

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

Desarrollo de Algoritmos para la Extracción de Parámetros del Canal MIMO y Aplicación con Medidas



AUTORA: Rocío Martín Pardo
DIRECTOR: José María Molina García-Pardo

Julio / 2008



Autor	Rocío Martín Pardo
E-mail del Autor	[rociomartinpardo@gmail.com]
Director(es)	José María Molina García-Pardo.
E-mail del Director	[josemaria.molina@upct.es]
Título del PFC	Desarrollo de Algoritmos para la Extracción de Parámetros del Canal MIMO y Aplicación con Medidas
<p>Resumen</p> <p>Los sistemas de comunicaciones móviles celulares de tercera generación o los sistemas futuros requieren de gran capacidad de transmisión de datos en la interfaz radio. Por tanto, resulta necesario analizar el canal de radiocomunicaciones no sólo para estimar las pérdidas por propagación sino para evaluar la dispersión de la señal en el tiempo y en el espacio. Además, recientemente se ha descubierto que mediante el uso de múltiples antenas en transmisión y múltiples antenas en recepción (MIMO) se pueden alcanzar grandes tasas de transferencia de información comparado con los sistemas tradicionales. Por todo lo anterior, resulta de gran interés el analizar el uso de estos novedosos sistemas en entornos micro-celulares comunes. Una forma de conocer el canal de radiocomunicaciones es proponiendo modelos de propagación basados en trazado de rayos y teorías electromagnéticas de óptica geométrica, cuyos resultados deben ser validados con medidas experimentales. En el desarrollo de modelos, es primordial conocer las direcciones de llegada y de salida de las diferentes contribuciones existentes en el canal para poder modelizarlas.</p> <p>El objetivo del proyecto es la programación de un algoritmo para la estimación de los ángulos de llegada, salida y retardo de la energía para sistemas de múltiples antenas. El algoritmo elegido para tal efecto es un algoritmo paramétrico denominado SAGE (Space Alternating Generalized Expectation Maximization) que es una aproximación del algoritmo de máxima verosimilitud (ML). También se realizará una comparativa de este algoritmo con otros como MUSIC, ESPRIT o BEAMFORMING mediante el análisis de medidas.</p>	
Titulación	Ingeniero de Telecomunicaciones
Intensificación	Sistemas y Redes de Telecomunicaciones
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	Julio – 2008



INDICE

INDICE	4
Indice de Figuras	6
Capítulo 1. Introducción.....	10
1.1 Contexto	10
1.2 Objetivo	10
1.3 Trabajo previo	11
1.4 Entorno de trabajo	11
1.5 Estructura del proyecto	11
Capítulo 2. Algoritmos	13
2.1 Introducción.....	13
2.2 Modelo de datos del canal SIMO	13
2.3 Modelo de datos del canal MIMO.....	17
2.3.1 Restricciones	18
2.3.2 Estimación de la matriz de transferencia.....	19
2.3.3 Modelo de datos	20
2.3.4 Propiedades	21
2.4 Conceptos Generales de los Algoritmos.....	22
2.4.1 Algoritmo Beamforming	22
2.4.2 Algoritmo MUSIC.....	23
2.4.3 Algoritmo 2D-Unitary ESPRIT	23
2.4.4 Algoritmo SAGE	26
Capítulo 3. Adaptación de Algoritmos	30
3.1 Estructura General	30
3.1.1. Beamforming.....	31
3.1.2. MUSIC	31
3.1.3 2-D Unitary ESPRIT.....	32
3.1.4 SAGE.....	32
3.2 Funciones.....	33
3.2.1 Funciones Comunes.....	34



3.2.2 Beamforming.....	34
3.2.3 MUSIC	35
3.2.4 2-D ESPRIT	35
3.2.5 SAGE.....	36
Capítulo 4. Resultados Entorno Sintético.....	41
Capítulo 5. Resultados Entorno Real.....	68
5.1 Medidas	68
Capítulo 6. Conclusiones	93
Referencias.....	95
Apéndices	97
Apéndice I. Verosimilitud	97
Apéndice II.....	98
II.1 Operador Vec	98
II.2 Producto Kronecker.....	98
II.3 Teorema.....	99



Índice de Figuras

Figura 2.2.1 - Geometría de un array uniforme lineal (ULA) y circular (UCA)	14
Figura 2.2.2 – Definición del ángulo de llegada para un array lineal	15
Figura 2.2.3 – Representación de la frecuencia espacial en función del ángulo de llegada. [6].....	15
Figura 2.3.1 – Propagación multitrayecto en entornos móvil radio	17
Figura 2.4.1- Selección de los sub-arrays para un canal MIMO de M=4x4 sensores	25
Figura 2.4.2- Diagrama de flujo del algoritmo SAGE [7].....	27
Figura 2.4.3- Implementación de la función z.....	28
Figura 3.1- Pantalla de Entorno Simulado.....	37
Figura 3.2 - Área de Introducción de Parámetros.....	38
Figura 3.3 – Área de Calculo de H y Aplicación de Algoritmos	39
Figura 3.4 – Área de Representación gráfica	39
Figura 3.5 – Ventana de Resolución.....	40
Figura 4.1.1: Señal generada a partir de los valores por defecto de los parámetros.....	42
Figura 4.1.10: Señal generada al aumentar el número de sensores de los valores por defecto	48
Figura 4.1.11: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	48
Figura 4.1.12: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	49
Figura 4.1.13: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	49
Figura 4.1.14: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada	50
Figura 4.1.15: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	50
Figura 4.1.16: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	51
Figura 4.1.17: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	51
Figura 4.1.18: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	52
Figura 4.1.19: Señal generada tras el aumento del número de snapshots respecto de los valores por defecto de los parámetros.....	53
Figura 4.1.2: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	43
Figura 4.1.20: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	54
Figura 4.1.21: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	54



Figura 4.1.22: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	55
Figura 4.1.23: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada	55
Figura 4.1.24: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	56
Figura 4.1.25: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	56
Figura 4.1.26: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	57
Figura 4.1.27: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	57
Figura 4.1.28: Señal generada a tras la modificación del SNR de las señales	59
Figura 4.1.29: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	59
Figura 4.1.3: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	43
Figura 4.1.30: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	60
Figura 4.1.31: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	60
Figura 4.1.32: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada	61
Figura 4.1.33: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	61
Figura 4.1.34: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	62
Figura 4.1.35: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	62
Figura 4.1.36: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	63
Figura 4.1.37: Señal generada a partir de los valores por defecto de los parámetros.....	64
Figura 4.1.38: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	65
Figura 4.1.39: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	65
Figura 4.1.4: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	44
Figura 4.1.40: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D	66
Figura 4.1.41: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	66
Figura 4.1.5: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	44
Figura 4.1.6: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	45



Figura 4.1.7: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	45
Figura 4.1.8: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D.....	46
Figura 4.1.9: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada.....	46
Figura 5.1.1: Sección transversal del túnel en el que se realizaron las medidas.	68
Figura 5.1.2: Disposición de arrays y contribuciones en el túnel.....	69
Figura 5.1.3: Esquema del equipo de medidas utilizado para MIMO-Banda Ancha.....	69
Figura 5.1.4: Fotos del equipo. A) Transmisor. B) Receptor.....	70
Figura 5.1.5: Configuración de las medidas.....	71
Figura 5.2.1: Resultados de los algoritmos en la estimación del parámetro DOD para cada una de las distancias de medición a una frecuencia de 3GHz.	72
Figura 5.2.2: Resultados de los algoritmos en la estimación del parámetro DOA para cada una de las distancias de medición a una frecuencia de 3GHz.	73
Figura 5.2.3: Resultados de los algoritmos en la estimación del retardo para cada una de las distancias de medición a una frecuencia de 3GHz.....	74
Figura 5.2.4: Resultados de los algoritmos en la estimación del parámetro DOD para cada una de las distancias de medición a una frecuencia de 4GHz.	79
Figura 5.2.5: Resultados de los algoritmos en la estimación del parámetro DOA para cada una de las distancias de medición a una frecuencia de 4GHz.	80
Figura 5.2.6: Resultados de los algoritmos en la estimación del retardo para cada una de las distancias de medición a una frecuencia de 4GHz.....	81
Figura 5.2.7: Resultados de los algoritmos en la estimación del parámetro DOD para cada una de las distancias de medición a una frecuencia de 5GHz.	86
Figura 5.2.8: Resultados de los algoritmos en la estimación del parámetro DOA para cada una de las distancias de medición a una frecuencia de 5GHz.	87
Tabla 4.1.1: Tabla de parámetros estándar.....	41
Tabla 4.1.10: Tabla de parámetros estándar con la información de la quinta simulación....	64
Tabla 4.1.11: Tabla de parámetros estimados tras la quinta simulación.	67
Tabla 4.1.2: Tabla de parámetros estándar con la información de la primera simulación. ..	42
Tabla 4.1.3: Tabla de parámetros estimados tras la primera simulación.....	47
Tabla 4.1.4: Tabla de parámetros estándar con la información de la segunda simulación. .	47
Tabla 4.1.5: Tabla de parámetros estimados tras la segunda simulación.	52
Tabla 4.1.6: Tabla de parámetros estándar con la información de la tercera simulación.....	53
Tabla 4.1.7: Tabla de parámetros estimados tras la tercera simulación.	58
Tabla 4.1.8: Tabla de parámetros estándar con la información de la cuarta simulación.....	58
Tabla 4.1.9: Tabla de parámetros estimados tras la cuarta simulación.....	63
Tabla 5.1: Tabla de resultados obtenidos con el algoritmo Beamforming a 3GHZ.....	75
Tabla 5.10: Tabla de resultados obtenidos con el algoritmo MUSIC a 3GHZ.	89
Tabla 5.11: Tabla de resultados obtenidos con el algoritmo 2D Unitary ESPRIT a 5GHZ. 87	
Tabla 5.12: Tabla de resultados obtenidos con el algoritmo SAGE a 5GHZ.....	91
Tabla 5.2: Tabla de resultados obtenidos con el algoritmo Music a 3GHZ.	76
Tabla 5.3: Tabla de resultados obtenidos con el algoritmo 2D Unitary ESPRIT a 3GHZ... 77	
Tabla 5.4: Tabla de resultados obtenidos con el algoritmo SAGE a 3GHZ.....	78
Tabla 5.5: Tabla de resultados obtenidos con el algoritmo Beamforming a 4GHZ.....	82



