo, número de micrófonos y distancia entre ellos por cada banda.

Se muestran las simulaciones de los algoritmos de formador de haz que se realizaron a fin de seleccionar el mejor desempeño.

También se presenta la estructura general del sistema implementado en el DSP mostrando el hardware y el diagrama de flujo general del software del sistema que muestra los procesos que éste debe realizar y se describe su funcionamiento e implementación.

Capítulo 5

Pruebas y Resultados



En este capítulo se describen las pruebas que se realizaron al sistema a fin de comprobar su funcionamiento y los resultados que se obtuvieron.

Se presentan los patrones direccionales por banda y total del sistema implementado en el DSP y se hace un análisis del desempeño en cuanto a los recursos del DSP que son utilizados durante la ejecución del proceso.

5.1 Pruebas al Sistema

A fin de obtener un sistema funcional se realizaron diversas pruebas básicas al sistema implementado en el DSP para comprobar su funcionalidad por partes, es decir, se probaron bloques de código con una función en específico y se comprobó su funcionamiento correcto y en algunos casos se modificaron hasta obtener mejores resultados.

5.1.1 Operaciones

Al implementar el sistema en el DSP se adecuaron las diferentes etapas del software para operar con números enteros de 16 bits y se comprobó que las operaciones funcionaran correctamente, sobre todo la multiplicación entre dos números enteros de 16 bits y que el resultado fuera un número de 16 bits también en formato de punto entero q_{12} , tanto en lenguaje ensamblador como en lenguaje C en el programa principal. Las multiplicaciones de dos números de 16 bits resultan en un número de 32 bits por lo cual se debe ajustar este resultado para tener de nuevo números de 16 bits para seguir operando los datos conforme se siguen procesando.

En lenguaje ensamblador las operaciones de multiplicación y suma necesarias para implementar los filtros se realizan con la función *MAC* y solo fue necesario hacer el ajuste del formato $q_i^{(1)}$ para tener resultados en q_{12} .

Para las multiplicaciones en lenguaje C se realizó una pequeña rutina para comprobar su funcionamiento, ya que se debe tomar en cuenta el tipo de variable que se está usando en la operación, los operandos y el resultado fueron declarados como enteros de 16 bits y se debe tener una variable auxiliar declarada como entero de 32 bits la cual guardará el resultado de la multiplicación en 32 bits antes de ajustarlo y guardarlo en la variable de resultado de 16 bits. El siguiente código muestra el proceso para hacer una multiplicación básica en lenguaje C de dos números de 16 bits en q_{12} para obtener también un resultado de 16 bits y q_{12} :

```
int y = 6144;           //Declara variable y = 1.5 en q12
int x = 6144;           //Declara variable x = 1.5 en q12
int z = 0;              //Declara variable z = 0
long int aux;          //Declara variable aux de 32 bits

aux = (long) x * (long) y; //Multiplica x por y
aux = aux >> 12;         //Ajusta qi a q12
z = (int) aux           //Guarda aux en z a 16 bits
```

Este código con una modificación se probó también para realizar el algoritmo LMS, el cual requiere multiplicar tres números de 16 bits.

⁽¹⁾ El formato de punto entero q_i significa que para una longitud de palabra L en la representación de una constante o variable decimal se utilizan i bits para la parte fraccionaria, $L-i-1$ bits para la parte entera y un bit para el signo [15].

5.1.2 Prueba de etapas de entrada y salida

Para determinar si el sistema está recibiendo correctamente las señales provenientes de los micrófonos se hizo una rutina la cual toma 256 datos de cada micrófono y los guarda en 15 buffers para las señales recibidas, esta prueba permitió comprobar la correcta configuración de la interrupción y del ADC ya que se observó que en los buffers de memoria se tenían datos correspondientes a la señal que se hizo llegar a los micrófonos a través de unas bocinas.

Para probar la etapa de salida se configuró el sistema para cargar una señal senoidal a la memoria del DSP y éste en cada interrupción mandara una muestra a la etapa de salida, después se modificó el sistema a manera de que tomara los datos de los micrófonos y los mandara por uno de ellos hacia la etapa de salida y se observó en el osciloscopio que la salida coincidiera con la señal alimentada a los micrófonos a través de las bocinas

5.2 Pruebas con Señales Fijas

Antes de probar el sistema con señales acústicas reales, se probó con señales fijas ideales generadas para estas pruebas, a fin de observar el comportamiento del sistema y determinar algunas características ideales del mismo.

Las señales fueron generadas a través de un oscilador digital basado en un sistema de segundo orden cuya función de transferencia es [14]:

$$H(z) = \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

donde

$$a_1 = -2r \cos \omega_0 \text{ y } a_2 = r^2.$$

Este sistema tiene polos complejos conjugados en $p = re^{\pm j\omega_0}$ y respuesta unitaria es:

$$h(n) = \frac{b_0 r^n}{\text{sen} \omega_0} \text{sen}(n+1) \omega_0 u(n)$$

Si los polos se colocan en $r=1$ y $b_0 = A \text{sen} \omega_0$ la respuesta es:

$$h(n) = A \text{sen}(n+1) \omega_0 u(n)$$

Entonces la respuesta al impulso de un sistema de segundo orden con polos complejos conjugados en el círculo unitario es una señal senoidal y el sistema es llamado oscilador senoidal digital y su ecuación en diferencias es [14]:

$$y(n) = -a_1 y(n-1) - y(n-2) + b_0 \delta(n)$$

donde:

$a_1 = -2 \cos \omega_0$, $b_0 = A \sin \omega_0$, $y(-1) = y(-2) = 0$, $\omega_0 = 2\pi \frac{f_0}{f_s}$, ω_0 : frecuencia de oscilación en radianes. f_0 : frecuencia de oscilación analógica en Hz y f_s : frecuencia de muestreo en Hz.

Un diagrama a bloques del oscilador se muestra en la figura 5.1.

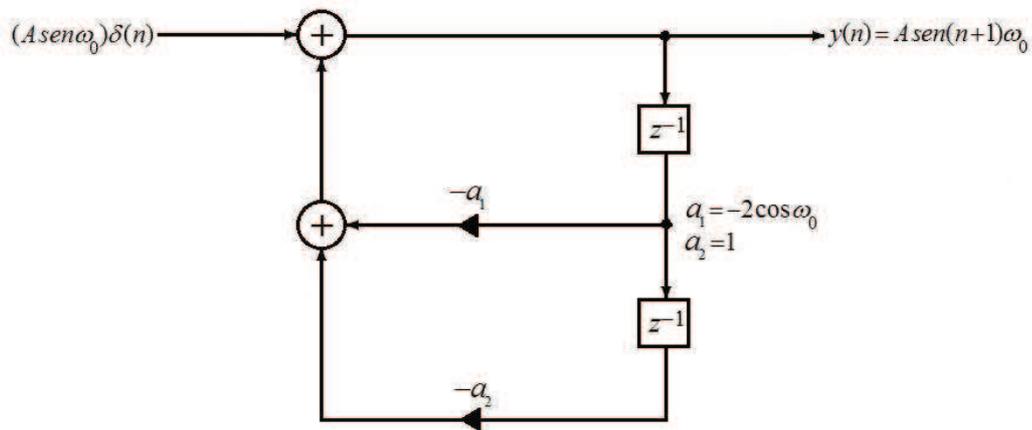


Figura 5.1 Oscilador digital.

5.2.1 Patrón direccional teórico

El algoritmo implementado en el DSP se probó para determinar el patrón direccional teórico por banda y del sistema total comprobando que el sistema efectivamente realizará un filtrado espacial de las señales acústicas. Para esta prueba se generaron cuatro señales senoidales de frecuencia igual a la frecuencia central de cada banda, teniendo $f_1 = 400 \text{ Hz}$, $f_2 = 800 \text{ Hz}$, $f_3 = 1600 \text{ Hz}$ y $f_4 = 3200 \text{ Hz}$ por cada banda respectivamente y se procesaron individualmente para obtener el patrón direccional por banda, después se procesó la suma de las señales para obtener el patrón direccional total del sistema. Estas pruebas se hicieron cargando estas señales a la memoria del DSP para procesarlas con el sistema implementado con algoritmos de formador de haz GSC y obtener un patrón direccional.

Para obtener el patrón direccional de la primera banda se generó una señal senoidal con frecuencia $f_1 = 400 \text{ Hz}$ y $N = 512$ como entrada del sistema y se procesó haciendo un barrido del formador de haz desde -90° a 90° con incrementos de 5° , el algoritmo se dirigió a 0° y se consideró que la señal estaba siendo recibida también a 0° , es decir, no tiene retrasos en cada uno de los micrófonos, se obtuvo la potencia de la señal de salida para cada ángulo de dicho barrido usando formadores de haz GSC y el patrón direccional obtenido se muestra en la figura 5.2.

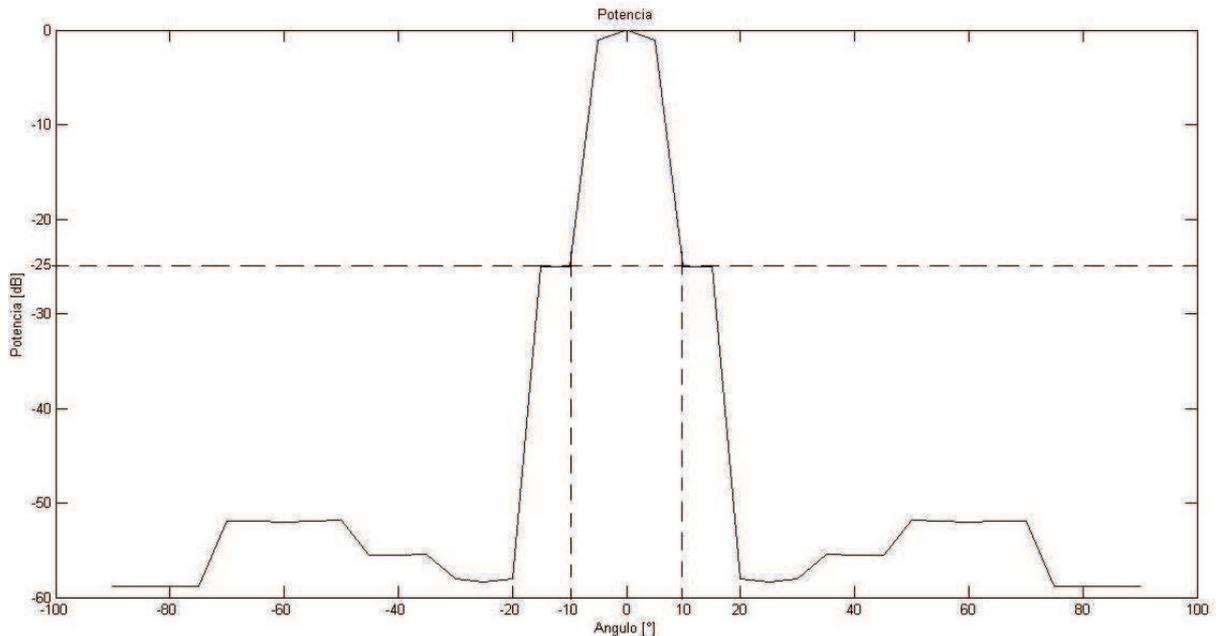


Figura 5.2 Patrón direccional de la banda 1 del sistema usando BF GSC en el DSP.

Al observar la figura se aprecia que el sistema efectivamente filtra las señales de prueba espacialmente, es decir la potencia máxima se presenta en 0° y las demás direcciones son atenuadas, se tienen un umbral para el patrón direccional con una ganancia de -25dB para el límite de lo que podemos considerar el lóbulo principal o el intervalo angular dentro del cual las señales acústicas se conservan y fuera del cual las señales acústicas se atenúan, teniendo un intervalo de 20° ($\pm 10^\circ$) centrado en 0° , se observa también que existen direcciones angulares en las cuales la ganancia es de casi -60dB y se observan lóbulos laterales muy pequeños.

Para obtener el patrón direccional de la segunda banda se repitió el procedimiento anterior pero con una señal de prueba con frecuencia $f_2 = 800\text{ Hz}$ y $N = 512$. La figuras 5.3 muestra el patrón direccional del sistema con formadores de haz GSC en el DSP.

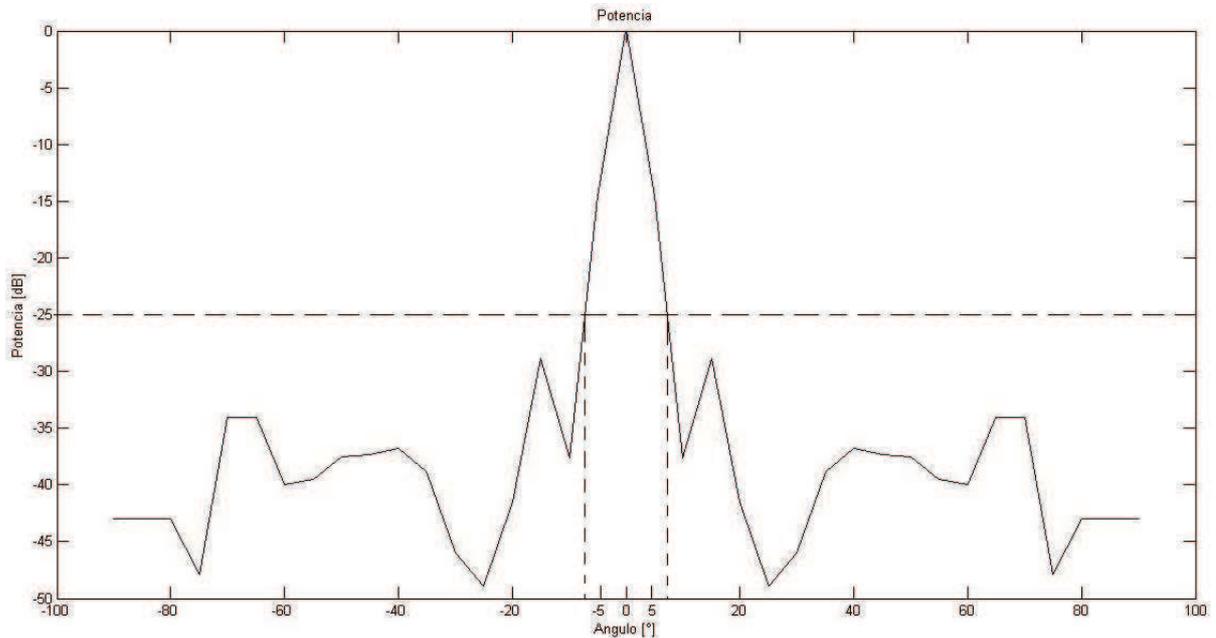


Figura 5.3 Patrón direccional de la banda 2 del sistema usando BF GSC en el DSP.

En la figura 5.3 se aprecia que la potencia máxima se presenta en 0° y las demás direcciones son atenuadas, se tiene un umbral para el patrón direccional obtenido con una ganancia de -25dB para el límite del intervalo angular o lóbulo principal, teniendo un intervalo de 15° ($\pm 7.5^\circ$) centrado en 0° , se observa también la presencia de mas lóbulos laterales que en el patrón direccional de la banda 1, sin embargo, todos están por debajo del umbral establecido y se observan direcciones angulares en donde la ganancia es de casi -50dB .

Para obtener el patrón direccional de la tercera banda se repitió el procedimiento de las dos bandas anteriores pero con una señal de prueba con frecuencia $f_3 = 1600\text{ Hz}$ y $N = 512$. La figura 5.4 muestra el patrón direccional del sistema con formadores de haz GSC en el DSP.

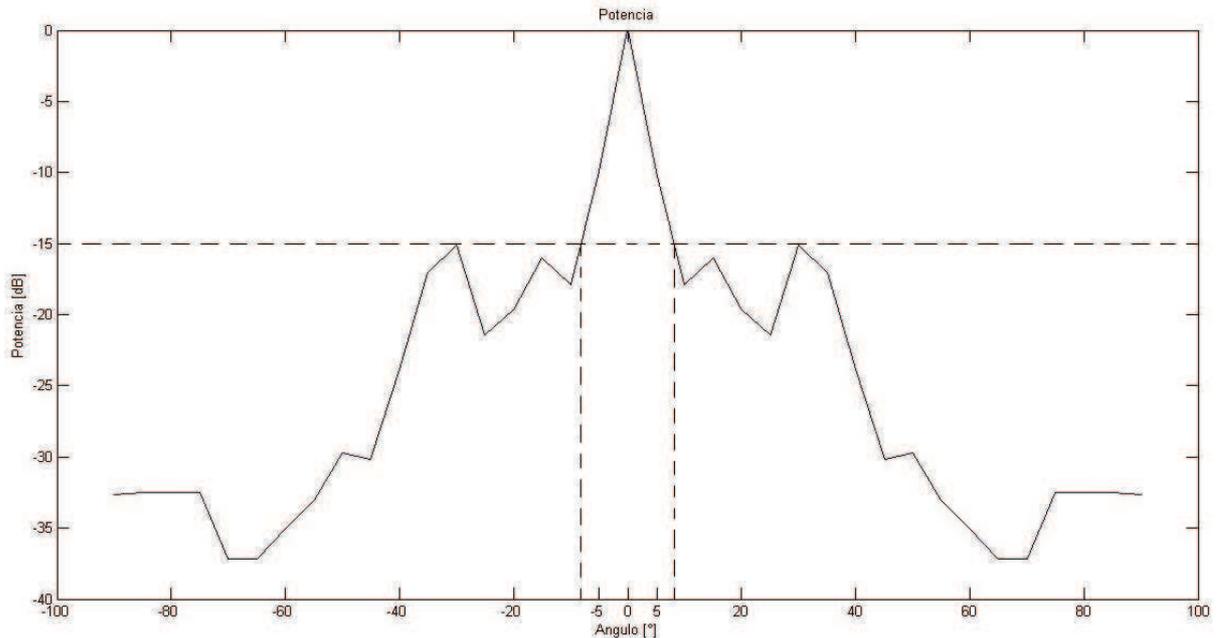


Figura 5.4 Patrón direccional de la banda 3 del sistema usando BF GSC en el DSP.

La figura 5.4 muestra una vez mas que la potencia máxima se presenta en 0° y las demás direcciones angulares son atenuadas, el umbral para el patrón direccional obtenido para esta banda tiene una ganancia de -15dB para los extremos del lóbulo principal teniendo un intervalo de aproximadamente $15^\circ (\pm 7.5^\circ)$ centrado en 0° , los lóbulos laterales son mas pronunciados que en las bandas anteriores y debido a que la ganancia del umbral representa una atenuación menor que en las bandas anteriores, y se observa que la máxima atenuación en algunas direcciones angulares es de mas de 35dB .

Para obtener el patrón direccional de la cuarta banda se repitió el procedimiento anterior pero con una señal de prueba con frecuencia $f_4 = 3200\text{ Hz}$ y $N = 512$. La figura 5.5 muestra el patrón direccional del sistema con formadores de haz GSC en el DSP obtenido para la banda 4.

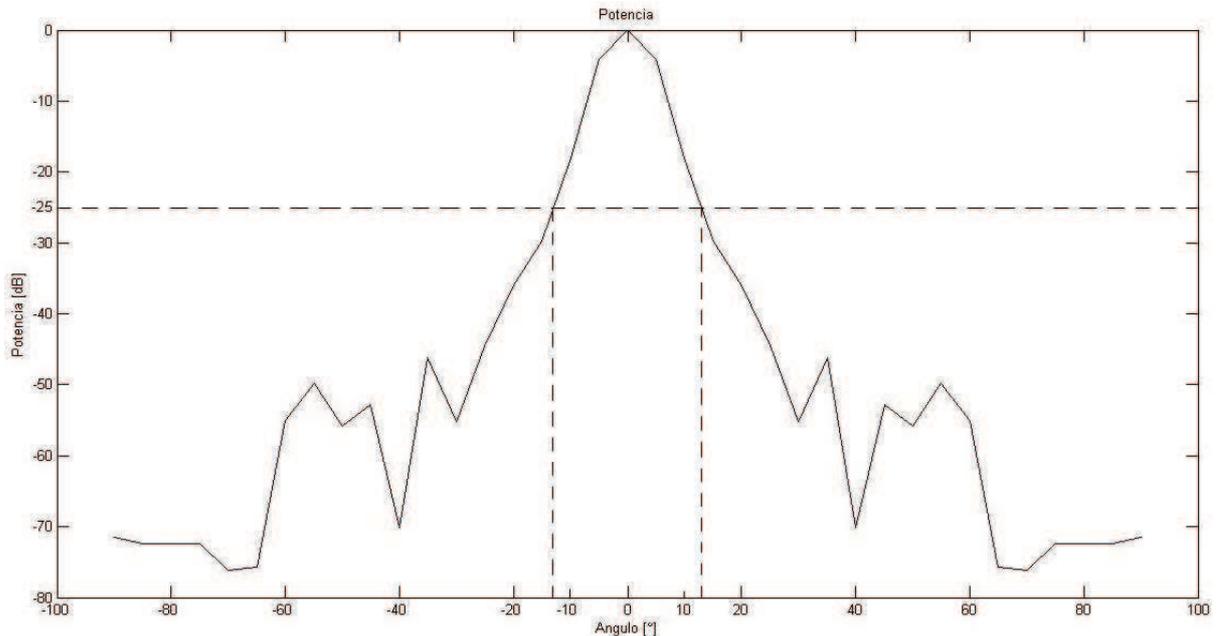


Figura 5.5 Patrón direccional de la banda 4 del sistema usando BF GSC en el DSP.

En la figura 5.5 se muestra que la potencia máxima se presenta en 0° como era esperado y las demás direcciones angulares son atenuadas, el umbral para el patrón direccional obtenido tiene una ganancia de -25dB para el límite del lóbulo principal, teniendo un intervalo de aproximadamente 25° ($\pm 12.5^\circ$) centrado en 0° para el patrón direccional obtenido en el DSP para la banda 4. Los lóbulos laterales están por debajo del umbral establecido y en algunas direcciones angulares la atenuación es mayor a los 70dB .

Una vez que se tienen los patrones direccionales de cada banda se obtuvo el patrón direccional total del sistema, para obtenerlo se repitió el procedimiento que se utilizó para cada banda pero con una señal de prueba que es la suma de las cuatro señales anteriores y $N = 512$. La figura 5.6 muestra el patrón direccional del sistema con formadores de haz GSC en el DSP.

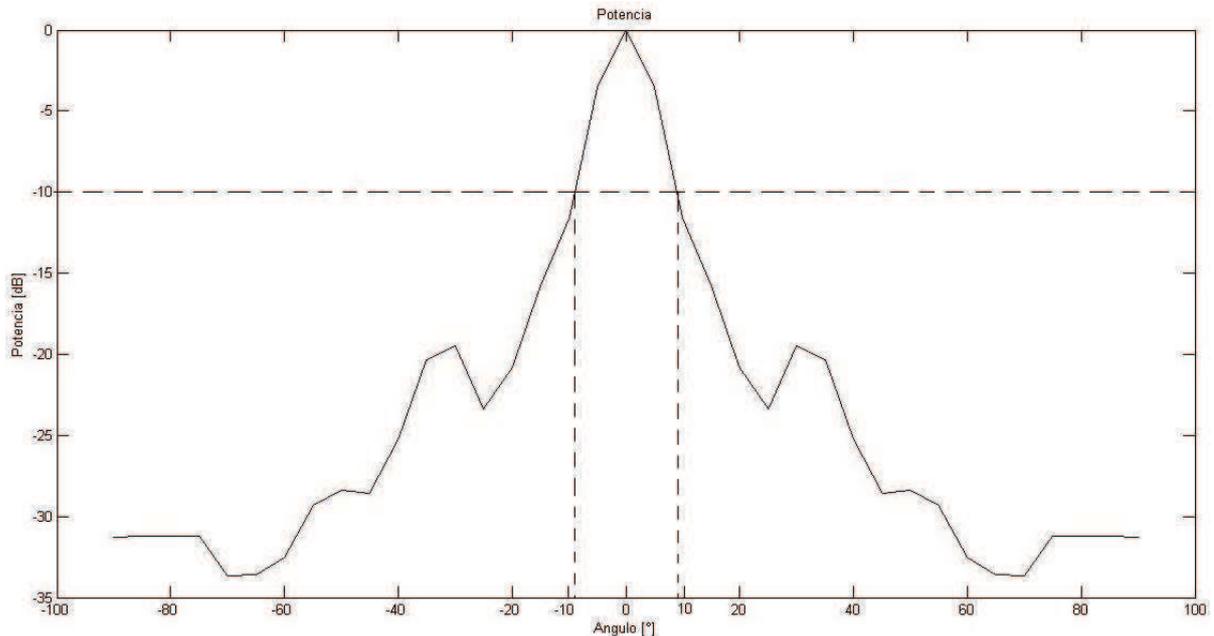


Figura 5.6 Patrón direccional total del sistema usando BF GSC en el DSP.

En la figura 5.6 se aprecia que para el patrón direccional total la potencia máxima se presenta en 0° como en las bandas individuales y las demás direcciones angulares son atenuadas, el umbral para el patrón direccional obtenido tiene una ganancia de -10dB para el lóbulo principal teniendo un intervalo de aproximadamente $10^\circ (\pm 5^\circ)$ centrado en 0° . El patrón direccional presenta lóbulos laterales muy por debajo del umbral y con atenuaciones mayores a 20dB y en algunas direcciones la atenuación llega hasta más de 30dB .

Estos resultados nos muestran que el sistema implementado hace un filtrado espacial de las señales que está procesando, simulando señales acústicas que llegan al arreglo de micrófonos con un ángulo de incidencia de 0° el sistema presenta en todas sus bandas un patrón direccional que indica que las señales provenientes de la dirección angular de interés son conservadas con su amplitud original, mientras que las señales de otras direcciones son atenuadas. Los anchos de los lóbulos principales de los patrones direccionales son estrechos lo que indica una selección angular reducida y las atenuaciones que presentan las señales fuera de este lóbulo principal son grandes indicando que la diferencia de amplitudes entre la señal de interés y las demás señales es muy significativa.

El patrón direccional total del sistema presenta también un lóbulo principal estrecho, sin embargo, el nivel de atenuación de las señales fuera del lóbulo no es tan grande como en las bandas individuales.

Todos estos resultados se obtuvieron suponiendo condiciones ideales sin ruido y sin interferencia y suponiendo señales acústicas sin retrasos en los micrófonos.

5.3 Pruebas con Señales Reales

Una vez probado el sistema con señales ideales, se probó con señales de tonos acústicos reales dirigidas hacia el arreglo de micrófonos mediante el uso de una bocina como muestra la figura 5.7.

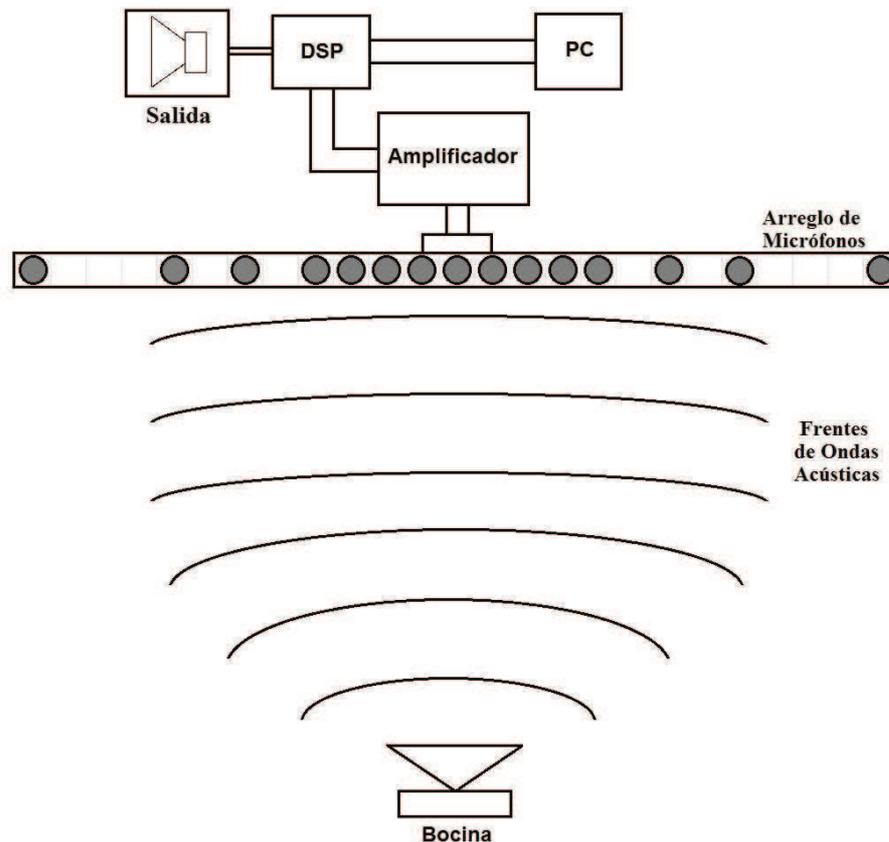


Figura 5.7 Diagrama del sistema para pruebas.