Tabla 5.6: Tabla de resultados obtenidos con el algoritmo Music a 4GHZ.....	83
Tabla 5.7: Tabla de resultados obtenidos con el algoritmo 2D Unitary ESPRIT a 4GHZ...	84
Tabla 5.8: Tabla de resultados obtenidos con el algoritmo SAGE a 4GHZ.....	85
Tabla 5.9: Tabla de resultados obtenidos con el algoritmo Beamforming a 5GHZ.....	88



Capítulo 1. Introducción

1.1 Contexto

En los últimos años se ha experimentado un gran crecimiento en el mercado de las comunicaciones móviles. Este gran crecimiento ha generado la necesidad de conseguir tanto mayores coberturas como mejores calidades de servicio (QoS). Para los nuevos servicios aparecidos, tales como servicios multimedia y transmisión de datos, se requieren velocidades de datos muy elevadas que cubran las necesidades de las futuras generaciones de sistemas de radio móviles. Este incremento de tráfico exige un aumento de capacidad en las redes. Uno de los métodos empleados para este propósito es el uso de dispositivos con múltiples antenas, es decir, con arrays de antenas [1] explotando la propagación multitrayecto. Su uso, junto con técnicas de codificación específicas, puede mejorar el funcionamiento móvil radio en términos de capacidad y velocidad de datos.

La utilización de estos arrays de antenas tanto en el transmisor como en el receptor en un medio que presente la suficiente dispersión (scattering) constituye lo que se denomina un sistema MIMO (Multiple-Input Multiple-Output) [2], [3].

La optimización de los esquemas que implementan estas técnicas es sólo factible cuando los modelos exactos de canal se hacen válidos para los diseñadores de sistemas. Estos modelos deben reproducir de una manera realista todo el comportamiento dispersivo del canal de propagación, por ejemplo, en retardo, dirección, polarización, etc. La elaboración del modelo efectivo de canal está relacionada con la caracterización realista de la función densidad de probabilidad de los parámetros relevantes del canal.

El conocimiento de la dirección de llegada en el lado de la transmisión debería ser suficiente para guiar al usuario, sin embargo, en un escenario en el que la línea de visión se encuentre obstruida, presentando ondas reflejadas, el ángulo de llegada se vuelve insuficiente y necesitamos conocer la dirección de salida.

Aplicar métodos que nos permitan estimar estos parámetros utilizando algoritmos de transmisión y recepción permite mejorar su funcionamiento.

1.2 Objetivo

El objetivo principal de este proyecto es el de comparar la bondad de los resultados de diferentes algoritmos de estimación de parámetros (Beamforming, 2-D Unitary ESPRIT, MUSIC y SAGE) para así poder estudiar su comportamiento ante diferentes condiciones de medición y llegar a conclusiones sobre su comportamiento.



Los parámetros que vamos a considerar van a ser la dirección de salida (DOD, Direction Of Departure), la dirección de llegada (DOA, Direction Of Arrival), el retardo y la potencia de cada contribución. Cada uno de los algoritmos está implementado en MatLab. Para la correcta comparación de los mismos, éstos se analizarán en dos entornos: un entorno real y otro simulado. En el caso del entorno simulado se han adaptado y añadido los algoritmos a una herramienta que fue creada por otros alumnos como PFC y en la cual podremos observar los diferentes resultados obtenidos con la utilización de los diferentes algoritmos (Beamforming, MUSIC, 2D-Unitary ESPRIT y SAGE).

Finalmente, en el caso del entorno real, se llevará a cabo la comparación de los resultados obtenidos en el entorno simulado con las medidas reales realizadas por el grupo de investigación TELICE (Télécommunications, Interférences et Compatibilité Electromagnétique) de la Universidad de Lille (Francia) en las que se han utilizado arrays lineales uniformes (ULA) tanto en recepción como en transmisión.

1.3 Trabajo previo

En una primera parte se llevará a cabo un estudio teórico en el que se obtendrán los resultados teóricos a los que deberían corresponder las estimaciones de los algoritmos Beamforming, MUSIC, 2D-Unitary ESPRIT y SAGE que se obtendrán posteriormente en la segunda parte del proyecto.

La segunda parte de este proyecto consistirá en la adaptación de los diferentes algoritmos para su correcto funcionamiento ante un generador de señales y, por otro lado, en la adaptación de los mismos para la obtención de las estimaciones sobre las medidas reales obtenidas por TELICE.

1.4 Entorno de trabajo

Durante el desarrollo del proyecto consideraremos arrays lineales de antenas (ULA) tanto en recepción como en transmisión. Consideraremos también que la propagación se produce en el plano horizontal en un medio multicamino. La separación entre los sensores del array es $\frac{\lambda}{2}$ en el caso de máxima frecuencia, para el resto de frecuencias el espaciado entre sensores cumplirá la relación $\frac{\lambda_{max}}{2\lambda_0}$.

1.5 Estructura del proyecto

En primer lugar se presenta una pequeña introducción de la que forma parte este apartado en la que se explica a modo de resumen los pasos a seguir y los objetivos de este proyecto. En el apartado siguiente, Algoritmos, se explicará tanto el modelo del canal como el



modelo de señal y, posteriormente, conceptos generales de los algoritmos. El paso siguiente será el de la adaptación de los algoritmos en MatLab, incluyendo aclaraciones sobre cada una de las funciones utilizadas. Posteriormente se hará un análisis de los resultados obtenidos tanto en entorno simulado y en entorno real ya haciendo uso de SAGE, Beamforming, 2D-Unitary ESPRIT, MUSIC finalizando con las conclusiones obtenidas al observar los resultados obtenidos para cada uno de ellos.



Capítulo 2. Algoritmos

2.1 Introducción

La idea principal del algoritmo Beamforming es que, en un canal MIMO, la energía que se propaga en cada instante se concentra en una dirección de llegada y en una dirección de salida. El algoritmo Beamforming es una extensión natural del análisis espectral de Fourier clásico.

El algoritmo MUSIC se basa en las propiedades de descomposición en autovalores de la matriz de covarianza teniendo en cuenta que el subespacio de señal y el subespacio de ruido son ortogonales y, por tanto, los vectores propios del ruido serán perpendiculares al subespacio de señal.

El algoritmo Unitary ESPRIT es un algoritmo basado en técnicas basadas en el subespacio para estimar DoAs o frecuencias. En nuestro caso utilizaremos una particularización de este algoritmo a dos dimensiones. En relación al coste computacional del algoritmo, 2D Unitary ESPRIT resulta más potente que otros métodos basados en el espectro y que las técnicas ML. Además esta técnica requiere una sola descomposición en autovalores y algunos productos entre matrices por lo que no es necesario buscar una optimización

El algoritmo SAGE actualiza los parámetros secuencialmente alternando con varios espacios ocultos de datos definidos por el diseñador del algoritmo [5], en vez de utilizar un único espacio completo de datos, asociamos cada grupo de parámetros con un subespacio oculto de datos. Esta reducción conduce a una convergencia más rápida. La tasa de convergencia es una de las motivaciones del método SAGE frente al método EM.

2.2 Modelo de datos del canal SIMO

En el modelo que vamos a utilizar durante el proyecto nos interesará trabajar con señales de banda reducida representadas por sus equivalentes banda base. Esto no es realmente una restricción, puesto que cualquier señal puede expresarse como una combinación lineal de componentes en banda reducida. Empleando notación compleja la parte real de una señal paso-banda con frecuencia central f_c , transmitida en un canal físico se puede escribir como:

$$z(t) = \text{Re} \{s(t)e^{j2\pi f_c t}\} \quad (2.1)$$

Donde la señal en banda base $s(t)$ es la envolvente compleja de la señal transmitida $z(t)$. Trabajar con la representación equivalente banda base es más sencillo que hacerlo con la representación paso-banda.

Por otra parte una propiedad importante de las señales en banda reducida es que los retardos en el tiempo son menores que el tiempo de coherencia resultante de sumar los desplazamientos de fase. Con lo que si la frecuencia central de la señal es f_c un retardo pequeño de tiempo, τ , se corresponderá con un desplazamiento de fase $e^{j2\pi f_c \tau}$ en la representación banda base compleja de la señal.

Esta formulación de la señal implica que se aceptan las condiciones impuestas al suponer campo lejano, es decir el radio de propagación de la señal, o la distancia entre la fuente y el array, se considera suficientemente grande como para considerar un frente de onda plano.

Por otro lado supondremos también que se trata de un medio lineal lo que implica la validez del principio de superposición y permite más de una onda viajando por el medio.