Se colocó la bocina que se observa en la figura 5.7 a 4 metros de distancia del arreglo de micrófonos y con un ángulo de 0° , se hicieron pasar por está las mismas señales senoidales generadas para las pruebas ideales, a fin de obtener el patrón direccional de cada banda y total del sistema en un ambiente con señales e interferencias reales, estas interferencias son el ruido presente en el ambiente y también los rebotes de la señal de prueba en las paredes, techo y piso del laboratorio.

Para cada prueba se hizo un barrido angular, orientando el algoritmo, desde 0° a 90° y se guardaron 512 puntos de la señal de salida para cada ángulo a fin de calcular la potencia de la salida en cada ángulo para obtener un patrón direccional del sistema y observando que el patrón direccional es simétrico cuando las señales acústicas arriban con un ángulo de 0° , se duplicaron los resultados para los ángulos de 0° a -90° para el patrón direccional completo.

5.3.1 Patrón direccional de la banda 1

Para obtener el patrón direccional de la banda 1 se hizo la prueba con la señal senoidal con frecuencia $f_1 = 400 \text{ Hz}$ como entrada del sistema. El resultado se muestra en la figura 5.8.

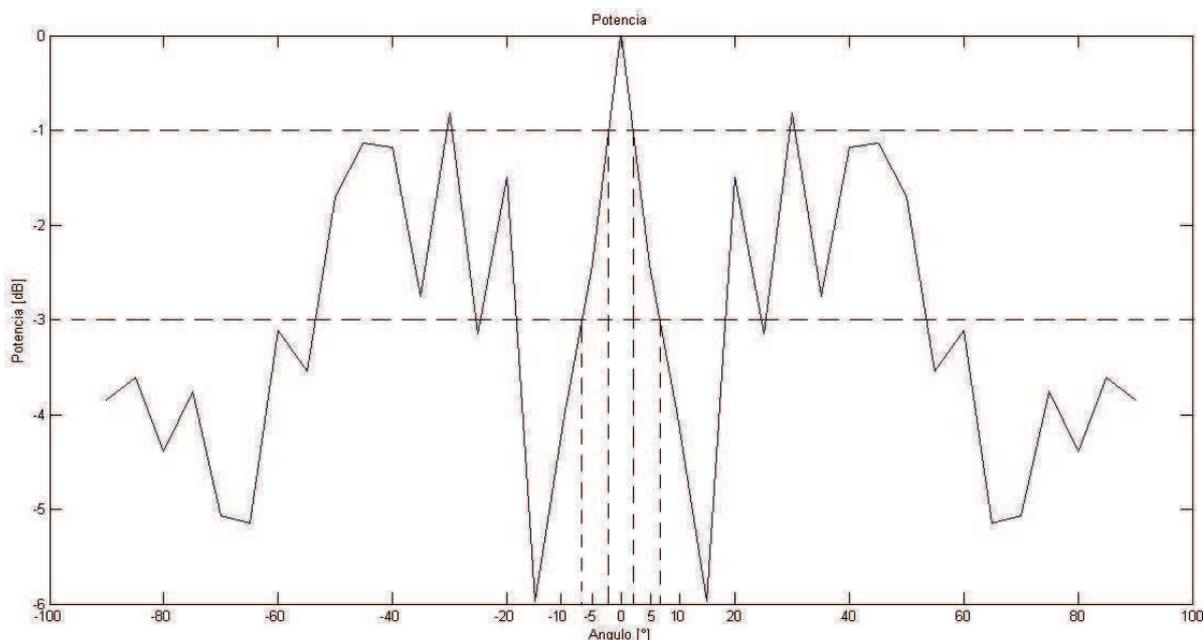


Figura 5.8 Patrón direccional de la banda 1 del sistema en el DSP con señales reales y GSC.

En la figura 5.8 se observa que la salida del sistema para la banda 1 tiene su máxima potencia cuando el ángulo de incidencia de las señales acústicas es igual al ángulo al que está dirigido el algoritmo, es decir 0° , y atenúa las señales que provienen de ángulos diferentes a éste. De acuerdo a la forma del patrón direccional un umbral de ganancia de -1dB define el intervalo angular o lóbulo principal dentro del cual las señales acústicas se conservan y fuera del cual las señales acústicas se atenúan siendo de aproximadamente $5^\circ (\pm 2.5^\circ)$ centrado en 0° . Si se usa un umbral de -3dB este intervalo angular es de aproximadamente $14^\circ (\pm 7^\circ)$ centrado en 0° , sin embargo, existen ángulos fuera de este intervalo en donde las señales no se atenúan debajo del umbral de ganancia establecido y se observan también direcciones angulares con atenuaciones de hasta 6 dB .

La diferencia con el patrón direccional teórico es muy grande debido a las condiciones de prueba ya que se presentan interferencias que son el ruido presente en el laboratorio proveniente del exterior y también el ruido de los aparatos utilizados como la fuente de alimentación, el osciloscopio, la PC e incluso de las lámparas, otra fuente de interferencias son los rebotes de la señal en el piso, techo y paredes del laboratorio, e incluso en el mobiliario, lo cual puede provocar que el sistema detecte mas señales provenientes de diferentes direcciones y genere lóbulos pronunciados en el patrón direccional, sin embargo, estas condiciones son las más frecuentes en la realidad, lo que resulta en una buena prueba para el sistema. También el método de prueba genera estas interferencias, ya que el arreglo de micrófonos está fijo al igual que la fuente de audio, y lo que se va moviendo es la dirección a la cual esta orientado el algoritmo, por lo que si hay un rebote de señal y arriba justo en la dirección a la que esta dirigido el algoritmo, el

sistema dejará pasar esta señal completamente y se generara un pico de alta potencia en el patrón direccional.

5.3.2 Patrón direccional de la banda 2

Para el patrón direccional de la banda 2 se hizo la misma prueba que para la banda 1 pero con la señal senoidal con frecuencia $f_2 = 800 \text{ Hz}$ como entrada del sistema. El resultado se muestra en la figura 5.9.

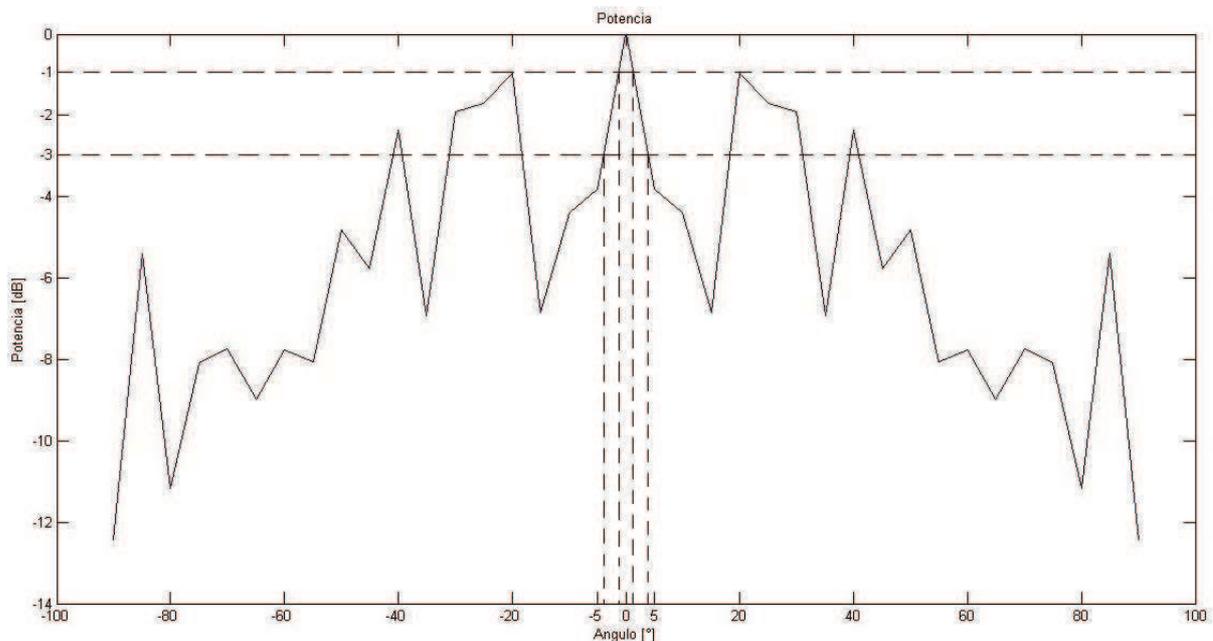


Figura 5.9 Patrón direccional de la banda 2 del sistema en el DSP con señales reales y GSC.

En la figura 5.9 se observa que la salida del sistema para la banda 2 tiene también su máxima potencial a un ángulo de 0° , y que las señales que provienen de ángulos diferentes a éste se ven atenuadas por el sistema. Se tiene un umbral de ganancia de -1dB que define el lóbulo principal siendo de aproximadamente $2^\circ (\pm 1^\circ)$ centrado en 0° . Si se considera un umbral de -3dB el intervalo angular es entonces de aproximadamente $8^\circ (\pm 4^\circ)$ centrado en 0° , sin embargo, existen ángulos fuera de este intervalo en donde las señales no se atenúan debajo del umbral, en algunas direcciones se presentan atenuaciones de entre 8dB hasta 12dB .

El patrón direccional obtenido difiere del patrón teórico debido a las interferencias que se presentan al hacer las pruebas en condiciones reales y que se mencionaron anteriormente, también el número de micrófonos en esta banda es un factor, ya que aunque teóricamente el algoritmo presente un comportamiento muy bueno, al tener condiciones reales el comportamiento cambia, sin embargo, sería mucho mejor si el número de micrófonos para cada banda fuera mayor, en nuestro caso esto no es posible por las limitaciones en el número de entradas del ADC del DSP.

5.3.3 Patrón direccional de la banda 3

Para el patrón direccional de la banda 3 se hizo la prueba que con las bandas anteriores pero con la señal senoidal con frecuencia $f_3 = 1600 \text{ Hz}$ como entrada del sistema. El resultado se muestra en la figura 5.10.

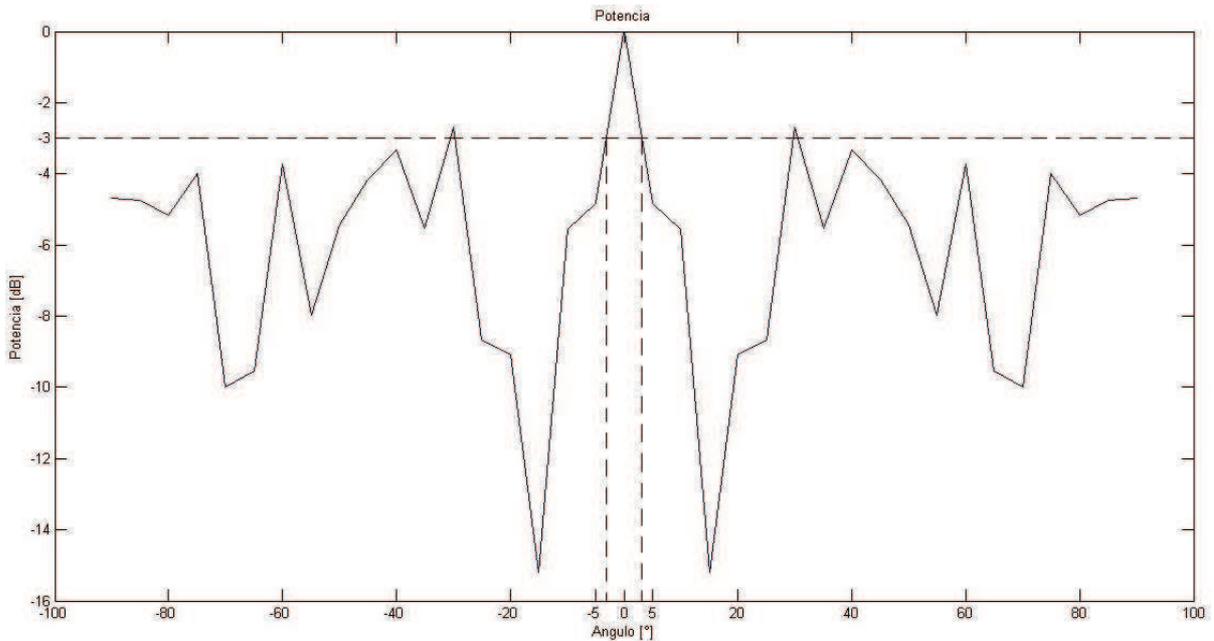


Figura 5.10 Patrón direccional de la banda 3 del sistema en el DSP con señales reales y GSC.

La figura 5.10 muestra que también la salida del sistema para la banda 3 tiene la máxima potencia cuando el ángulo de incidencia de las señales acústicas es igual a 0° , y que las señales que provienen de ángulos diferentes tienen una ganancia menor con un umbral para el patrón direccional de ganancia de -3dB definiendo el ancho del lóbulo principal fuera del cual las señales acústicas presentan la atenuación mencionada siendo de aproximadamente $6^\circ (\pm 3^\circ)$ centrado en 0° , se muestra también que existen direcciones angulares con atenuaciones de hasta más de 14dB .

Una vez más el patrón direccional difiere del patrón teórico obtenido, sin embargo, la respuesta del sistema para esta banda es mejor que en las anteriores ya que el número de micrófonos usados en esta banda es mayor por lo que el sistema presenta un mejor comportamiento en condiciones reales en presencia de las interferencias antes mencionadas.

5.3.4 Patrón direccional de la banda 4

Para el patrón direccional de la banda 4 se hizo la prueba con la señal senoidal con frecuencia $f_4 = 3200 \text{ Hz}$ como entrada del sistema. El resultado se muestra en la figura 5.11.

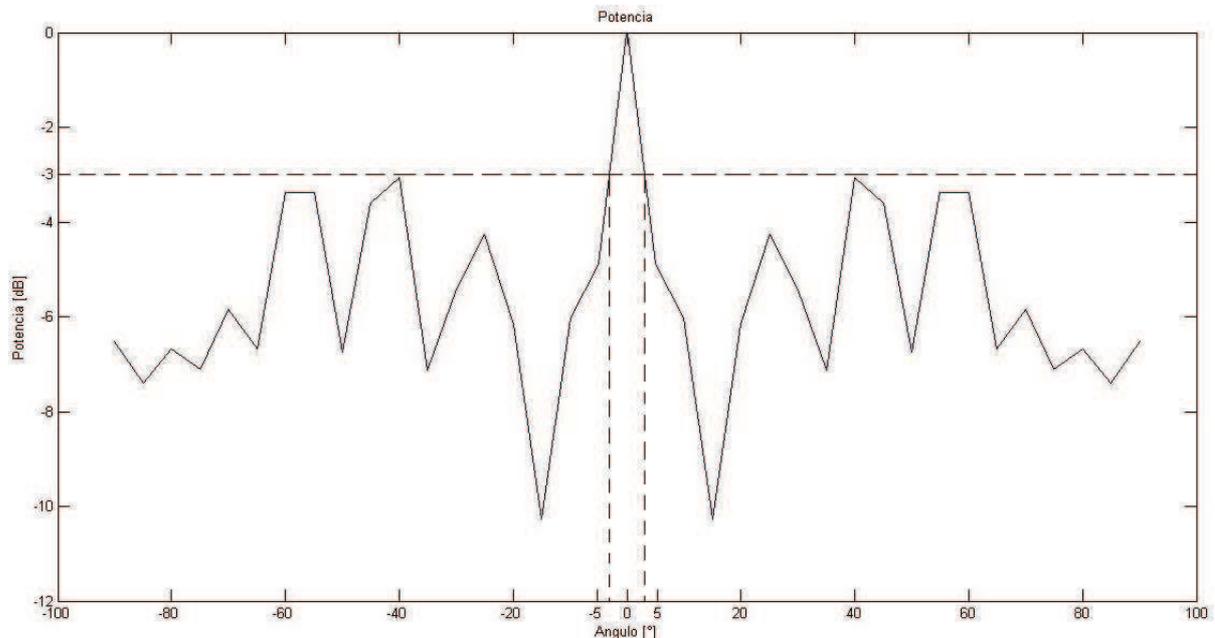


Figura 5.11 Patrón direccional de la banda 4 del sistema en el DSP con señales reales y GSC.

En la figura 5.11 se observa que el patrón direccional del sistema para la banda 4 tiene su máxima potencia para un ángulo de incidencia de las señales de 0° y la atenuación de las señales que provienen de ángulos diferentes a éste. Considerando los valores del patrón direccional se tiene un umbral de ganancia de -3dB que define el intervalo angular del lóbulo principal fuera del cual las señales acústicas se atenúan siendo dicho intervalo de aproximadamente $6^\circ (\pm 3^\circ)$ centrado en 0° , algunas direcciones angulares presentan atenuaciones mayores a 6dB y en algunos casos de hasta 10dB .

Para esta banda el comportamiento del sistema es mejor que para las bandas anteriores debido que es la que utiliza el mayor número de micrófonos, aun así, el patrón direccional obtenido difiere el obtenido en condiciones de prueba ideales debido a las interferencias mencionadas, sin embargo, la respuesta presenta la forma esperada.

5.3.5 Patrón direccional total del sistema

Para obtener el patrón direccional total del sistema se hizo la prueba con señales senoidales con frecuencia $f = 400 \text{ Hz} + 800 \text{ Hz} + 1600 \text{ Hz} + 3200 \text{ Hz}$ como entrada del sistema. El resultado se muestra en la figura 5.12.

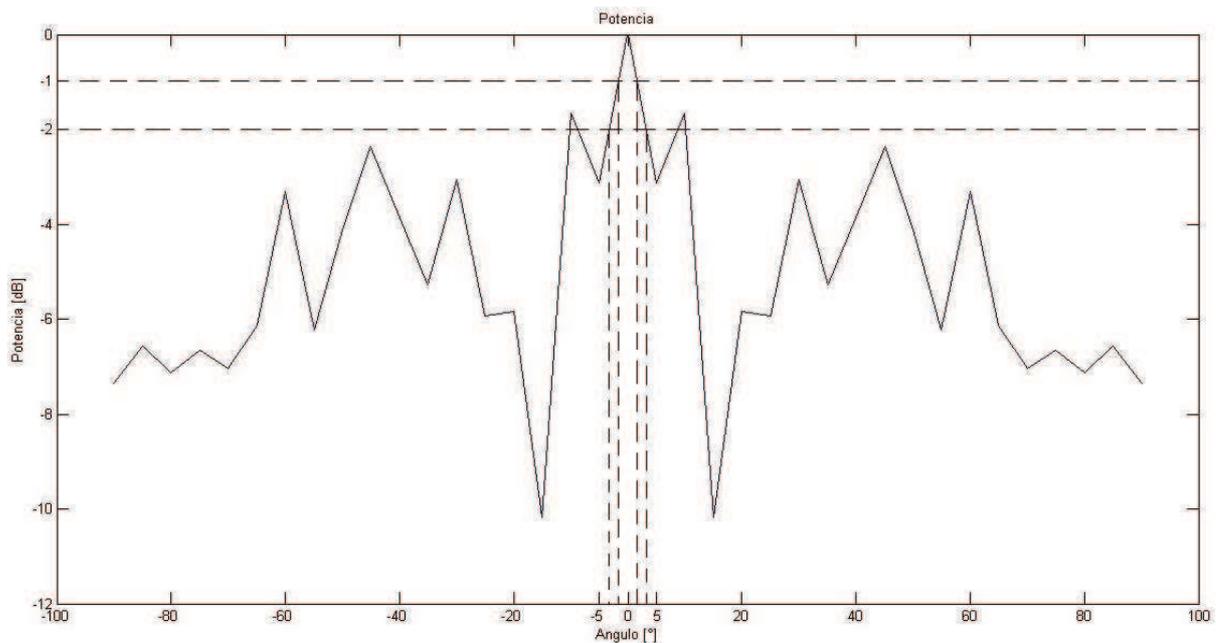


Figura 5.12 Patrón direccional total del sistema en el DSP con señales reales y GSC.

En la figura 5.12 se observa que el patrón direccional del sistema para la banda de frecuencias total tiene su máxima potencia cuando el ángulo de incidencia de las señales acústicas es igual al ángulo al que está dirigido el algoritmo, es decir, 0° y se atenúan las señales que provienen de ángulos diferentes a éste. Para éste patrón direccional un umbral de ganancia de -2dB define el intervalo angular o lóbulo principal que es el límite a partir del cual las señales acústicas se atenúan siendo de aproximadamente $8^\circ (\pm 4^\circ)$ centrado en 0° , en algunas direcciones el sistema atenúa las señales a más de 6dB y en algunos casos hasta 10dB .

El patrón direccional total difiere del obtenido de las pruebas en condiciones ideales debido a las interferencias mencionadas, muestra también un comportamiento mejor que las bandas 1 y 2, pero inferior a las bandas 3 y 4, debido a que este patrón es la combinación de las repuestas individuales de cada banda para señales de tonos acústicos con frecuencias que están en los intervalos de cada una de las bandas.

Para todos los patrones obtenidos se define el umbral a una cierta ganancia en donde se observa que solo las señales provenientes de la dirección angular de interés y direcciones muy cercanas están presentes, sin embargo, si se observan las demás direcciones hay algunas en donde las atenuaciones son mayores que el umbral, indicando que estas direcciones son atenuadas a un nivel muy bajo, y existen otras direcciones en las que el nivel de ganancia de las señales es muy cercano al máximo del patrón direccional, estos picos o lóbulos representan las interferencias presentes en las pruebas y en específico los rebotes de las señales en las paredes del laboratorio

que arriban al arreglo de micrófonos provenientes de direcciones angulares diferentes a la de incidencia de las señales acústicas producidas por la bocina que las genera. Si la amplitud de la señal de rebote es similar a la original y el sistema está dirigido a esa dirección en específico durante la prueba, deja pasar esta señal en su totalidad.

5.4 Pruebas con Voz

Una vez obtenidos los patrones direccionales con señales senoidales, se realizó la prueba del sistema utilizando una señal de voz para ver su comportamiento con una señal con alto contenido frecuencial y comprobar que el sistema filtra espacialmente este tipo de señales acústicas. Se grabó una señal de voz para hacer las pruebas y se repitió el procedimiento descrito para las pruebas con tonos acústicos reales, haciendo la prueba para el sistema en su totalidad y ver su comportamiento. El resultado se muestra en la figura 5.13.

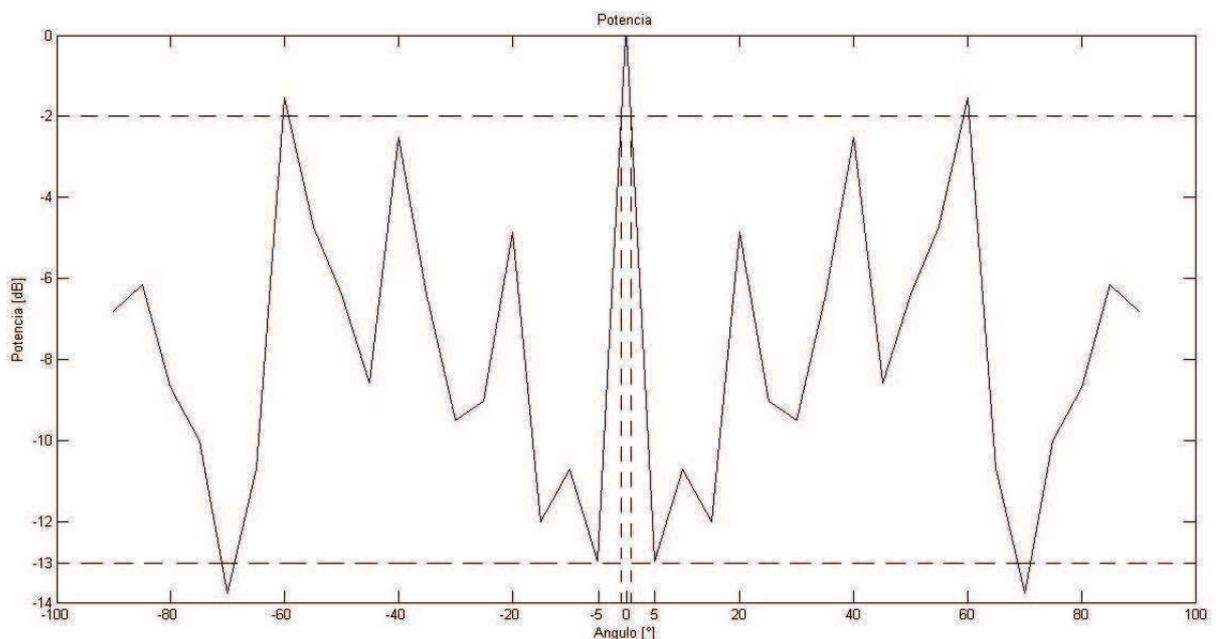


Figura 5.13 Patrón direccional total del sistema en el DSP con voz y GSC.

En la figura 5.13 se observa que la salida del sistema para la banda total tiene su máxima potencia cuando el ángulo de incidencia de las señales acústicas es igual al ángulo al que está dirigido el algoritmo, es decir 0° , y atenúa las señales que provienen de ángulos diferentes a éste. De acuerdo a la forma del patrón direccional un umbral de ganancia de -2dB define el intervalo angular dentro del cual las señales acústicas se conservan o lóbulo principal siendo de aproximadamente 2° ($\pm 4^\circ$) centrado en 0° , se observan atenuaciones que llegan a más de 13dB en algunas direcciones angulares.

La figura 5.14 muestra otra prueba con voz real.

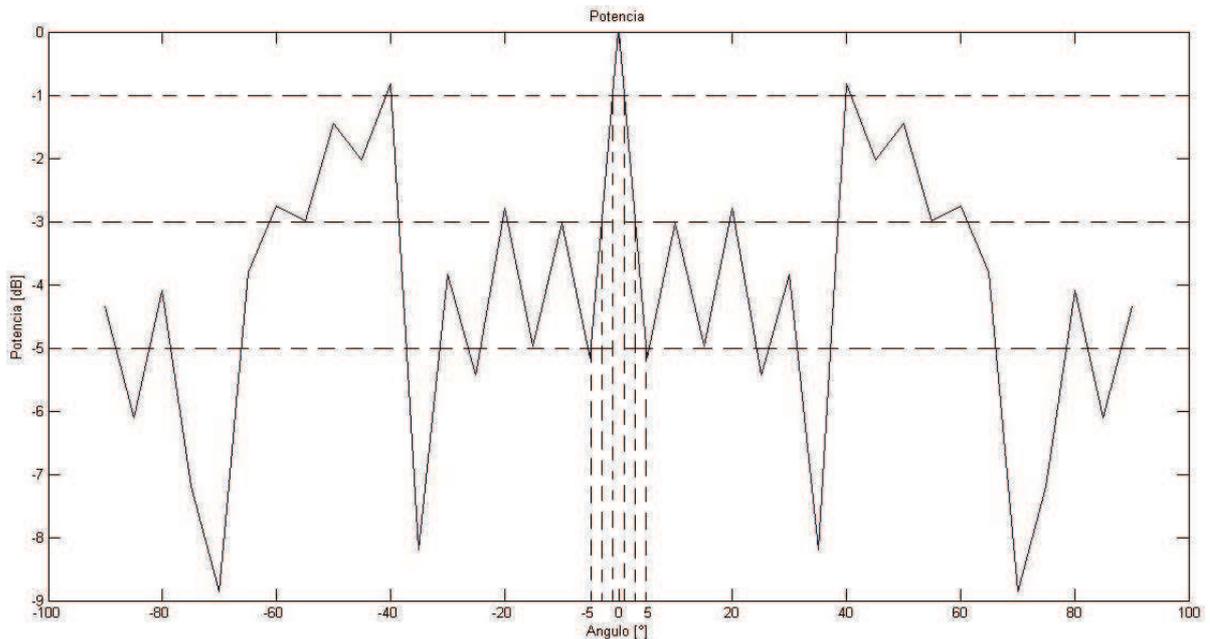


Figura 5.14 Patrón direccional total del sistema en el DSP con voz y GSC.