A continuación mostraremos dos formas geométricas comunes que puede presentar un array de sensores:

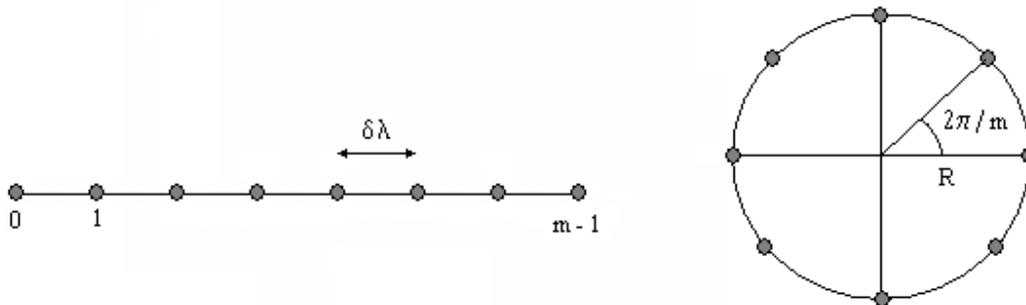


Figura 2.2.1 - Geometría de un array uniforme lineal (ULA) y circular (UCA)

Nosotros consideraremos un array lineal (ULA) consistente en m antenas uniformemente espaciadas por δ longitudes de onda, λ , a lo largo de una línea. Las antenas se asumen omnidireccionales y la ganancia de cada antena se normaliza a uno.

Si llamamos $s(t)$ a la señal recibida por la primera antena entonces la señal recibida en cualquier antena sufrirá un retardo adicional τ_i , que dependerá de la distancia a la que se encuentre de la primera antena, y se podrá escribir como:

$$x_i = e^{-j2\pi f_c \tau_i} s(t) = e^{-j(i-1)2\pi f_c \tau_a} s(t) = \alpha_i s(t) \tag{2.2}$$

donde τ_d es el retardo existente entre dos antenas cualesquiera. Este parámetro lo podemos definir en función de la distancia extra que ha de recorrer una señal desde que ha llegado a la antena $m-1$ hasta llegar a la antena m .

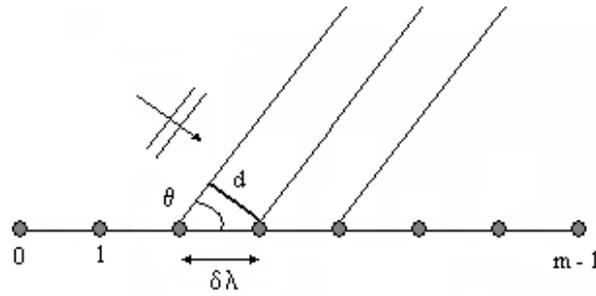


Figura 2.2.2 – Definición del ángulo de llegada para un array lineal

Así podemos escribir:

$$\sin(\theta) = \frac{d}{\delta\lambda} \tag{2.3}$$

$$\tau_d = \frac{\sin(\theta)\delta\lambda}{c} \tag{2.4}$$

donde d se corresponde con la distancia extra a recorrer y c es la velocidad de propagación del medio.

Una señal que llega con un ángulo $\theta=0$ con respecto al array, es decir totalmente perpendicular, provocará un desfase nulo entre las antenas de los arrays ya que llega de la misma manera a todas las antenas del array. A medida que aumenta el ángulo se crea una varianza temporal entre sensores que hace aumentar la frecuencia en el eje del espacio, como se muestra en la siguiente figura.

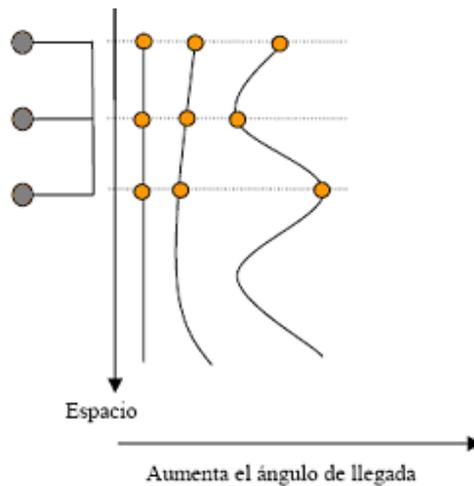


Figura 2.2.3 – Representación de la frecuencia espacial en función del ángulo de llegada. [6]



Hay que tener en cuenta que se nos pueden presentar problemas de *aliasing*, para evitarlo la frecuencia máxima en el eje del espacio será π . En base a esta restricción debemos fijar la distancia mínima entre antenas para que no se produzca aliasing espacial.

Empleamos la expresión que relaciona el tiempo de desfase entre 2 antenas adyacentes podemos escribir:

$$2\pi f_{max} \tau_m \leq \pi, \quad 2f_{max} \frac{\delta \lambda \sin(\theta)}{c} \leq 1 \quad (2.5)$$

$$\delta \lambda \leq \frac{c}{2f_{max} \sin(\theta)} \quad (2.6)$$

Y teniendo en cuenta que el mayor retardo se produce para $\theta = 90^\circ$, finalmente queda:

$$\delta \lambda \leq \frac{\lambda_{min}}{2} \quad (2.7)$$

Que es la distancia mínima entre dos antenas consecutivas del array.

Uniendo las señales recibidas por cada una de las antenas en un único vector $x(t)$, obtenemos $x(t) = a(\theta) \cdot s(t)$, donde $a(\theta)$ es el llamado vector de direcciones. Para un array uniformemente espaciado (ULA) tenemos la siguiente expresión.

$$A(\theta) = \begin{bmatrix} 1 \\ e^{-j2\pi\delta \sin(\theta)} \\ \vdots \\ e^{-j(m-1)2\pi\delta \sin(\theta)} \end{bmatrix} \quad (2.8)$$

Si K señales inciden en el array desde distintas DOAs el vector de salida toma, utilizando el principio de superposición, la forma $x(t) = A(\theta) \cdot s(t)$. Con lo que podemos definir una matriz de direcciones que contenga la respuesta del array a todas las K fuentes y un vector de señales de la forma:

$$A(\theta) = [a(\theta_1) \ a(\theta_2) \ \dots \ a(\theta_K)] \quad (2.9)$$

$$A(\theta) = \begin{bmatrix} 1 & \dots & 1 \\ e^{j\theta_1} & \dots & e^{j\theta_K} \\ \vdots & \ddots & \vdots \\ e^{j(m-1)\theta_1} & \dots & e^{j(m-1)\theta_K} \end{bmatrix} \quad (2.10)$$

$$s(t) = [s_1(t), \dots, s_K(t)]^T \quad (2.11)$$

En presencia de un ruido aditivo $n(t)$, obtenemos el siguiente modelo:

$$x(t) = A(\theta) \cdot s(t) + n(t) \tag{2.12}$$

que es el modelo que se utiliza normalmente para la recepción en el procesado de arrays de antenas ULA.

2.3 Modelo de datos del canal MIMO

Siendo el canal móvil radio, el modelo es MIMO [7], con un entorno multitrayecto:

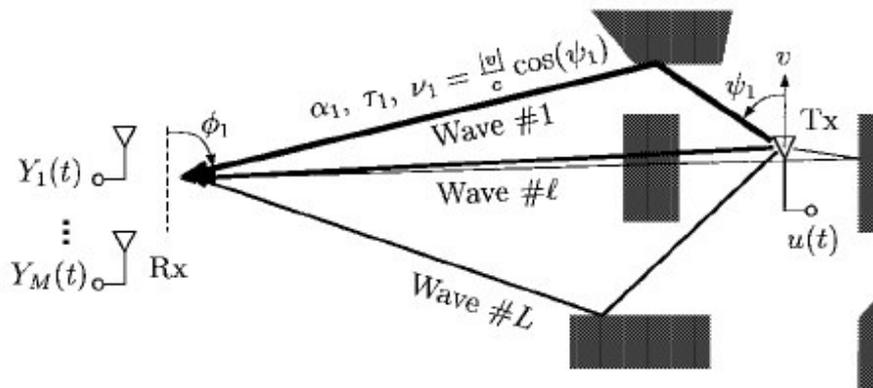


Figura 2.3.1 – Propagación multitrayecto en entornos móvil radio [7].

Asumimos que se producen L trayectos diferentes, estando definidos por su amplitud compleja (α_l), el retardo relativo hasta la llegada a su receptor, los ángulos que definen su DOA (Direction Of Arrival), dirección de llegada ($\Omega_{RX,l}$), los ángulos que definen su DOD (Direction Of Departure), dirección de salida ($\Omega_{TX,l}$). Éste es por tanto el conjunto de parámetros que tratamos de estimar:

$$\theta_L = [\alpha_l, \tau_l, \Omega_{RX,l}, \Omega_{TX,l}] \tag{2.13}$$

Además asumimos que los arrays de transmisión y recepción se encuentran lo suficientemente alejados entre sí como para considerar campo lejano, con lo que cada una de las ondas presenta un frente de onda plano. Por otro lado vamos a establecer que la propagación se produce en el plano horizontal, y sólo estimaremos el ángulo azimut (la extensión de este modelo a un medio real 3-D considera también el ángulo de elevación).



La matriz de transferencia H del canal MIMO relaciona el vector de excitación del array de transmisión $u(n)$ con el vector de respuesta obtenido en el array de recepción $y(n)$ de la siguiente forma:

$$y(n) = H \cdot u(n) \quad (2.14)$$

donde los vectores $y(n)$ y $u(n)$ son las representaciones banda base complejas de las señales.

La matriz de transferencia H está formada por la superposición de las respuestas de los dos arrays se define:

$$H = \sum_{k=1}^K b_k a_{Rx}(\theta_k) a_{Tx}(\varphi_k)^H \quad (2.15)$$

Siendo los parámetros φ_k y θ_k , respectivamente, los ángulos de llegada y salida de la señal k -ésima. Cada camino de propagación se encuentra afectado por una ganancia compleja b_k . El vector $a_{Rx}(\theta_k)$ constituye el vector recibido por el array de recepción en la dirección θ_k , y se define como se mostró en el apartado anterior para el caso de canales SIMO. El vector $a_{Tx}(\varphi_k)$ es el vector respuesta del array de transmisión correspondiente a una señal emitida desde la dirección φ_k . Asumiremos a lo largo del proyecto que tenemos m transmisores y n elementos receptores, con lo que la matriz de transmisión es una matriz $n \times m$.