En la figura 5.14 se observa una vez más que la salida del sistema para la banda total probado con voz tiene su máxima potencia cuando el ángulo de incidencia de las señales acústicas es igual al 0° , y atenúa las señales que provienen de ángulos diferentes a éste. Se tiene una zona en el centro del patrón direccional en donde se aprecia un comportamiento muy bueno, con atenuaciones desde 3dB hasta 5dB desde -40° a 40° , fuera de este intervalo se presentan los lóbulos laterales muy altos. Tomando un umbral de 3dB el lóbulo principal es aproximadamente de $6^\circ (\pm 3)$

Existen diversos trabajos que demuestran prácticamente que los sistemas con formador de haz funcionan dentro del intervalo de -60° a 60° [16]

Para observar el comportamiento del sistema con señales acústicas que provienen de direcciones angulares diferentes de 0° se hicieron pruebas girando el arreglo de micrófonos para que las señales acústicas de las bocinas arribaran con un ángulo diferente, se hicieron las pruebas con voz real para la totalidad del sistema. Debido a que el patrón direccional es simétrico cuando las señales arriban a 0° se hizo un barrido de -90 a 90° para obtener el patrón direccional. La figura 5.15 muestra el resultado de la prueba con un ángulo de -40° .

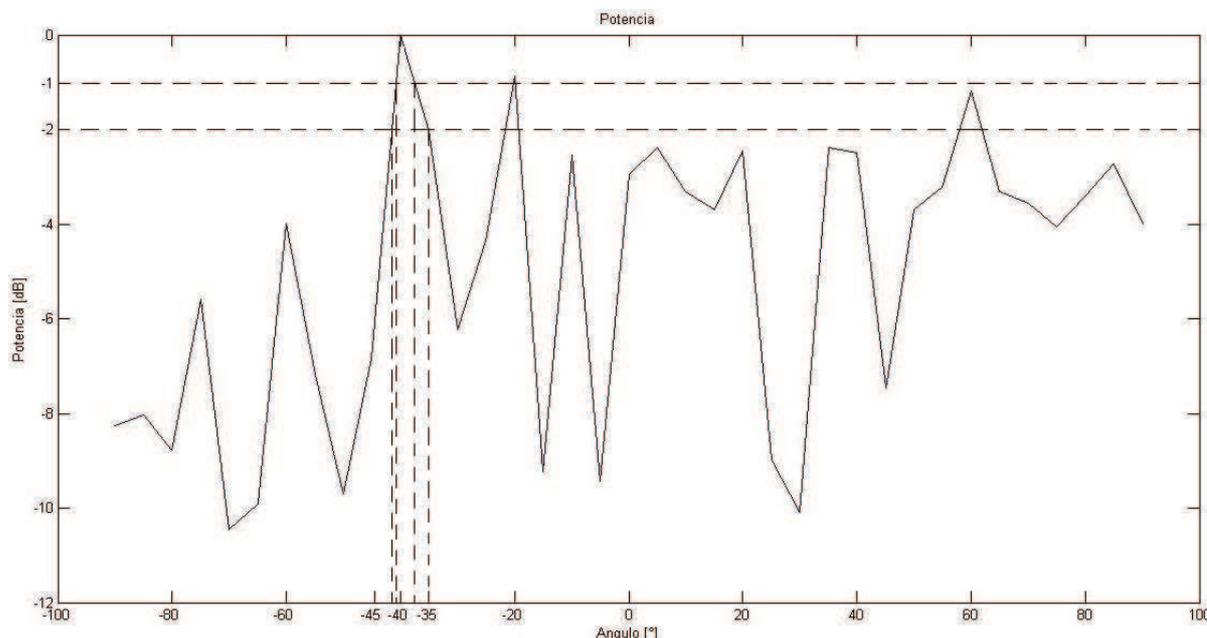


Figura 5.15 Patrón direccional total del sistema en el DSP con voz, GSC y ángulo -40° .

En la figura 5.15 se observa que la salida del sistema para la banda total tiene su máxima potencia cuando el ángulo de incidencia de las señales acústicas es igual al ángulo al que está dirigido el algoritmo, es decir -40° , y atenúa las señales que provienen de ángulos diferentes a éste. De acuerdo a la forma del patrón direccional un umbral de ganancia de -1dB define el intervalo angular de 3° (-41° a -38°) dentro del cual las señales acústicas se conservan y fuera del cual las señales acústicas se atenúan. Si se usa un umbral de -2dB este intervalo angular es aproximadamente de 7° (-42° a -35°), sin embargo existen ángulos fuera de este intervalo en donde las señales no se atenúan debajo del umbral de ganancia establecido y existen ciertas direcciones en donde la atenuación es de hasta 10 dB .

5.5 Desempeño del Sistema

Una evaluación esencial del sistema es la medida de su desempeño, es decir, el número de instrucciones que ejecuta o los ciclos que le toma realizar todo el procesamiento de los datos y por ende que le toma entregar un resultado, el sistema implementado realiza todas las operaciones sobre los datos dentro del tiempo de interrupción que es de $65\mu\text{seg}$ como se mencionó en el capítulo 4, sin embargo, no ejecuta el mismo número de instrucciones en cada interrupción debido a que las bandas se procesan dependiendo de los índices de submuestreo que tiene el algoritmo, por lo que es necesario medir el desempeño del sistema en cada caso posible de proceso.

La frecuencia del CPU del DSP es de 150MHz , lo que representa 6.667nseg por cada ciclo de instrucción del programa y si se calcula el número de ciclos por interrupción se tiene

$$65\mu\text{seg} \times 150\text{MHz} = 9750 \text{ (ciclos/ interrupción)}$$

que son el número total de ciclos que tiene el sistema para entrar a la interrupción, hacer el procesamiento de los datos y regresar al ciclo infinito en espera de la siguiente interrupción.

Se habilitó el reloj de la utilería *profile* en el CCS, cuando se ejecuta el programa para cuantificar el número de ciclos que le toma al sistema realizar las operaciones programadas y también cuantificar cuantos ciclos quedan sin utilizar antes de la siguiente interrupción y así mismo verificar que cada interrupción conste de 9750 ciclos.

Se corrió el programa hasta la entrada a la primera interrupción, para cuantificar cuantos ciclos le toma al sistema configurar las variables y periféricos y estar listo para iniciar el procesamiento de los datos y se obtuvo una cuenta de 799,913 ciclos equivalentes a 5.332mseg de configuración, a partir de este tiempo el sistema ejecuta 9750 ciclos por interrupción.

De acuerdo a los índices de submuestreo existen cuatro posibles combinaciones de bandas a procesar en las interrupciones, para cada combinación se cuantificó el número de ciclos de proceso y el número de ciclos libres antes de la siguiente interrupción y se calculó el equivalente en segundos para cada caso, la siguiente tabla muestra los resultados.

Caso	Bandas procesadas	Ciclos de proceso y regreso	Tiempo de proceso y regreso	Ciclos de espera	Tiempo de espera	Porcentaje de uso de tiempo
1	4	5047	33.647 μ seg	4703	31.353 μ seg	51.76 %
2	3, 4	6332	42.213 μ seg	3418	22.787 μ seg	64.943 %
3	2, 3, 4	7247	48.313 μ seg	2503	16.687 μ seg	74.328 %
4	1, 2, 3, 4	7963	53.087 μ seg	1787	11.913 μ seg	81.671 %

Tabla 5.1 Ciclos y tiempos del sistema.

En la tabla 5.1 se observan los ciclos cuantificados mediante el reloj del *profile*, la columna de ciclos de proceso y regreso se refiere a la cantidad de ciclos que toma el sistema en procesar los datos y regresar de la interrupción, el tiempo de proceso y regreso es el equivalente en tiempo del valor anterior, los ciclos de espera en los que el sistema no hace nada mas que esperar a que se presente la siguiente interrupción, el tiempo de espera es el equivalente en tiempo del valor anterior y el porcentaje de uso indica que porcentaje del tiempo total de interrupción el sistema se encuentra trabajando.

Se observa que el procesamiento de los datos se realiza totalmente dentro del tiempo disponible de la interrupción y cuando se procesan las cuatro bandas aún queda tiempo disponible para poder realizar más operaciones.

5.6 Recursos Utilizados

Los recursos del DSP utilizados por el sistema en su implementación también indican una medida de desempeño, ya que se puede observar qué porcentaje de la capacidad total del DSP, en cuanto a hardware, se está utilizando para hacer el procesamiento de las señales acústicas.

En cuanto a periféricos el sistema usa 15 entradas del convertidor analógico digital, de un total de 16, por lo que solo queda una entrada que se pudiera utilizar para monitorear alguna otra variable en caso de ser necesaria.

Se utiliza uno de los tres temporizadores de 32 bits con los que cuenta el DSP, quedando dos libres para generar otras interrupciones en base al tiempo.

El DSP cuenta con 88 pines de entrada/salida de propósito general, de los cuales se utilizan solo 12 para desplegar la salida del sistema hacia el convertidor digital analógico, quedando sin utilizar 76.

Los seis canales de DMA, los seis módulos PWM, los dos módulos QEP, los dos módulos eCAN, los tres módulos de interfaz de comunicación serial, el modulo SPI, el modulo I2Cy los dos puertos seriales multicanal (McBSP) no son utilizados.

La memoria dato se utilizó desde la localidad 0x0C000 hasta la localidad 0x0C9F8 para un total de 2552 localidades de memoria dato de un total disponible de 4096 siendo ocupado un 62.30% del total.

La memoria programa se utilizó desde la localidad 0x09000 hasta la localidad 0x099CD para un total de 2509 localidades de memoria programa de un total disponible de 4096 siendo ocupado un 61.25% del total.

En general se esta utilizando aproximadamente un 30% del potencial del DSP, es decir, que quedan recursos disponibles para incrustar este sistema en otras aplicaciones.

5.7 Resumen.

En este capítulo se presentaron las pruebas realizadas al sistema y los resultados obtenidos. Los principales resultados de estas pruebas son los patrones direccionales por cada banda y total del sistema, los cuales muestran la respuesta del sistema a señales acústicas dependiendo de la dirección angular en las que éstas arriban al arreglo de micrófonos.

Se hizo un análisis de los tiempos de ejecución de los procesos del sistema para cada uno de los casos de combinación de bandas a procesar y se presentan los resultados en una tabla.

También se hizo un análisis de los recursos del DSP utilizados para la implementación del sistema; así como los tiempos de ejecución para todo el proceso.

Capítulo 6

Conclusiones



Capítulo 6. Conclusiones.

Se ha diseñado e implementado un sistema para filtrar espacialmente señales acústicas, logrando un funcionamiento y comportamiento adecuados dentro de las expectativas teóricas analizadas para su implementación.

El sistema es capaz de atenuar señales acústicas provenientes de direcciones angulares diferentes a la dirección angular de interés definida en el algoritmo que conforma el software del sistema, el cual ejecuta formadores de haz adaptables en cada banda.

La respuesta del sistema en condiciones ideales es bastante óptima, ya que atenúa señales en direcciones angulares no deseadas a más de 10dB, haciendo que la señal proveniente de la dirección angular de interés sea la de mayor potencia en la salida, descartando así a las demás. Esta respuesta en condiciones ideales tanto del sistema en su totalidad como por banda es alcanzada gracias a los algoritmos adaptables integrados al sistema, mejorando el resultado usando formadores de haz fijos.

La respuesta del sistema a señales reales es aceptable y se tienen atenuaciones de al menos 2dB para señales de tonos acústicos y señales de voz, aun considerando que en el laboratorio donde se realizan las pruebas existen muchas interferencias que afectan el desempeño del sistema.

El sistema funciona con señales de alto contenido frecuencial ya que se implementó el procesamiento de las señales a través de sub bandas, dividiendo el ancho de banda de la voz en cuatro intervalos definidos en octavas, lo cual permite un mejor resultado al procesar las señales con los parámetros adecuados de acuerdo a su contenido frecuencial. Los resultados obtenidos con este sistema son aceptables, empezando por los patrones direccionales teóricos, los cuales muestran que el algoritmo hace de manera correcta la separación de las señales por medio de los filtros, el procesamiento de los datos a través de un algoritmo adaptable para cada banda y por último la integración de los resultados parciales en un solo resultado total. Esta característica del sistema lo hace más útil para usarse en señales como la voz ya que muchos de los algoritmos presentados en artículos científicos solo se diseñan y prueban para señales de una sola frecuencia o no son implementados físicamente.

Los tiempos de ejecución que se midieron son aceptables, considerando que el sistema puede ser requerido para funcionar en tiempo real, donde como es sabido, siempre hay un retraso entre la señal de entrada y la de salida; sin embargo, si este retraso es muy pequeño es indetectable para los sentidos humanos y se puede considerar como funcionamiento en tiempo real. Los tiempos de ejecución en este caso lo permiten.

El DSP utilizado no es de los más poderosos en capacidad de procesamiento y se eligió con base al número de canales del convertidor analógico digital disponibles, ya que se utilizan 15 micrófonos, y la mayoría de los DSP's de alto desempeño no tienen ADC. Con un DSP de mayor capacidad de cálculo se pueden reducir los tiempos obtenidos.

La memoria total del DSP es de 4GB, la memoria dato y programa utilizadas están alrededor del 65% de la disponible inicialmente, lo cual permitiría programar más procesos en el DSP; sin embargo, la limitación del tiempo de la interrupción debe tomarse en cuenta para ello.

Capítulo 6. Conclusiones.

En cuanto al trabajo futuro se puede mejorar el funcionamiento del sistema primero haciendo las pruebas en un ambiente libre de interferencias, y también utilizando un DSP de mayor capacidad de procesamiento, para lo cual se necesitaría un ADC externo con número de canales y mayor número de bits.

Una manera de mejorar las mediciones sería hacer las pruebas en una cámara anecoica en la cual no se presentan rebotes de ningún tipo en ninguna de las superficies que las conforman y generaría un patrón direccional más óptimo y cercano al generado por condiciones ideales.

Las aplicaciones de este sistema pueden ser en el ámbito de la grabación de audio, permitiendo mejorar la calidad de la grabación eliminando ruido o interferencias no deseadas presentes en donde se esté grabando el audio de interés; su uso en auditorios para conferencias a fin de captar mejor la voz de las personas participantes y discriminar el ruido ambiental y el generado por las personas que atienden la conferencia; e incluso podría usarse con algunas modificaciones y mejoras para captar señales lejanas muy puntuales, crear una especie de sonar, detectar la dirección de arribo de una señal acústica (DOA) y para direccionar robots por medio de la voz.

Bibliografía



Bibliografía.

- [1] Wauters, P. *Adaptive Speech Beamforming Using the TMS320C40 Multi-DSP*, EFRIE, France 1995.
- [2] Chen, J. C., Hudson, R. E. & Yao, K., *Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field*, IEEE Trans. Signal Processing, vol. 50, no. 8, pp. 1843–1854, 2002.
- [3] Chen, J. C., Hudson, R. E. & Yao, K., *Acoustic Source Localization and Beamforming: Theory and Practice*, EURASIP Journal on Applied Signal Processing 2003:4, 359–370, 2003.
- [4] Blomberg, A., Austeng, A., Hansen, R. & Holm, S. *Minimum Variance Adaptive Beamforming Applied to a Circular Sonar Array*, Oslo Noruega, 2003.
- [5] Xu, W., Liu, T. C. & Schmidt, H., *Beamforming based on spatial-wavelet decomposition*, Ocean Acoustic Group, Massachusetts Institute of Technology, Cambridge MA, 2003.
- [6] Zarb-Adami, K., Faulkner, A., Bij de Vaate, J. G., Kant, G. W. & Picard, P., *Beamforming Techniques for Large-N Aperture Arrays*, arXiv:1008.4047v1 24 Aug 2010.
- [7] Hansen, C. H., *Fundamentals of Acoustics*, Department of Mechanical Engineering, University of Adelaide, Australia.
- [8] Hueter, T. & Bolt, R. *Sonics: Techniques for the use of sound and ultrasound in engineering and science*. Jhon Wileyand Sons Inc. New York 1995.
- [9] McCowan, I. *Robust Speech Recognition Using Microphone Arrays, PhD Thesis*, Queensland University of Technology, Australia 2001.
- [10] Monzingo, R. *Introduction to Adaptive Arrays*, Wiley – Interscience, Fullerton, California 1980.
- [11] Merry R. J. E., *Wavelet Theory and Applications, A Literature Study*. Eindhoven University of Technology, Department of Mechanical Engineering, Control Systems Technology Group. Eindhoven, June 7, 2005
- [12] Doclo, S. *Multi-Microphone noise reduction and dereverberation techniques for speech applications*. Katholieke Universiteit Leuven Faculteit Toegepaste Wetenschappen, Departement Elektrotechniek. Belgica 2003.
- [13] Silva, A. *Detector de ángulo de arribo de una señal en un arreglo de micrófonos*. Facultad de Ingeniería UNAM, Tesis de licenciatura. México 2010.
- [14] Proakis J.G. *Digital Signal Processing, Principles, Algorithms and Applications*. Prentice Hall. Tercera Edición. E.E.U.U. 1996.

Bibliografía.

- [15] Escobar, S. L. *Conceptos básicos de procesamiento digital de señales*. F.I. UNAM 2009.
- [16] Escobar, S. L. y Silva, A. Sistema de detección de ángulo de arribo de señales acústicas. F.I. UNAM, CINDET 2010. México.
- [17] Vázquez, S. *Reducción de ruido en señales de voz mediante un formador de haz*. Facultad de Ingeniería UNAM, Tesis de licenciatura. México 2012.

Anexos



Glosario de términos

ADC.

Analog – Digital Converter.

AMNOR.

Adaptive Microphone-array System for Noise Reduction.

Algoritmo.

Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Campo cercano.

Región del espacio a la cual la fuente de ondas emite un frente de onda esférico.

Campo lejano.

Región del espacio a la cual la fuente de ondas esta lo suficientemente alejada para considerar que el frente de onda esférico tiende a ser plano.

DAC.

Digital – Analog Converter.

DOA.

Dirección de arribo de una seña acústica.

DSP.

Digital Signal Processor.

Fasor.

Representación en el plano complejo de una señal senoidal.

Filtro digital.

Sistema Lineal Invariable en el Tiempo que produce una alteración selectiva de las componentes frecuenciales de la señal de entrada.

Frecuencia.

Magnitud que mide el número de repeticiones por unidad de tiempo de cualquier fenómeno o suceso periódico.

FT.

Fourier Transform.

GPIO.

General Purpose Input/Output.

GSC.

Generalised Sidelobe Canceler.

LCMV.

Linear Constrained Minimum Variance.

LMS.

Least Mean Square.

MAC.

Multiply and Accumulate.

Micrófono.

Elemento capaz de captar ondas sonoras convirtiendo la potencia acústica en eléctrica de similares características ondulatorias.

Qi.

Formato de punto entero.

Ruido.

Perturbación eléctrica que interfiere en la transmisión o el procesamiento de la señales.

Sensor.

Dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica para cuantificar y manipular.

Señal.

Variación de una corriente eléctrica u otra magnitud física que se utiliza para transmitir información.

Sistema.

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia.

SLITD.

Sistema Lineal Invariante en el Tiempo Discreto.

STFT.

Short Time Fourier Transform.

USB.

Universal Serial Bus.

Viscosidad.

Propiedad de un fluido que tiende a oponerse a su flujo cuando se le aplica una fuerza.

Tarjeta DSP

El DSP TMS320F28335 es miembro de la generación TMS320C28x™/ Delfino™ DSC/MCU, altamente integrado, con soluciones de alto desempeño para aplicaciones demandantes de control, instrumentación y procesamiento de señales.

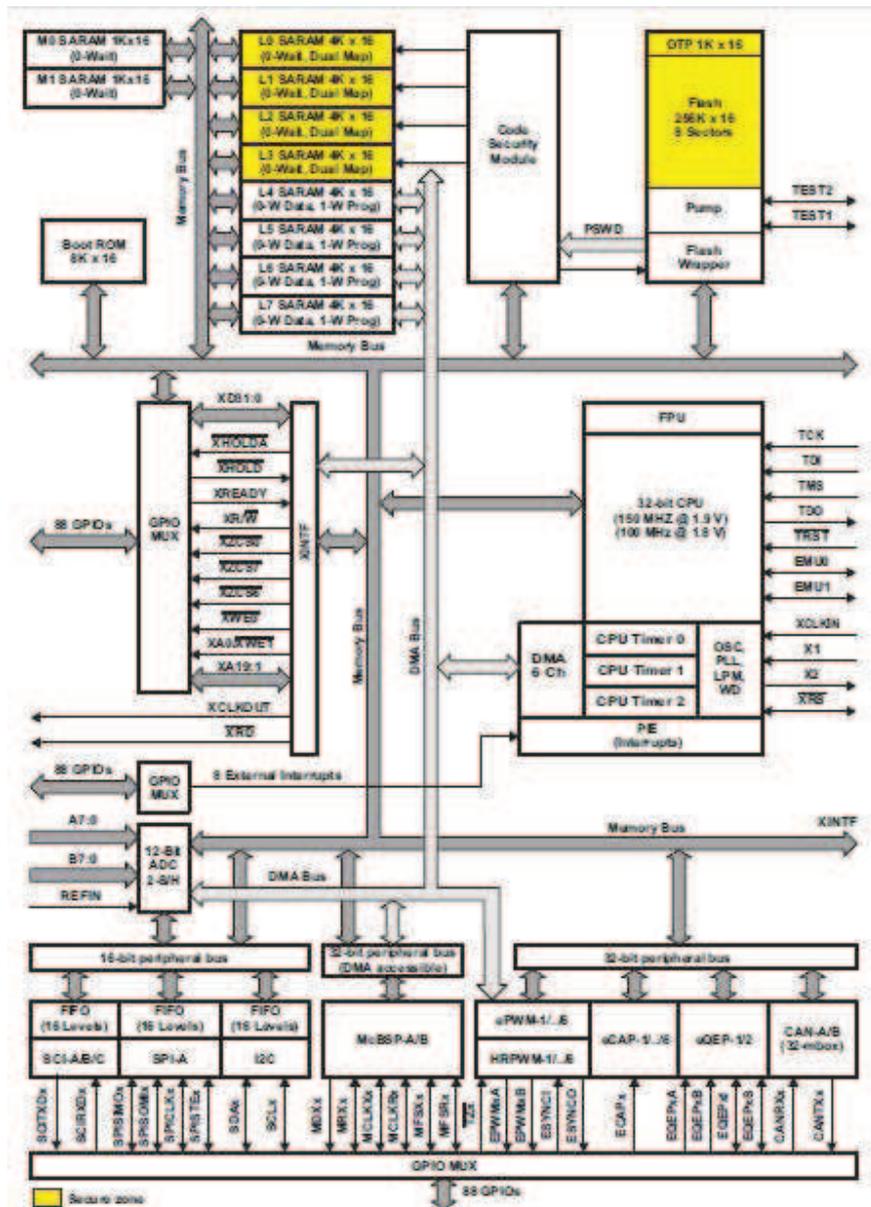


Figura A.1 Diagrama de bloques funcional del TMS320F28335.

Código Principal

Programa principal:

```
#include "DSP28x_Project.h"

interrupt void cpu_timer0_isr(void);

//Buffers de datos entrantes
int Buff1[51];
int Buff2[51];
int Buff3[51];
int Buff4[51];
int Buff5[51];
int Buff6[51];
int Buff7[51];
int Buff8[51];
int Buff9[51];
int Buff10[51];
int Buff11[51];
int Buff12[51];
int Buff13[51];
int Buff14[51];
int Buff15[51];

//Índices de submuestreo
int inds1;
int inds2;
int inds3;

//Buffer de señal filtrada
//Banda 1
int fil11[6];
int fil12[6];
int fil13[6];
int fil14[6];

//Banda 2
int fil21[7];
int fil22[7];
int fil23[7];
int fil24[7];
int fil25[7];

//Banda 3
int fil31[10];
int fil32[10];
int fil33[10];
int fil34[10];
int fil35[10];
int fil36[10];
int fil37[10];

//Banda 4
int fil41[14];
int fil42[14];
int fil43[14];
int fil44[14];
int fil45[14];
int fil46[14];
```

Anexos.

```
int fil47[14];
int fil48[14];
int fil49[14];

//Buffers de formador de haz
int BFb1[51];
int BFb2[51];
int BFb3[51];
int BFb4[51];

//Salidas parciales
int BInt1;
int BInt2;
int BInt3;
int BInt4;

//Salida Final
int sal[512];

//Buffers para matriz de bloqueo GSC
//Banda 1
int B1[3] = {0, 0, 0};
//Banda 2
int B2[4] = {0, 0, 0, 0};
//Banda 3
int B3[6] = {0, 0, 0, 0, 0, 0};
//Banda 4
int B4[8] = {0, 0, 0, 0, 0, 0, 0, 0};

//Coeficientes A para GSC
//Banda 1
int A1[3] = {0, 0, 0};
//Banda 2
int A2[4] = {0, 0, 0, 0};
//Banda 3
int A3[6] = {0, 0, 0, 0, 0, 0};
//Banda 4
int A4[8] = {0, 0, 0, 0, 0, 0, 0, 0};

//Variables aux para matriz de bloqueo GSC
//Banda 1
int x11 = 0;
int x12 = 0;
int x13 = 0;
int x14 = 0;
//Banda 2
int x21 = 0;
int x22 = 0;
int x23 = 0;
int x24 = 0;
int x25 = 0;
//Banda 3
int x31 = 0;
int x32 = 0;
int x33 = 0;
int x34 = 0;
int x35 = 0;
int x36 = 0;
int x37 = 0;
//Banda 4
int x41 = 0;
int x42 = 0;
```

Anexos.

```
int x43 = 0;
int x44 = 0;
int x45 = 0;
int x46 = 0;
int x47 = 0;
int x48 = 0;
int x49 = 0;

//Filtros Q15 (coefs muy pequeños)
const int h1[51] = {-51, -71, -101, -143, -200, -271, -354, -443, -529, -602, -650, -
661, -621, -524, -362, -136, 150, 487, 859, 1248, 1631, 1985, 2288, 2520, 2666, 2716,
2666, 2520, 2288, 1985, 1631, 1248, 859, 487, 150, -136, -362, -524, -621, -661, -650,
-602, -529, -443, -354, -271, -200, -143, -101, -71, -51};
const int h2[51] = {24, 5, -14, -29, -37, -29, 0, 47, 97, 123, 88, -43, -290, -645, -
1064, -1470, -1762, -1838, -1619, -1076, -243, 780, 1841, 2769, 3401, 3626, 3401, 2769,
1841, 780, -243, -1076, -1619, -1838, -1762, -1470, -1064, -645, -290, -43, 88, 123,
97, 47, 0, -29, -37, -29, -14, 5, 24};
const int h3[51] = {-41, -71, -73, -36, 20, 45, -0, -73, -54, 150, 470, 662, 507, 60, -
308, -221, 265, 532, -150, -1881, -3728, -4179, -2255, 1545, 5308, 6875, 5308, 1545, -
2255, -4179, -3728, -1881, -150, 532, 265, -221, -308, 60, 507, 662, 470, 150, -54, -
73, -0, 45, 20, -36, -73, -71, -41};
const int h4[51] = {4, -11, 22, -66, -116, 26, -0, -43, 307, 278, -144, 105, -50, -970,
-399, 364, -436, 688, 2436, 185, -591, 1278, -4166, -8748, 3128, 13771, 3128, -8748, -
4166, 1278, -591, 185, 2436, 688, -436, 364, -399, -970, -50, 105, -144, 278, 307, -43,
-0, 26, -116, -66, 22, -11, 4};

//Ventanas Q12
const int v1[4] = {193, 1855, 1855, 193};
const int v2[5] = {146, 987, 1829, 987, 146};
const int v3[7] = {99, 382, 950, 1234, 950, 382, 99};
const int v4[9] = {74, 200, 503, 805, 931, 805, 503, 200, 74};

//Dirección angular
const int ang = 90;
int ind;

//Offset de los micrófonos
long int soff = 0;
int off = 0;

//Tamaños Incrementos GSC
const int miu1 = 324;
const int miu2 = 164;
const int miu3 = 82;
const int miu4 = 41;

//Vectores de dirección
int dir1[4] = {0, 0, 0, 0};
int dir2[5] = {0, 0, 0, 0, 0};
int dir3[7] = {0, 0, 0, 0, 0, 0, 0};
int dir4[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0};

//Matriz Dirección
int tau[37][9] =
    {{13, 11, 9, 8, 6, 5, 3, 2, 0},
     {12, 11, 9, 8, 6, 5, 3, 2, 0},
     {12, 11, 9, 8, 6, 5, 3, 2, 0},
     {12, 11, 9, 8, 6, 5, 3, 2, 0},
     {12, 10, 9, 7, 6, 4, 3, 1, 0},
     {11, 10, 9, 7, 6, 4, 3, 1, 0},
     {11, 9, 8, 7, 5, 4, 3, 1, 0},
     {10, 9, 8, 6, 5, 4, 3, 1, 0},
     {10, 8, 7, 6, 5, 4, 2, 1, 0},
```