La formulación anterior se puede expresar en forma matricial de la siguiente forma:

$$H = A_{Rx} B A_{Tx} \quad (2.16)$$

donde $B = \text{diag}\{b\}$ con $b = [b_1, \dots, b_K]^T$. Las matrices de direcciones A_{Tx} y A_{Rx} se definen de la misma manera que la matriz de direcciones del canal SIMO.

2.3.1 Restricciones

En cualquier tipo de medida aparece inevitablemente ruido, por lo que también debe ser modelado. Con lo que tal y como hicimos en el modelo SIMO introducimos el ruido en el modelo:

$$y(n) = H \cdot u(n) + v(n) \quad (2.17)$$

Existen muchas fuentes de ruido [8], como por ejemplo el ruido ambiental que incluye el ruido cósmico y el atmosférico. Otro ejemplo de ruido es el producido por el hombre, causado por ejemplo por equipos eléctricos próximos al receptor. El receptor también introduce ruido, principalmente ruido térmico producido por el movimiento térmico de las partículas existente en todos los materiales. El ruido térmico se modela fácilmente como un proceso gaussiano estacionario, con características espectrales planas, y recibe el nombre



de ruido blanco. El ruido producido por el receptor normalmente constituye la fuente de ruido predominante con lo que asumimos $v(n)$ como ruido aditivo blanco gaussiano (AWGN).

$$E\{v(n)\} = 0 \quad E\{v(n)v(s)^H\} = \sigma_v^2 I \delta_{ns} \quad E\{v(n)v(s)^T\} = 0 \quad \forall n, s \quad (2.18)$$

Se asume también que el ruido producido por el transmisor presenta la misma distribución gaussiana y que la señal transmitida se encuentra incorrelada con el ruido.

La ganancia b_k de la señal se puede modelar de muy diversas formas dependiendo del tipo de señal introducido, en nuestro proyecto utilizaremos la siguiente exponencial.

$$b_k = \beta_k e^{-\frac{k}{\tau}} \quad (2.19)$$

siendo:

$$\tau = \frac{1}{BW} \quad (2.20)$$

$$\beta_k = 1 \quad (2.21)$$

2.3.2 Estimación de la matriz de transferencia

A partir de la ecuación (2.17) podemos estimar la matriz H utilizando el método de los mínimos cuadrados (LS) y obtener un sistema sobre determinado. Los arrays de antenas son preprocesados y muestreados con un periodo T , normalizado a $T=1$, con lo que si cogemos N muestras en la matriz tendremos:

$$Y = HU + V \quad (2.22)$$

donde $Y = [y(1) \dots y(N)]$, $U = [u(1) \dots u(N)]$ y $V = [v(1) \dots v(N)]$. Si U fuera conocida debido, por ejemplo, a pruebas anteriores, [9] entonces

$$\hat{H} = YU^+ = YU^H(UU^H)^{-1} \quad (2.23)$$

donde U^+ representa la pseudo-inversa de Moore-Penrose de U . Finalmente queda la siguiente expresión:

$$\hat{H} = H + \tilde{H} = A_{Rx} B A_{Tx} + N \quad (2.24)$$

donde N representa la estimación del error que todavía es coloreado debido a que $N = VU^+$. Sin embargo simplificamos asumiendo que N es blanco.

Si U es blanco la varianza del ruido N se relaciona con V de la siguiente forma



$$\sigma^2 = \frac{\sigma_v^2}{N} \quad (2.25)$$

Esta relación se ha verificado mediante datos simulados y establece que incrementar el número de muestras disminuye la varianza del error estimado en H.

La ecuación final, definida en (2.24), establece el modelo de datos que vamos a utilizar a lo largo del proyecto, aunque modificado para poder aplicar los distintos métodos de estimación que veremos en el apartado 2.4 de este capítulo.

2.3.3 Modelo de datos

El objetivo final de este proyecto consiste en estimar los ángulos de salida y llegada, el retardo de propagación y la potencia de las distintas contribuciones a partir de la matriz de transferencia H. Como vimos en la introducción existen muchos algoritmos para estimar estos parámetros en el caso de un único array de antenas, en nuestro caso, al tratarse de un canal doble direccional el problema adquiere una dimensión más y los datos pasan de ser vectores a matrices. Con lo que el principal objetivo será sectorizar la matriz apilando sus columnas, ya que de esta manera separamos los ángulos de salida de los de llegada. Utilizando el operador vec (Apéndice II) la ecuación (2.15) queda:

$$\text{vec}H = \text{vec}\left(\sum_{k=1}^K b_k a_{Rx}(\theta_k) a_{Tx}(\varphi_k)^H\right) = \sum_{k=1}^K \text{vec}\left(a_{Rx}(\theta_k) a_{Tx}(\varphi_k)^H\right) b_k \quad (2.26)$$

Expresándolo en forma matricial:

$$\text{vec}H = A(\varphi, \theta) b \quad (2.27)$$

$$\begin{aligned} \text{Donde } A(\varphi, \theta) &= [\text{vec}(a_{Rx}(\theta_1) a_{Tx}(\varphi_1)^H) \dots \text{vec}(a_{Rx}(\theta_K) a_{Tx}(\varphi_K)^H)] \\ &= [(a_{Tx}(\varphi_1)^c \otimes a_{Rx}(\theta_1)) \dots \text{vec}(a_{Tx}(\varphi_K)^c \otimes a_{Rx}(\theta_K))] \end{aligned} \quad (2.28)$$

El subíndice c representa el complejo conjugado y \otimes el producto Kronecker. La matriz A es una matriz M por K, donde M=mxn. Finalmente denominamos vec(H) como h, y teniendo en cuenta que en presencia de ruido gaussiano la operación vec no cambia las propiedades del ruido, obtenemos:

$$h(\mathbf{t}) = A(\varphi, \theta) b(\mathbf{t}) + n(\mathbf{t}) \quad (2.29)$$

Expresión que podemos identificar con el modelo (2.12) definido para el caso del canal SIMO. El conjunto de datos $\{h(\mathbf{t})\}$ se corresponde con $\{x(\mathbf{t})\}$ y el vector $b(\mathbf{t})$, conteniendo las ganancias de las señales de cada dirección de propagación, se corresponde con las formas de onda de las señales denominadas $s(\mathbf{t})$ en el modelo para el canal SIMO. La principal diferencia es que la matriz de direcciones A es una función de dos parámetros, por ejemplo dos ángulos por rayo.



2.3.4 Propiedades

En este apartado estudiaremos las propiedades básicas del modelo (2.29). Muchos de los esquemas utilizados para estimar los ángulos de salida y llegada dependen de las propiedades del momento de segundo orden de los datos medidos. Por lo tanto, podemos definir la matriz de covarianza del conjunto de datos finitos $\{h(t)\}$ como

$$R_h = E\{h(t)h(t)^H\} = A(\varphi, \theta) E\{b(t)b(t)^H\} A(\varphi, \theta)^H + E\{n(t)n(t)^H\} \quad (2.30)$$

donde $E\{\square\}$ denota la esperanza estadística, la media.

$$E\{b(t)b(t)^H\} = P \quad (2.31)$$

La ecuación anterior es la matriz de covarianza del camino de propagación, y

$$E\{n(t)n(t)^H\} = \sigma^2 I \quad (2.32)$$

es la matriz de covarianza del ruido, con lo que finalmente queda:

$$R_h = E\{h(t)h(t)^H\} = A(\varphi, \theta) P A(\varphi, \theta)^H + \sigma^2 I \quad (2.33)$$

La factorización espectral de R_h resulta de gran importancia y es muy utilizada en el campo del procesamiento de arrays [10] por lo que también podemos representar la matriz de covarianza R_h mediante su descomposición en autovalores. Si el canal MIMO presenta K contribuciones propagándose y no se encuentran totalmente correladas entre sí, el producto APA tendrá un rango K y R_h podrá expresarse como la suma de dos partes.

$$R_h = E A E^H = E_s \Lambda_s E_s^H + E_n \Lambda_n E_n^H \quad (2.34)$$

Una parte se encuentra relacionada con la señal donde $E_s = [e_1, \dots, e_K]$ representa los autovectores de la señal y Λ_s es la matriz diagonal conteniendo los K autovalores dominantes. La segunda parte está constituida por autovectores, representada por $E_n = [e_{K+1}, \dots, e_M]$, y se corresponde con los autovalores iguales a la varianza del ruido $\Lambda_n = \sigma^2 I$. Finalmente, puesto que el subespacio que se extiende a lo largo de las columnas de E_s es igual al subespacio que se extiende por las columnas de A , se suele llamar subespacio de señal, o, en nuestro caso, camino de la señal y el correspondiente espacio extendido de E_n recibe el nombre de subespacio de ruido. Ambos espacios son ortogonales.

Una estimación natural de la matriz de covarianza se basa en las muestras de los datos utilizadas,

$$\hat{R}_h = \frac{1}{N} \sum_{y=1}^N h(t)h(t)^H \quad (2.35)$$



una representación espectral para esta expresión se definiría como

$$\mathbf{R}_h = \hat{\mathbf{E}}_s \hat{\Lambda}_s \hat{\mathbf{E}}_s^H + \hat{\mathbf{E}}_n \hat{\Lambda}_n \hat{\mathbf{E}}_n^H \quad (2.36)$$

Expresión utilizada en las distintas descripciones e implementaciones de los algoritmos.

2.4 Conceptos Generales de los Algoritmos

En este apartado se explicarán los aspectos generales de los algoritmos, dando una visión general de los mismos para así obtener una mejor comprensión de su funcionamiento en la estimación de parámetros.

2.4.1 Algoritmo Beamforming

Como ya se mencionó al inicio de este capítulo, en el apartado *Introducción*, éste es un algoritmo que basado en el análisis espectral de Fourier.

La idea de que, sobre un canal MIMO, en cada instante la energía que se propaga se encuentra concentrada en una dirección de llegada y en una dirección de salida se puede expresar como $\mathbf{y}(t) = \mathbf{w}^H \mathbf{h}(t)$ donde \mathbf{w} es el vector de pesos que resalta un camino particular de propagación con sus respectivos ángulos. Dadas las muestras de la señal y desde 1 hasta N la potencia en estas técnicas se mide como:

$$P(\mathbf{w}) = \frac{1}{N} \sum_{t=1}^N |\mathbf{y}(t)|^2 = \frac{1}{N} \sum_{t=1}^N \mathbf{w}^H \mathbf{h}(t) \mathbf{h}(t)^H \mathbf{w} = \mathbf{w}_h^H \hat{\mathbf{R}}_h \mathbf{w} \quad (2.37)$$

Siendo $\hat{\mathbf{R}}_h$ la matriz de covarianza, definida en el apartado de *Propiedades*. Diferentes tipos de aproximaciones para el vector de pesos \mathbf{w} dan lugar a distintas técnicas de Beamforming. En nuestro caso el algoritmo utilizado es el Beamforming convencional.

El vector de pesos utilizado para maximizar la potencia en un determinado camino de propagación es [15]:

$$\mathbf{w}_{Beamforming} = \frac{\mathbf{a}(\varphi, \theta)}{\sqrt{\mathbf{a}(\varphi, \theta)^H \hat{\mathbf{R}}_h \mathbf{a}(\varphi, \theta)}} \quad (2.38)$$

En este caso $\mathbf{a}(\varphi, \theta)$ representa el vector de direcciones del canal MIMO. Si insertamos este valor en la ecuación (2.37) obtenemos el espectro Beamforming Clásico que queda:

$$P_{Beamforming} = \frac{\mathbf{a}(\varphi, \theta)^H \hat{\mathbf{R}}_h \mathbf{a}(\varphi, \theta)}{\mathbf{a}(\varphi, \theta)^H \mathbf{a}(\varphi, \theta)} \quad (2.39)$$

Quedando el espectro de este modo representado como una función de los parámetros DOA y DOD. El espectro mostrará los picos de potencia en aquellas localizaciones que presenten máxima potencia. En la práctica la resolución de este algoritmo depende del tamaño de los ángulos introducidos como φ y θ .



2.4.2 Algoritmo MUSIC

En el caso de este algoritmo se basa en las propiedades de la descomposición en autovalores de la matriz de covarianza. Como se vio en el apartado de *Propiedades* la señal y el ruido son ortogonales, esta propiedad implica que los autovectores del ruido son perpendiculares a la matriz A o al subespacio expandido estimado de la señal (E_s). Teniendo en cuenta esto, se establece la siguiente condición de ortogonalidad:

$$E_n^H a(\varphi, \theta) = 0, \quad (\varphi, \theta) \in \{(\varphi_1, \theta_1), \dots, (\varphi_K, \theta_K)\} \quad (2.40)$$

La idea fundamental del algoritmo es encontrar K vectores $a(\varphi, \theta)$ que sean lo más ortogonales posible a E_n . El subespacio de señal se estima a partir de la descomposición en autovalores de la matriz de covarianza estimada R_h .

Finalmente el espectro de señal generado por el algoritmo MUSIC se puede definir como:

$$P_{MUSIC}(\varphi, \theta) = \frac{a^H(\varphi, \theta) a(\varphi, \theta)}{a^H(\varphi, \theta) E_n E_n^H a(\varphi, \theta)} \quad (2.41)$$

2.4.3 Algoritmo 2D-Unitary ESPRIT

Este algoritmo es un algoritmo basado en subespacios, como ya se explicó en el apartado *Introducción* de este capítulo.

Debido a su simplicidad y rapidez este algoritmo se ha convertido en la más potente de las técnicas basadas en el subespacio para estimar DOAs o frecuencias. Para geometrías de arrays centro-simétricas se ha formulado un tipo de algoritmo ESPRIT (Unitary ESPRIT) que reduce considerablemente la complejidad. En nuestro caso utilizaremos la particularización de esta extensión a dos dimensiones.

Comenzaremos considerando un elemento de la matriz $H(t)$ del canal:

$$H_{l,p}(t) = \sum_{k=1}^K b_p(t) e^{j\mu_k(k-1)} e^{-j\nu_k(p-1)} + n_{l,p}(t) \quad (2.42)$$

Siendo l y p, respectivamente, las filas y las columnas de la matriz. Las frecuencias μ_k y ν_k son, respectivamente, función del DOA y del DOD de cada rayo k. Utilizando este algoritmo podemos estimar dos vectores de frecuencias con sus respectivos ángulos estimados:

$$\mu = [\mu_1 \mu_2 \dots \mu_N]^T \quad (2.43) \quad \text{y} \quad \nu = [\nu_1 \nu_2 \dots \nu_N]^T \quad (2.44)$$

Todos los canales MIMO se vectorizan y se apilan formando una matriz (modelo 2.29). Estos vectores de medidas constituyen las columnas de una matriz de datos de dimensión



$M \times N$ que denominamos X . A partir de ahora denominaremos r a la dimensión de la matriz y podrá tomar los valores 1 y 2.

El algoritmo Unitary ESPRIT explota la propiedad centro-simétrica de los arrays de sensores, es decir, sus elementos son simétricos con respecto a un centroide y los patrones de respuesta complejos son los mismos. Una matriz de un canal MIMO con $M = M_1 \times M_2 = n \times m$ elementos posee esta propiedad y se puede considerar como un array uniforme rectangular (URA) (siendo M_1 el número de sensores en el array de transmisión y M_2 el número de sensores en el array de recepción).

Debido a que un array de sensores posee una estructura dual-invariante, se pueden formar dos subarrays idénticos, denominados matrices de selección de tamaño $m_r \times M$, $1 \leq r \leq 2$, que sean centro-simétricos entre sí:

$$J_{(r)1} = \prod_{m_r} J_{(r)2} \prod_M \quad (2.45)$$

$$m_r = (M_r - 1) \prod_{k=r}^2 M_k = \frac{M(M_r-1)}{M_r}, \quad 1 \leq r \leq 2 \quad (2.46)$$

donde \prod_M representa una matriz $M \times M$ formada por unos en su antidiagonal y ceros en las demás posiciones.

Para obtener una fórmula explícita para las matrices de selección definimos

$$J_2^{(M_r)} = [0_{(M_r-1) \times 1} \quad I_{M_r-1}] \quad (2.47)$$

Como una matriz de selección 1-D que se corresponde con el mayor solapamiento posible a lo largo de la dimensión r . Aquí $0_{(M_r-1) \times 1}$ representa a una matriz de ceros de tamaño $(M_r - 1) \times 1$. A partir de esta expresión podemos construir la matriz de selección esperada de dimensión $m_r \times M$ de la siguiente forma:

$$J_{(r)2} = I_{\prod_{k=(r+2)}^2 M_k} \otimes J_2^{(M_r)} \otimes I_{\prod_{k=1}^{r-1} M_k} \quad (2.48)$$

Una vez que tenemos una de las matrices de selección podemos obtener la otra introduciéndola en la ecuación (2.45). A continuación mostramos gráficamente una posible opción de las matrices de selección para un canal MIMO con ULAs formados por 4 elementos tanto en transmisión como en recepción [16].

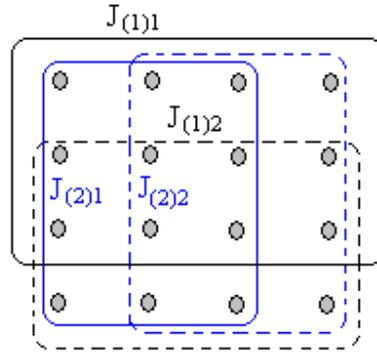


Figura 2.4.1- Selección de los sub-arrays para un canal MIMO de M=4x4 sensores

A continuación la media hacia delante y hacia atrás se logra transformando la matriz de datos X en la matriz $[X \quad \prod_M X^c \prod_M]$ de tamaño $M \times 2N$, que es una matriz de datos compleja. Esta matriz se transforma en una matriz real del mismo tamaño quedando:

$$\Gamma(X) = Q_M^H [X \quad \prod_M X^c \prod_M] Q_{2N} \quad (2.49)$$

donde el operador c indica la matriz compleja conjugada pero no transpuesta y las matrices Q se definen como las matrices unitarias \prod —reales a la izquierda. Siendo para el caso par e impar respectivamente:

$$Q_{2n} = \begin{bmatrix} I_n & jI_n \\ \Pi_n & -j\Pi_n \end{bmatrix} \quad Q_{2n+1} = \begin{bmatrix} I_n & 0 & jI_n \\ 0^T & \sqrt{2} & 0^T \\ \Pi_n & 0 & -j\Pi_n \end{bmatrix} \quad (2.50) \text{ y } (2.51)$$

A continuación calculamos los K autovectores dominantes de la matriz de covarianza de $\Gamma(X)$ y formamos el subespacio de señal E_s , a partir del cual podemos obtener las frecuencias espaciales relacionadas con los parámetros de interés resolviendo el siguiente conjunto de ecuaciones utilizando la técnica de los mínimos cuadrados [17].