Anexos.

```
{ 9, 8, 7, 6, 4, 3, 2, 1, 0},
{ 8, 7, 6, 5, 4, 3, 2, 1, 0},
{ 7, 6, 5, 4, 4, 3, 2, 1, 0},
{ 6, 5, 5, 4, 3, 2, 2, 1, 0},
{ 5, 5, 4, 3, 3, 2, 1, 1, 0},
{ 4, 4, 3, 3, 2, 2, 1, 1, 0},
{ 3, 3, 2, 2, 2, 1, 1, 0, 0},
{ 2, 2, 2, 1, 1, 1, 1, 0, 0},
{ 1, 1, 1, 1, 1, 0, 0, 0, 0},
{ 0, 0, 0, 0, 0, 0, 0, 0, 0},
{ 0, 0, 0, 0, 1, 1, 1, 1, 1},
{ 0, 0, 1, 1, 1, 1, 2, 2, 2},
{ 0, 0, 1, 1, 2, 2, 2, 3, 3},
{ 0, 1, 1, 2, 2, 3, 3, 4, 4},
{ 0, 1, 1, 2, 3, 3, 4, 5, 5},
{ 0, 1, 2, 2, 3, 4, 5, 5, 6},
{ 0, 1, 2, 3, 4, 4, 5, 6, 7},
{ 0, 1, 2, 3, 4, 5, 6, 7, 8},
{ 0, 1, 2, 3, 4, 6, 7, 8, 9},
{ 0, 1, 2, 4, 5, 6, 7, 8, 10},
{ 0, 1, 3, 4, 5, 6, 8, 9, 10},
{ 0, 1, 3, 4, 5, 7, 8, 9, 11},
{ 0, 1, 3, 4, 6, 7, 9, 10, 11},
{ 0, 1, 3, 4, 6, 7, 9, 10, 12},
{ 0, 2, 3, 5, 6, 8, 9, 11, 12},
{ 0, 2, 3, 5, 6, 8, 9, 11, 12},
{ 0, 2, 3, 5, 6, 8, 9, 11, 12},
{ 0, 2, 3, 5, 6, 8, 9, 11, 13}};

// Parámetros del ADC
#if (CPU_FRQ_150MHZ) //Por default - 150 MHz SYSCLKOUT
#define ADC_MODCLK 0x3 //HSPCLK = SYSCLKOUT/2*ADC_MODCLK2=150/(2*3)=25.0MHz
#endif
#if (CPU_FRQ_100MHZ)
#define ADC_MODCLK 0x2 // HSPCLK=SYSCLKOUT/2*ADC_MODCLK2 =100/(2*2)=25.0MHz
#endif
#define ADC_CKPS 0x0 //Modulo reloj de ADC=HSPCLK/2*ADC_CKPS=25.0MHz/(1*2)=12.5MHz
#define ADC_SHCLK 0x1 //Ancho de S/H en periodos de ADC = 1 reloj de ADC
#define AVG 1500 //Limite promedio de muestra
#define BUF_SIZE 51 //Tamaño de buffer de muestra

Uint16 i;

//Variables auxiliar para multiplicaciones
long int aux = 0;
int aux2 = 0; //GSC

//Variable para GSC
long int ya = 0;

//Variable de índice de retraso
int ret;

//Variables de potencias
long spot1 = 0;
long spot2 = 0;
long spot3 = 0;
long spot4 = 0;
long spotT = 0;

int pot1 = 0;
int pot2 = 0;
```

Anexos.

```
int pot3 = 0;
int pot4 = 0;
int potT = 0;

//Rutinas externas
extern void fill (void);           //Todos los Buffers los inicia en 0
extern void Ffir1 (void);         //Filtro fir 1
extern void Ffir2 (void);         //Filtro fir 2
extern void Ffir3 (void);         //Filtro fir 3
extern void Ffir4 (void);         //Filtro fir 4
extern void movdat(void);         //Mueve los datos de entrada en el Buffer
extern void movfil(void);        //Mueve los datos filtrados en el Buffer
extern void movbf (void);        //Mueve los datos del formador fijo en el
extern void FfirInt1(void);      //Filtro de interpolación fir 1
extern void FfirInt2(void);      //Filtro de interpolación fir 2
extern void FfirInt3(void);      //Filtro de interpolación fir 3
extern void FfirInt4(void);      //Filtro de interpolación fir 4

void main(void)
{
    // PLL, WatchDog, activa relojes de periféricos
    InitSysCtrl();

    //Configuración de reloj de periféricos
    EALLOW; //Permite escribir en registros protegidos
    SysCtrlRegs.HISPCP.all = ADC_MODCLK; // HSPCLK = SYSCLKOUT/ADC_MODCLK
    EDIS; //Protege registros para escritura

    DINT; //desactiva interrupciones de CPU

    // Inicializa los registros de control del PIE a su estado por de fault.
    InitPieCtrl();

    // Desactiva interrupciones de CPU y limpia todas las banderas:
    IER = 0x0000; //Registro de activación de interrupciones
    IFR = 0x0000; //Registro de banderas de interrupción

    // Inicializa la tabla de vectores del PIE
    // Rutinas de servicio (ISR).
    InitPieVectTable();

    EALLOW; //Permite escribir en registros
    PieVectTable.TINT0 = &cpu_timer0_isr;
    EDIS; //No permite escribir en registros

    //Inicializa Timers
    InitCpuTimers();

    #if (CPU_FRQ_150MHZ)
    // Configura CPU-Timer 0 para interrumpir cada 65 microsegundos:
    ConfigCpuTimer(&CpuTimer0, 150, 65);
    #endif
    #if (CPU_FRQ_100MHZ)
    // Configura CPU-Timer 0 para interrumpir cada 65 microsegundos:
    ConfigCpuTimer(&CpuTimer0, 100, 65);
    #endif

    // Activa Timer e Interrupción
    CpuTimer0Regs.TCR.all = 0x4001;

    // ADC
    InitAdc(); // Inicializa el ADC
}
```

Anexos.

```
    AdcRegs.ADCTRL1.bit.ACQ_PS = ADC_SHCLK;    //Tamaño de ventana de adquisición
ADC_SHCLK = 0x1 (ADC_SHCLK+1)*ADCLK
    AdcRegs.ADCTRL3.bit.ADCCLKPS = ADC_CKPS; // Divisor de reloj ADC_CKPS = 0x0 HSPCLK
Pasa igual
    AdcRegs.ADCTRL1.bit.SEQ_CASC = 1;          // 1 modo cascada
    AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0;    // Secuencia de conversión ADCINA0 (después
poner los 16 canales)
    AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x1;
    AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x2;
    AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x3;
    AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x4;
    AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 0x5;
    AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 0x6;
    AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 0x7;
    AdcRegs.ADCCHSELSEQ3.bit.CONV08 = 0x8;
    AdcRegs.ADCCHSELSEQ3.bit.CONV09 = 0x9;
    AdcRegs.ADCCHSELSEQ3.bit.CONV10 = 0xA;
    AdcRegs.ADCCHSELSEQ3.bit.CONV11 = 0xB;
    AdcRegs.ADCCHSELSEQ4.bit.CONV12 = 0xC;
    AdcRegs.ADCCHSELSEQ4.bit.CONV13 = 0xD;
    AdcRegs.ADCCHSELSEQ4.bit.CONV14 = 0xE;
    AdcRegs.ADCCHSELSEQ4.bit.CONV15 = 0xF;

    AdcRegs.ADCTRL1.bit.CONT_RUN = 1;          //Modo continuo
    AdcRegs.ADCMAXCONV.all = 0xF; //Configura cada secuencia con 15 SOCs

//Configuración de salidas para desplegar la señal
//Primero como entradas/salidas de uso general
EALLOW;
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0; //LSB
GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO4 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO6 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO7 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO8 = 0;
GpioCtrlRegs.GPAMUX1.bit.GPIO9 = 0; //MSB
//Configuradas como salidas
GpioCtrlRegs.GPADIR.bit.GPIO0 = 1; //LSB
GpioCtrlRegs.GPADIR.bit.GPIO1 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO2 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO3 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO4 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO5 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO6 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO7 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO8 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO9 = 1; //MSB
EDIS;

//Selección de retrasos
ind = ((ang+90)/5);
if (ang>=0)
{
    dir1[0] = tau[ind][0];
    dir1[1] = tau[ind][1];
    dir1[2] = tau[ind][2];
    dir1[3] = tau[ind][3];

    dir2[0] = tau[ind][0];
```

Anexos.

```
        dir2[1] = tau[ind][1];
        dir2[2] = tau[ind][2];
        dir2[3] = tau[ind][3];
        dir2[4] = tau[ind][4];

        dir3[0] = tau[ind][0];
        dir3[1] = tau[ind][1];
        dir3[2] = tau[ind][2];
        dir3[3] = tau[ind][3];
        dir3[4] = tau[ind][4];
        dir3[5] = tau[ind][5];
        dir3[6] = tau[ind][6];

        dir4[0] = tau[ind][0];
        dir4[1] = tau[ind][1];
        dir4[2] = tau[ind][2];
        dir4[3] = tau[ind][3];
        dir4[4] = tau[ind][4];
        dir4[5] = tau[ind][5];
        dir4[6] = tau[ind][6];
        dir4[7] = tau[ind][7];
        dir4[8] = tau[ind][8];
    }
    else
    {
        dir1[0] = tau[ind][5];
        dir1[1] = tau[ind][6];
        dir1[2] = tau[ind][7];
        dir1[3] = tau[ind][8];

        dir2[0] = tau[ind][4];
        dir2[1] = tau[ind][5];
        dir2[2] = tau[ind][6];
        dir2[3] = tau[ind][7];
        dir2[4] = tau[ind][8];

        dir3[0] = tau[ind][2];
        dir3[1] = tau[ind][3];
        dir3[2] = tau[ind][4];
        dir3[3] = tau[ind][5];
        dir3[4] = tau[ind][6];
        dir3[5] = tau[ind][7];
        dir3[6] = tau[ind][8];

        dir4[0] = tau[ind][0];
        dir4[1] = tau[ind][1];
        dir4[2] = tau[ind][2];
        dir4[3] = tau[ind][3];
        dir4[4] = tau[ind][4];
        dir4[5] = tau[ind][5];
        dir4[6] = tau[ind][6];
        dir4[7] = tau[ind][7];
        dir4[8] = tau[ind][8];
    }

    //Se llenan todos los buffers con 0
    fill();

    // Índices de submuestreo en 0
    inds1 = 0;
    inds2 = 0;
    inds3 = 0;
```

Anexos.

```
// Inicializa SEQ1
AdcRegs.ADCTRL2.all = 0x2000;

i=0;

//Calculo de Offset para restárselo a las señales recibidas
for (i=0; i<512; i++)
{
    soff = soff + (AdcRegs.ADCRESULT8>>4);
}

soff = soff >> 9;
off = (int)soff;

i = 0;

// Activa CPU INT1 conectada a CPU-Timer 0:
IER |= M_INT1;

// Activa TINT0 en el PIE: Grupo 1 interrupción 7
PieCtrlRegs.PIEIER1.bit.INTx7 = 1;

// Activa interrupciones globales
EINT; // Activa interrupción global INTM
ERTM; // Activa interrupción global en tiempo real DBGM

//Loop infinito en espera de interrupción
for(;;)
{
}

} // Fin de Main

// Interrupción del Timer
interrupt void cpu_timer0_isr(void)
{
    CpuTimer0.InterruptCount++; //Contador de interrupciones

    // ADC //
    AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1; // Limpia bandera de interrupción INT_SEQ1

    if (i<512)
    {

        //Recepción de datos y acomodo en los buffers en Q12
        Buff1[50] = ((AdcRegs.ADCRESULT1>>4)) - off;
        Buff2[50] = ((AdcRegs.ADCRESULT2>>4)) - off;
        Buff3[50] = ((AdcRegs.ADCRESULT3>>4)) - off;
        Buff4[50] = ((AdcRegs.ADCRESULT4>>4)) - off;
        Buff5[50] = ((AdcRegs.ADCRESULT5>>4)) - off;
        Buff6[50] = ((AdcRegs.ADCRESULT6>>4)) - off;
        Buff7[50] = ((AdcRegs.ADCRESULT7>>4)) - off;
        Buff8[50] = ((AdcRegs.ADCRESULT8>>4)) - off;
        Buff9[50] = ((AdcRegs.ADCRESULT9>>4)) - off;
        Buff10[50] = ((AdcRegs.ADCRESULT10>>4)) - off;
        Buff11[50] = ((AdcRegs.ADCRESULT11>>4)) - off;
        Buff12[50] = ((AdcRegs.ADCRESULT12>>4)) - off;
        Buff13[50] = ((AdcRegs.ADCRESULT13>>4)) - off;
    }
}
```

Anexos.

```
Buff14[50] = ((AdcRegs.ADCRESULT14>>4)) - off;
Buff15[50] = ((AdcRegs.ADCRESULT15>>4)) - off;

//Filtro y submuestreo
//Banda 1
inds1 = inds1 + 1;
if (inds1 == 8)
{
    Ffir1();
    inds1 = 0;
}

//Banda 2
inds2 = inds2 + 1;
if (inds2 == 4)
{
    Ffir2();
    inds2 = 0;
}

//Banda 3
inds3 = inds3 + 1;
if (inds3 == 2)
{
    Ffir3();
    inds3 = 0;
}

//Banda 4
Ffir4();

//Formador de Haz
//Banda 1
if (inds1 == 0)
{ //Formador de Haz Fijo
    ret = 5 - dir1[0];
    x11 = fil11[ret]; //GSC
    aux = (long) fil11[ret] * (long) v1[0];
    ret = 5 - dir1[1];
    x12 = fil12[ret]; //GSC
    aux = aux + (long) fil12[ret] * (long) v1[1];
    ret = 5 - dir1[2];
    x13 = fil13[ret]; //GSC
    aux = aux + (long) fil13[ret] * (long) v1[2];
    ret = 5 - dir1[3];
    x14 = fil14[ret]; //GSC
    aux = aux + (long) fil14[ret] * (long) v1[3];
    aux = aux >> 12;

    //Matriz de bloqueo 1 GSC
    B1[0] = x11 - x12;
    B1[1] = x12 - x13;
    B1[2] = x13 - x14;

    //Coefs A's Xb*A GSC
    ya = (long) A1[0] * (long) B1[0];
    ya = ya + (long) A1[1] * (long) B1[1];
    ya = ya + (long) A1[2] * (long) B1[2];
    ya = ya >> 12;

    //Salida parcial 1
```