$$K_{(r)1} E_s \Psi_r \approx K_{(r)2} E_s \quad (2.52)$$

Las matrices de selección $K_{(r)1}$ y $K_{(r)2}$ se obtienen de las matrices de selección de la siguiente forma:

$$K_{(r)1} = 2\text{Re}\{Q_{m_r}^H J_{(r)2} Q_M\} \quad (2.53)$$

$$K_{(r)2} = 2\text{Im}\{Q_{m_r}^H J_{(r)2} Q_M\} \quad (2.54)$$

El siguiente paso del algoritmo consiste en estimar las frecuencias espaciales calculando la autodescomposición de la matriz compleja definida como:

$$\Psi_1 + j\Psi_2 = TAT^{-1} \quad \text{con} \quad A = \text{diag}\{\lambda_k\}_{k=1}^K \quad (2.55)$$



Tomando la parte real y la imaginaria de los anteriores autovalores obtenemos automáticamente las parejas de frecuencias

$$\mu_k = 2 \arctan(\operatorname{Re}\{\lambda_k\}) \quad k = 1, 2, \dots, K \quad (2.56)$$

$$v_k = 2 \arctan(\operatorname{Im}\{\lambda_k\}) \quad k = 1, 2, \dots, K \quad (2.57)$$

A continuación, podemos extraer fácilmente los DODs y DOAs a partir de las frecuencias.

$$DOD_k = \arcsen\left(\frac{v_k}{\pi}\right) \frac{180}{\pi} \quad k = 1, 2, \dots, K \quad (2.58)$$

$$DOA_k = \arcsen\left(\frac{\mu_k}{\pi}\right) \frac{180}{\pi} \quad k = 1, 2, \dots, K \quad (2.59)$$

Y finalmente, a partir de los ángulos estimados, reconstruimos la señal:

$$S = (DT^{-1}E_s^H Q_M^H)X \quad (2.60)$$

Donde $D \in C^{K \times K}$ es una matriz de escala unitaria aleatoria.

El algoritmo 2-D Unitary ESPRIT no requiere ninguna búsqueda preliminar de las estimaciones, y puede ser aplicado directamente. Otra característica importante es que esta técnica requiere solamente una descomposición en autovalores y algunos productos entre matrices, con lo que no se necesita buscar una optimización.

2.4.4 Algoritmo SAGE

SAGE, como cualquier método basado en EM [4], resuelve el problema de la estimación de los parámetros del canal de manera iterativa: en cada iteración se computa una expectativa de una función relacionada con la verosimilitud de las observaciones y, después la maximiza para obtener la estimación de los parámetros.

La idea de SAGE y mayor diferencia respecto a EM se basa en 2 conceptos:

- Separar los parámetros en varios subconjuntos, que pueden intersectar, de tal manera que en cada iteración algunos de los subconjuntos permanecen fijos mientras los otros son actualizados (posteriormente, los parámetros ahora fijos pueden actualizarse o no).
- Se introduce un espacio de datos oculto respecto a los parámetros. Suponemos que nuestras observaciones que son conocidas, están incompletas y que, además, existe un conjunto “completo” de datos inobservables que está relacionado con esas observaciones estadísticamente con una función de probabilidad que depende de los parámetros a estimar.

Por tanto en vez de realizar una estimación simultánea, como en el EM clásico, la realiza secuencialmente para mejorar la tasa de convergencia y reducir la complejidad de la maximización.

SAGE, es un algoritmo general aplicable a diferentes casos de procesamiento de señal, como reconstrucción de imágenes o, como en nuestro caso, estimación de parámetros del canal.

Los pasos a seguir por el algoritmo son [5]:

1. Inicialización.
2. Elección del espacio oculto de datos.
3. Paso E.
4. Paso M.
5. Repetir pasos 3 y 4 hasta alcanzar la condición de convergencia fijada

El diagrama de flujo del algoritmo es:

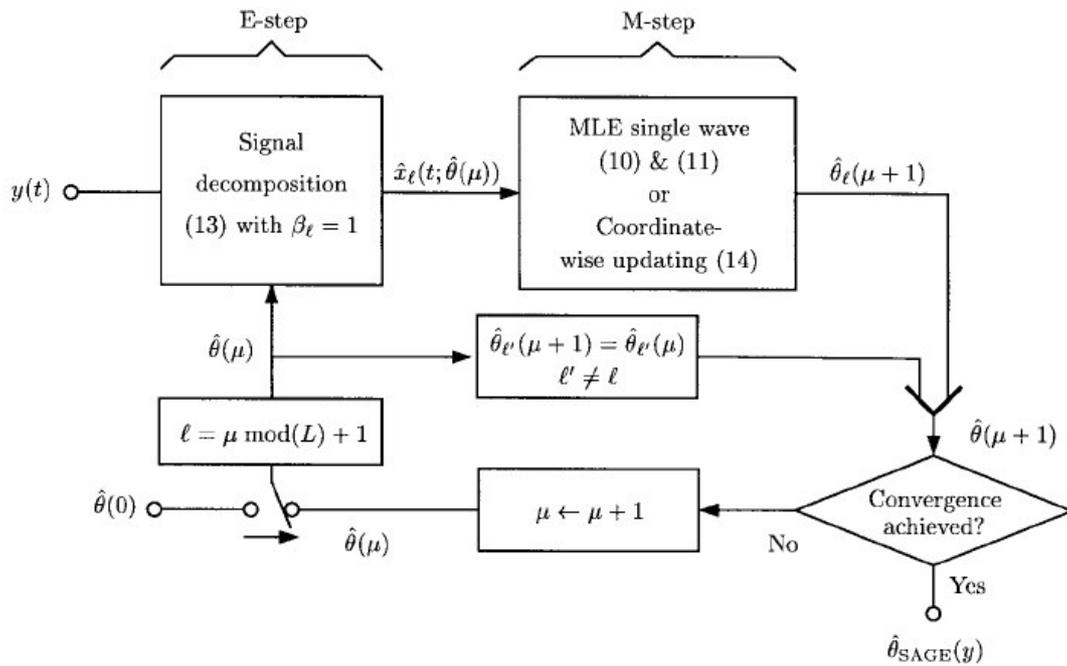


Figura 2.4.2- Diagrama de flujo del algoritmo SAGE [7]

El espacio oculto de datos se identifica como la aportación de las señales individuales $(s(t; \theta_L))$ con su correspondiente ruido aditivo siendo el *conjunto completo* de datos:

$$X_i(t) = s(t; \theta_i) + \sqrt{\frac{N_0}{2}} N_i \quad (2.61)$$

Y la señal recibida:

$$Y(t) = \sum_{i=1}^L X_i(t) \tag{2.62}$$

Dada una observación $x_i(t)$, la verosimilitud logarítmica (Apéndice I) viene dada por:

$$A(\theta_i; x_i(t)) \propto 2 \int_{\mathcal{D}} \Re\{s(t; \theta_i)^H x_i(t)\} dt - \int_{\mathcal{D}} \|s(t; \theta_i)\|^2 dt \tag{2.63}$$

Como $X_i(t)$ es inobservable, no se pueden aplicar directamente las estimaciones ML. El paso E consiste en calcular $A(\theta_i; x_i(t))$, donde $\hat{x}_i(t; \theta_i)$ tiene el sentido de una estimación de los datos “completos” pero inobservables basados en la observación $Y(t) = y(t)$, considerando los parámetros θ_L estimados previamente.

$$\begin{aligned} \hat{x}_i(t; \hat{\theta}^i) &= E\{X_i(t) | y(t); \hat{\theta}^i\} = s(t; \hat{\theta}^i) + [y(t) - \sum_{l \neq i}^L s(t; \hat{\theta}_l^i)] = \\ &= y(t) - \sum_{l \neq i}^L s(t; \hat{\theta}_l^i) \end{aligned} \tag{2.64}$$

Finalmente tendremos que maximizar la función integral z que se implementa siguiendo la siguiente estructura

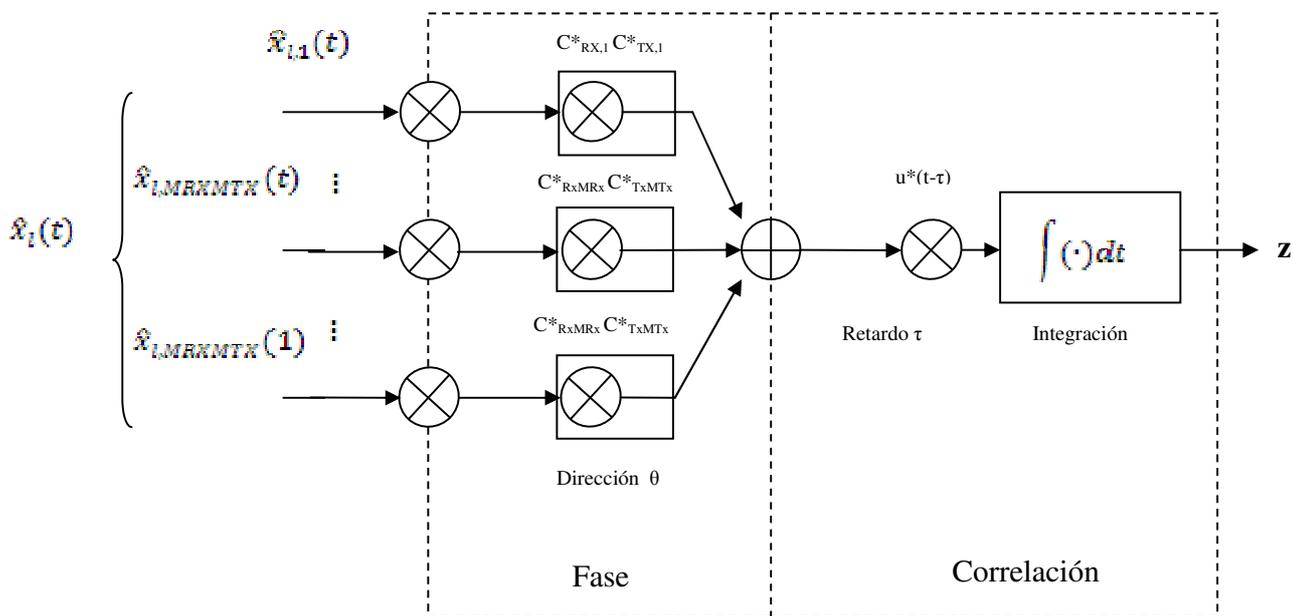


Figura 2.4.3- Implementación de la función z .



La ecuación de la función integral z se define como:

$$z(\bar{\theta}_i; \hat{x}_i) = \sum_{i=1}^I \int_0^T u^*(t - \tau_i) c_{RX}(\Omega_{RX,i})^H \hat{x}_i(t + t_{i,n,m}) c_{TX}(\Omega_{TX,i})^* dt \quad (2.65)$$

La combinación del paso E y de los M recibe el nombre de paso de iteración de SAGE. Un ciclo iterativo de SAGE consiste en L pasos de iteración de SAGE donde el vector de parámetros de cada onda se estima una vez [13].



Capítulo 3. Adaptación de Algoritmos

En este capítulo se pretende explicar los pasos que han sido necesarios para llevar a cabo la adaptación de algoritmos, exponiendo el funcionamiento de las funciones desarrolladas que componen los algoritmos, cuya implementación se lleva a cabo con el lenguaje de programación de alto nivel Matlab [12], en su versión 6.5.

3.1 Estructura General

En este apartado se explicará, de forma general, la estructura de implementación de cada uno de los algoritmos haciendo hincapié en aquellos puntos que ha sido necesario adaptar para conseguir un correcto funcionamiento de los mismos.

Para una más fácil comprensión distinguiremos el entorno real del entorno simulado puesto que las necesidades de adaptación han sido diferentes en cada caso.

El entorno simulado consiste en la simulación de unas condiciones ideales del canal radio, para una posterior estimación de los parámetros mediante la aplicación de los algoritmos (Capítulo 4).

En el caso del entorno real se dispone de unas medidas reales tomadas por el grupo de investigación TELICE a partir de las cuales, y mediante la aplicación de los diferentes algoritmos, se obtendrán los parámetros DOD, DOA y Retardo (Capítulo 5).

En el caso real, las medidas han sido tomadas en el dominio de la frecuencia: N frecuencias equiespaciadas en un ancho de banda total BW. De tal forma que ha sido necesario transformar al dominio del tiempo para poder estimar el retardo de las diferentes contribuciones. Tal señal al transformarla al dominio temporal resulta en una señal en el dominio del tiempo en una ventana temporal de duración:

$$(N^{\circ} \text{ de frecuencias Seleccionadas})/BW \quad (3.1)$$

Por tanto el tiempo de observación corresponde a dicho valor, si lo transformamos para obtener distancias nos quedará:

$$D_{TX-RX} = (N^{\circ} \text{ de frecuencias Seleccionadas}/BW) * 3e8 \text{ (m)} \quad (3.2)$$

Por lo que los valores de retardo obtenidos tendrán que ser múltiplos de este valor. En nuestro caso, el Número de frecuencias seleccionadas es de 73 y el valor del ancho de banda es de 200MHz con lo que observamos que habrá que tener en cuenta que el retardo serán múltiplos del valor 365 ns, o lo que es lo mismo habrá que medirlo en ventanas de 109.5 m.



3.1.1. Beamforming

La versión de Beamforming utilizada tiene la siguiente estructura de funcionamiento:

- En el caso del entorno simulado:
 - Generación de la señal: composición de la señal recibida mediante el uso de un generador de señales. La señal generada es una señal exponencial en el dominio del tiempo de dimensiones $[n_{Tx} * n_{Rx} \times \text{numMuestras}]$.
 - Se aplica el algoritmo Beamforming para la estimación de parámetros.
 - Presentación de resultados: representación gráfica de los parámetros estimados (DOA/DOD, retardo/DOD/amplitud, retardo/DOA/amplitud).
- En el caso del entorno real:
 - Se cargan los datos a partir de archivos en los que se encuentran las medidas tomadas por el grupo de investigación TELICE (Télécommunications, Interférences et Compatibilité Electromagnétique) de la Universidad de Lille (Francia). Las medidas están en el dominio de la frecuencia, por tanto, para conseguir el correcto funcionamiento del algoritmo habrá que pasarla al dominio del tiempo (aplicando la transformada inversa de Fourier).
 - Se divide la señal recibida en intervalos de frecuencia de ancho de banda de 200MHz, para trabajar con una señal de una banda más reducida que la original de la señal. Además, también será necesario el adaptar las dimensiones de la señal a las de trabajo del algoritmo $[n_{Tx} * n_{Rx} \times \text{numMuestras}]$.
 - Se aplica el algoritmo Beamforming para la estimación de parámetros.
 - Presentación de resultados: representación gráfica de los parámetros estimados.

3.1.2. MUSIC

La versión de MUSIC utilizada tiene la siguiente estructura de funcionamiento:

- En el caso del entorno simulado:
 - Generación de la señal: composición de la señal recibida mediante el uso de un generador de señales. La señal generada es una señal exponencial en el dominio del tiempo de dimensiones $[n_{Tx} * n_{Rx} \times \text{numMuestras}]$.
 - Se aplica el algoritmo MUSIC para la estimación de parámetros.
 - Presentación de resultados: representación gráfica de los parámetros estimados (DOA/DOD, retardo/DOD/amplitud, retardo/DOA/amplitud).
- En el caso del entorno real:
 - Se cargan los datos a partir de archivos en los que se encuentran las medidas tomadas por el grupo de investigación TELICE. Las medidas están en el dominio de la frecuencia, por tanto, para conseguir el correcto funcionamiento



del algoritmo habrá que pasarla al dominio del tiempo (aplicando la transformada inversa de Fourier).

- Se divide la señal recibida en intervalos de frecuencia de ancho de banda de 200MHz, para trabajar con una señal de banda una banda más reducida que la original de la señal. Además, también será necesario el adaptar las dimensiones de la señal a las de trabajo del algoritmo $[nTx*nRx \times \text{numMuestras}]$.
- Se aplica el algoritmo MUSIC para la estimación de parámetros.
- Presentación de resultados: representación gráfica de los parámetros estimados.

3.1.3 2-D Unitary ESPRIT

La versión de ESPRIT utilizada tiene la siguiente estructura de funcionamiento:

- En el caso del entorno simulado:
 - Generación de la señal: composición de la señal recibida mediante el uso de un generador de señales. La señal generada es una señal exponencial en el dominio del tiempo de dimensiones $[nTx*nRx \times \text{numMuestras}]$.
 - Se aplica el algoritmo 2-D Unitary ESPRIT para la estimación de parámetros.
 - Presentación de resultados: representación gráfica de los parámetros estimados (DOA/DOD, retardo/DOD/amplitud, retardo/DOA/amplitud).
- En el caso del entorno real:
 - Se cargan los datos a partir de archivos en los que se encuentran las medidas tomadas por el del grupo de investigación TELICE. Las medidas están en el dominio de la frecuencia, por tanto, para conseguir el correcto funcionamiento del algoritmo habrá que pasarla al dominio del tiempo (aplicando la transformada inversa de Fourier).
 - Se divide la señal recibida en intervalos de frecuencia de ancho de banda de 200MHz, para trabajar con una señal de banda una banda más reducida que la original de la señal.
 - Generación de la señal: composición de la señal recibida. Además, también será necesario el adaptar las dimensiones de la señal a las de trabajo del algoritmo $[nTx*nRx \times \text{numMuestras}]$.
 - Se aplica el algoritmo 2-D Unitary ESPRIT para la estimación de parámetros.
 - Presentación de resultados: representación gráfica de los parámetros estimados.

3.1.4 SAGE

La versión de SAGE realizada lleva cabo la estimación del ángulo de llegada (DOA), del ángulo de salida (DOD) y del retardo.

La estructura de funcionamiento que sigue el programa es la siguiente:

- En el caso del entorno simulado:



- Generación de la señal: composición de la señal recibida en el dominio de la frecuencia de dimensiones $[nTx \times nRx \times \text{numMuestras}]$.
 - Se aplica el algoritmo SAGE para la estimación de parámetros.
 - Presentación de resultados: representación gráfica de los parámetros estimados. En este algoritmo se obtiene un solo valor para cada uno de los parámetros estimados, por tanto, y para una mejor visualización y comparación con los resultados del resto de algoritmos y con la señal generada, se llevará a cabo una reconstrucción de la señal exponencial.
- En el caso del entorno real:
 - Se cargan los datos a partir de archivos en los que se encuentran las medidas tomadas por el grupo de investigación TELICE.
 - Se divide la señal recibida en intervalos de frecuencia de ancho de banda de 200MHz, para trabajar con una señal de banda una banda más reducida que la original de la señal.
 - Se aplica el algoritmo SAGE para la estimación de parámetros.
 - Presentación de resultados: representación gráfica de los parámetros estimados. En este algoritmo se obtiene un solo valor para cada uno de los parámetros estimados, por tanto, y para una mejor visualización y comparación con los resultados del resto de algoritmos y con las medidas de la señal, se llevará a cabo una reconstrucción de la señal exponencial.