Anexos.

```
//BFb1[50] = (int) aux;

//Salida parcial 1 GSC
    BFb1[50] = (int) aux - (int) ya;

        //LMS 1 GSC
aux = (long) BFb1[50] * (long) B1[0];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu1 * (long) aux2;
aux = aux >> 12;
A1[0] = A1[0] + (int) aux;

aux = (long) BFb1[50] * (long) B1[1];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu1 * (long) aux2;
aux = aux >> 12;
A1[1] = A1[1] + (int) aux;

aux = (long) BFb1[50] * (long) B1[2];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu1 * (long) aux2;
aux = aux >> 12;
A1[2] = A1[2] + (int) aux;

}
else
{
    BFb1[50] = 0;
}

//Banda 2
if (inds2 == 0)
{ //Formador de Haz Fijo
    ret = 6 - dir2[0];
    x21 = fil21[ret]; //GSC
    aux = (long) fil21[ret] * (long) v2[0];
    ret = 6 - dir2[1];
    x22 = fil22[ret]; //GSC
    aux = aux + (long) fil22[ret] * (long) v2[1];
    ret = 6 - dir2[2];
    x23 = fil23[ret]; //GSC
    aux = aux + (long) fil23[ret] * (long) v2[2];
    ret = 6 - dir2[3];
    x24 = fil24[ret]; //GSC
    aux = aux + (long) fil24[ret] * (long) v2[3];
    ret = 6 - dir2[4];
    x25 = fil25[ret]; //GSC
    aux = aux + (long) fil25[ret] * (long) v2[4];
    aux = aux >> 12;

    //Matriz de bloqueo 2 GSC
B2[0] = x21 - x22;
B2[1] = x22 - x23;
B2[2] = x23 - x24;
B2[3] = x24 - x25;

//Coefs A's Xb*A GSC
ya = (long) A2[0] * (long) B2[0];
ya = ya + (long) A2[1] * (long) B2[1];
```

Anexos.

```
ya = ya + (long) A2[2] * (long) B2[2];
ya = ya + (long) A2[3] * (long) B2[3];
ya = ya >> 12;

//Salida parcial 2
    //BFb2[50] = (int) aux;

//Salida parcial 2 GSC
    BFb2[50] = (int) aux - (int) ya;

    //LMS 2 GSC
aux = (long) BFb2[50] * (long) B2[0];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu2 * (long) aux2;
aux = aux >> 12;
A2[0] = A2[0] + (int) aux;

aux = (long) BFb2[50] * (long) B2[1];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu2 * (long) aux2;
aux = aux >> 12;
A2[1] = A2[1] + (int) aux;

aux = (long) BFb2[50] * (long) B2[2];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu2 * (long) aux2;
aux = aux >> 12;
A2[2] = A2[2] + (int) aux;

aux = (long) BFb2[50] * (long) B2[3];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu2 * (long) aux2;
aux = aux >> 12;
A2[3] = A2[3] + (int) aux;

}
else
{
    BFb2[50] = 0;
}

//Banda 3
if (inds3 == 0)
{ //Formador de Haz Fijo
    ret = 9 - dir3[0];
    x31 = fil31[ret]; //GSC
    aux = (long) fil31[ret] * (long) v3[0];
    ret = 9 - dir3[1];
    x32 = fil32[ret]; //GSC
    aux = aux + (long) fil32[ret] * (long) v3[1];
    ret = 9 - dir3[2];
    x33 = fil33[ret]; //GSC
    aux = aux + (long) fil33[ret] * (long) v3[2];
    ret = 9 - dir3[3];
    x34 = fil34[ret]; //GSC
    aux = aux + (long) fil34[ret] * (long) v3[3];
    ret = 9 - dir3[4];
    x35 = fil35[ret]; //GSC
```

Anexos.

```
    aux = aux + (long) fil35[ret] * (long) v3[4];
    ret = 9 - dir3[5];
    x36 = fil36[ret]; //GSC
    aux = aux + (long) fil36[ret] * (long) v3[5];
    ret = 9 - dir3[6];
    x37 = fil37[ret]; //GSC
    aux = aux + (long) fil37[ret] * (long) v3[6];
    aux = aux >> 12;

//Matriz de bloqueo 3 GSC
B3[0] = x31 - x32;
B3[1] = x32 - x33;
B3[2] = x33 - x34;
B3[3] = x34 - x35;
B3[4] = x35 - x36;
B3[5] = x36 - x37;

//Coefs A's Xb*A GSC
ya = (long) A3[0] * (long) B3[0];
ya = ya + (long) A3[1] * (long) B3[1];
ya = ya + (long) A3[2] * (long) B3[2];
ya = ya + (long) A3[3] * (long) B3[3];
ya = ya + (long) A3[4] * (long) B3[4];
ya = ya + (long) A3[5] * (long) B3[5];
ya = ya >> 12;

//Salida parcial 3
//BFb3[50] = (int) aux;

//Salida parcial 3 GSC
BFb3[50] = (int) aux - (int) ya;

//LMS 3 GSC
aux = (long) BFb3[50] * (long) B3[0];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu3 * (long) aux2;
aux = aux >> 12;
A3[0] = A3[0] + (int) aux;

aux = (long) BFb3[50] * (long) B3[1];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu3 * (long) aux2;
aux = aux >> 12;
A3[1] = A3[1] + (int) aux;

aux = (long) BFb3[50] * (long) B3[2];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu3 * (long) aux2;
aux = aux >> 12;
A3[2] = A3[2] + (int) aux;

aux = (long) BFb3[50] * (long) B3[3];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu3 * (long) aux2;
aux = aux >> 12;
A3[3] = A3[3] + (int) aux;

aux = (long) BFb3[50] * (long) B3[4];
```

Anexos.

```
    aux = aux >> 12;
    aux2 = (int) aux;
    aux = (long) miu3 * (long) aux2;
    aux = aux >> 12;
    A3[4] = A3[4] + (int) aux;

    aux = (long) BFb3[50] * (long) B3[5];
    aux = aux >> 12;
    aux2 = (int) aux;
    aux = (long) miu3 * (long) aux2;
    aux = aux >> 12;
    A3[5] = A3[5] + (int) aux;

}
else
{
    BFb3[50] = 0;
}

//Banda 4
//Formador de Haz Fijo
ret = 13 - dir4[0];
x41 = fil41[ret]; //GSC
aux = (long) fil41[ret] * (long) v4[0];
ret = 13 - dir4[1];
x42 = fil42[ret]; //GSC
aux = aux + (long) fil42[ret] * (long) v4[1];
ret = 13 - dir4[2];
x43 = fil43[ret]; //GSC
aux = aux + (long) fil43[ret] * (long) v4[2];
ret = 13 - dir4[3];
x44 = fil44[ret]; //GSC
aux = aux + (long) fil44[ret] * (long) v4[3];
ret = 13 - dir4[4];
x45 = fil45[ret]; //GSC
aux = aux + (long) fil45[ret] * (long) v4[4];
ret = 13 - dir4[5];
x46 = fil46[ret]; //GSC
aux = aux + (long) fil46[ret] * (long) v4[5];
ret = 13 - dir4[6];
x47 = fil47[ret]; //GSC
aux = aux + (long) fil47[ret] * (long) v4[6];
ret = 13 - dir4[7];
x48 = fil48[ret]; //GSC
aux = aux + (long) fil48[ret] * (long) v4[7];
ret = 13 - dir4[8];
x49 = fil49[ret]; //GSC
aux = aux + (long) fil49[ret] * (long) v4[8];
aux = aux >> 12;

//Matriz de bloqueo 4 GSC
B4[0] = x41 - x42;
B4[1] = x42 - x43;
B4[2] = x43 - x44;
B4[3] = x44 - x45;
B4[4] = x45 - x46;
B4[5] = x46 - x47;
B4[6] = x47 - x48;
B4[7] = x48 - x49;

//Coefs A's Xb*A GSC
ya = (long) A4[0] * (long) B4[0];
```

Anexos.

```
ya = ya + (long) A4[1] * (long) B4[1];
ya = ya + (long) A4[2] * (long) B4[2];
ya = ya + (long) A4[3] * (long) B4[3];
ya = ya + (long) A4[4] * (long) B4[4];
ya = ya + (long) A4[5] * (long) B4[5];
ya = ya + (long) A4[6] * (long) B4[6];
ya = ya + (long) A4[7] * (long) B4[7];
ya = ya >> 12;

//Salida parcial 4
//BFb4[50] = (int) aux;

//Salida parcial 4 GSC
BFb4[50] = (int) aux - (int) ya;

//LMS 4 GSC
aux = (long) BFb4[50] * (long) B4[0];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
aux = aux >> 12;
A4[0] = A4[0] + (int) aux;

aux = (long) BFb4[50] * (long) B4[1];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
aux = aux >> 12;
A4[1] = A4[1] + (int) aux;

aux = (long) BFb4[50] * (long) B4[2];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
aux = aux >> 12;
A4[2] = A4[2] + (int) aux;

aux = (long) BFb4[50] * (long) B4[3];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
aux = aux >> 12;
A4[3] = A4[3] + (int) aux;

aux = (long) BFb4[50] * (long) B4[4];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
aux = aux >> 12;
A4[4] = A4[4] + (int) aux;

aux = (long) BFb4[50] * (long) B4[5];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
aux = aux >> 12;
A4[5] = A4[5] + (int) aux;

aux = (long) BFb4[50] * (long) B4[6];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
```

Anexos.

```
aux = aux >> 12;
A4[6] = A4[6] + (int) aux;

aux = (long) BFb4[50] * (long) B4[7];
aux = aux >> 12;
aux2 = (int) aux;
aux = (long) miu4 * (long) aux2;
aux = aux >> 12;
A4[7] = A4[7] + (int) aux;

//Filtros de interpolación
FfirInt1();
FfirInt2();
FfirInt3();
FfirInt4();

//Suma de salidas parciales
sal[i] = BInt1 + BInt2;
sal[i] = sal[i] + BInt3;
sal[i] = sal[i] + BInt4;

//Calculo de potencias
//Banda 1
aux = (long) BInt1 * (long) BInt1;
aux = aux >> 12;
spot1 = spot1 + aux;

//Banda 2
aux = (long) BInt2 * (long) BInt2;
aux = aux >> 12;
spot2 = spot2 + aux;

//Banda 3
aux = (long) BInt3 * (long) BInt3;
aux = aux >> 12;
spot3 = spot3 + aux;

//Banda 4
aux = (long) BInt4 * (long) BInt4;
aux = aux >> 12;
spot4 = spot4 + aux;

//Potencia total
aux = (long) sal[i] * (long) sal[i];
aux = aux >> 12;
spotT = spotT + aux;

//Desplegar salida
cont = sal[255] + off;

GpioDataRegs.GPADAT.all = cont >> 3;

//Movimiento de datos en buffers
movdat();
movfil();
movbf();

    i = i + 1;
}

if (i==512)
```

Anexos.

```
{
    aux = spot1;
    aux = aux >> 9;
    pot1 = (int) aux;

    aux = spot2;
    aux = aux >> 9;
    pot2 = (int) aux;

    aux = spot3;
    aux = aux >> 9;
    pot3 = (int) aux;

    aux = spot4;
    aux = aux >> 9;
    pot4 = (int) aux;

    aux = spotT;
    aux = aux >> 9;
    potT = (int) aux;

    i = i + 1;
}

PieCtrlRegs.PIEACK.all = PIEACK_GROUP1; // Informa interrupción al PIE
}
```

Las variables del programa en C se pasan a rutinas en ensamblador mediante una definición global dentro de éstas, se presenta un filtro FIR en ensamblador a manera de ejemplo.

```
Funcion Filtro fir 1
    .def          _Ffir1
    .sect         "ramfuncs"

    .global      _h1
    .global      _Buff1
    .global      _Buff4
    .global      _Buff12
    .global      _Buff15
    .global      _fil11
    .global      _fil12
    .global      _fil13
    .global      _fil14
    .global      _N

_Ffir1: NOP
    EALLOW
    SETC SXM      ;SXM = #1
    SETC OBJMODE
    EDIS

    ;Buffer 1
    MOVL XAR2, #_Buff1
    MOVL XAR7, #_h1
    MOVL XAR4, #_fil11
    ADDB XAR4, #5h
```

Anexos.

```
SPM 0
ZAPA

RPT #50
||MAC P,*XAR2++,*XAR7++
ADDL ACC,P<<PM
MOVH *XAR4,ACC<<1 ; hacer el ajuste del qi

;Buffer 2
MOVL XAR2,#_Buff4
MOVL XAR7,#_h1
MOVL XAR4,#_fil12
ADDB XAR4,#5h

SPM 0
ZAPA

RPT #50
||MAC P,*XAR2++,*XAR7++
ADDL ACC,P<<PM
MOVH *XAR4,ACC<<1 ; hacer el ajuste del qi

;Buffer 3
MOVL XAR2,#_Buff12
MOVL XAR7,#_h1
MOVL XAR4,#_fil13
ADDB XAR4,#5h

SPM 0
ZAPA

RPT #50
||MAC P,*XAR2++,*XAR7++
ADDL ACC,P<<PM
MOVH *XAR4,ACC<<1 ; hacer el ajuste del qi

;Buffer 4
MOVL XAR2,#_Buff15
MOVL XAR7,#_h1
MOVL XAR4,#_fil14
ADDB XAR4,#5h

SPM 0
ZAPA

RPT #50
||MAC P,*XAR2++,*XAR7++
ADDL ACC,P<<PM
MOVH *XAR4,ACC<<1 ; hacer el ajuste del qi

LRETR
```