3.2 Funciones

Al inicio de cada una de las funciones se ha introducido texto comentado con una pequeña explicación de las mismas para que cuando se introduzca desde el workspace de Matlab el comando “help nombreFunción” aparezca ésta a modo explicativo.

Para la realización de este proyecto se ha partido de códigos ya implementados, sin embargo, han sido necesarias modificaciones para adaptar los algoritmos (explicadas en el apartado anterior).

El código de ejecución del entorno real ha tenido que ser creado completamente, así como la representación de resultados.

Para el caso del entorno simulado, se han tenido que realizar modificaciones en el generador de señales para poder incluir la generación de una señal de dimensiones $[NTx \times NRx \times \text{NumFrecuencias}]$ necesaria para el funcionamiento del algoritmo SAGE. Además se han creado las funciones de representación de cada uno de los algoritmos y ha sido necesaria la adaptación del código del que se disponía del interfaz para la introducción del algoritmo SAGE en el mismo.



3.2.1 Funciones Comunes

3.2.1.1 *GENERA_H.m*

Esta función ya estaba generada anteriormente en [11], sin embargo se han realizado las modificaciones necesarias para generar la señal recibida en el dominio de la frecuencia con las dimensiones necesarias para el correcto funcionamiento del algoritmo SAGE. Sólo se hará uso de ella en el entorno simulado.

3.2.1.2 *pantalla_simulado.m*

Este script contiene todas las instrucciones para poder realizar todas las operaciones disponibles en la interfaz correspondiente al entorno simulado. Inicialmente existía una versión de este script en [11] aunque han sido necesarias muchas modificaciones entre las que se incluyen la incorporación del algoritmo Sage así como la adaptación de la interfaz a las nuevas funcionalidades.

3.2.1.3 *entornoReal.m*

Este script contiene todas las instrucciones para poder aplicar los diferentes algoritmos a las mediciones de las que se dispone permitiendo la estimación de parámetros de las mismas. Se ha llevado a cabo la creación íntegra de este script.

3.2.1.4 *vec.m*

Función utilizada en *GENERA_H.m* para calcular el vector *vec*, que es aquel que transforma una matriz en un vector apilando sus columnas. Esta función fue creada en [11].

3.2.1.5 *Resolucion.m*

Función utilizada en el entorno simulado para solicitar la introducción de la resolución que se desea aplicar en los algoritmos Beamforming o MUSIC. Esta función fue creada en [11].

3.2.2 Beamforming

3.2.2.1 *BEAMFORMING.m*

Función utilizada tanto en el entorno simulado como en el entorno real para la estimación de parámetros por medio del algoritmo Beamforming. Inicialmente se creó en [11] pero han sido necesarias ciertas adaptaciones como la modificación de la separación entre sensores.

3.2.2.2 *DibujarBeamforming_MUSIC.m*

Función utilizada para la representación gráfica de los resultados obtenidos tras la aplicación del algoritmo. Es utilizada también para la representación de los resultados obtenidos al aplicar el algoritmo MUSIC. Se ha llevado a cabo la creación íntegra de este script.



3.2.2.3 simplificar_Beam_Music.m

Función utilizada para simplificar los resultados obtenidos por los algoritmos Beamforming y MUSIC, para poder hacer una comparación con los resultados obtenidos al aplicar el resto de algoritmos. Este función ha sido generada íntegramente.

3.2.3 MUSIC

3.2.3.1 MUSIC.m

Función utilizada tanto en el entorno simulado como en el entorno real para la estimación de parámetros por medio del algoritmo MUSIC. Al igual que en el caso de Beamforming.m se creó en [11] pero han sido necesarias ciertas adaptaciones como la modificación de la separación entre sensores.

3.2.4 2-D ESPRIT

3.2.4.1 ESPRIT_unitary.m

Función utilizada tanto en el entorno simulado como en el entorno real para la estimación de parámetros por medio del algoritmo 2D Unitary ESPRIT. Al igual que en el caso de Beamforming.m y el Music.m se creó en [11] pero han sido necesarias ciertas adaptaciones como la modificación de la separación entre sensores.

3.2.4.2 Matriz_TT.m

Esta función genera la matriz antidiagonal unitaria de las dimensiones introducidas. Funcion creada en [11].

3.2.4.3 MatrizQ.m

Función utilizada para el cálculo de la matriz Q, que es aquella que se define como la matriz unitaria TT-reales. Funcion creada en [11].

3.2.4.4 DibujarEsprit.m

Función utilizada para la representación gráfica de los resultados obtenidos tras la aplicación del algoritmo. Este script ha sido creado de forma íntegra.

3.2.4.5 simplificar_Esprit.m

Función utilizada para simplificar los resultados obtenidos por el algoritmo 2D Unitary Esprit, para poder hacer una comparación con los resultados obtenidos al aplicar el resto de algoritmos. Este script ha sido creado de forma íntegra.



3.2.5 SAGE

3.2.5.1 model_funcion.m

Calcula el modelo de la señal con los parámetros estimados.

3.2.5.2 cost_funcion.m

Esta función lleva a cabo la evaluación de la función de costes y su maximización mediante el cálculo de la integral discreta.

3.2.5.3 ic.m

Una vez calculado el modelo de señal de la función anterior se eliminan componentes ya estimadas para llevar a cabo la estimación de la siguiente componente.

3.2.5.4 redim.m

Función que redimensiona la una matriz de entrada devolviendo a la salida una matriz con las dimensiones especificadas.

3.2.5.5 SAGE.m

Función utilizada tanto en el entorno simulado como en el entorno real para la estimación de parámetros por medio del algoritmo SAGE.

3.2.5.6 DibujarSAGE.m

Función utilizada para la representación gráfica de los resultados obtenidos tras la aplicación del algoritmo. Esta función ha sido generada de forma íntegra.

3.3 Herramienta

En este apartado se va a presentar la herramienta realizada en MatLab con el objetivo de implementar los algoritmos Beamforming convencional, MUSIC, 2-D Unitary ESPRIT y SAGE comentados en el capítulo anterior y observar gráficamente los resultados que se obtienen con cada uno de ellos.

La herramienta sólo se ha implementado para el caso del entorno simulado mientras que para el caso del entorno real se ha creado un script de ejecución en Matlab fácilmente modificable.

El lenguaje de programación escogido para programar nuestra herramienta ha sido el proporcionado por MatLab. MatLab es un programa para realizar cálculos numéricos con *vectores* y *matrices*. Como caso particular puede también trabajar con números escalares, tanto reales como complejos. Una de las mayores ventajas que presenta esta herramienta, además de su gran potencia de cálculo, es la variedad de opciones de representación que

posee permitiendo realizar una amplia variedad de *gráficos* en dos y tres dimensiones. Además el lenguaje de programación propio de Matlab a pesar de ser más limitado que otros lenguajes resulta muy sencillo de utilizar. También posee una herramienta para crear interfaces gráficas, GUIDE, lo que nos ha inclinado a Matlab como la herramienta de realización del proyecto.

La interfaz de Entorno Simulado realizada permite crear un sistema MIMO completo controlando todos los parámetros del mismo. El objetivo final es el de poder comprobar el comportamiento teórico de los algoritmos Beamforming convencional, MUSIC, 2-D Unitary ESPRIT y SAGE, y para ello simula un sistema MIMO completo permitiéndonos seleccionar desde el número de elementos de los arrays hasta el ruido del canal. Para ello se ha aprovechado parte del código que se realizó en [11] para el desarrollo de esta interfaz aplicando todos aquellos cambios que se han considerado necesarios, entre los que destaca la introducción del algoritmo SAGE en la misma.

La herramienta presenta el siguiente aspecto:

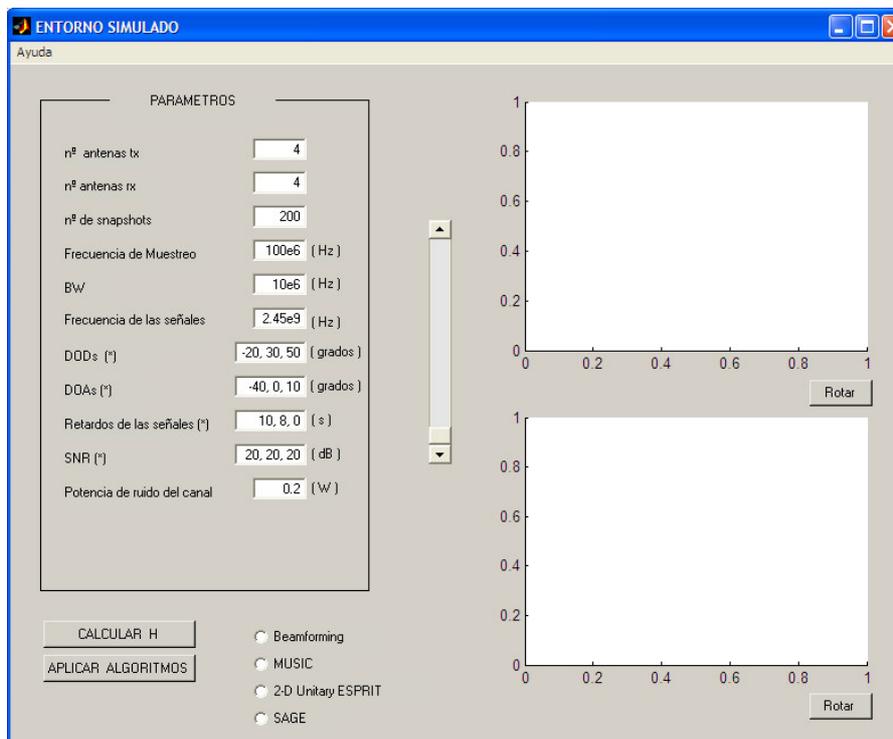
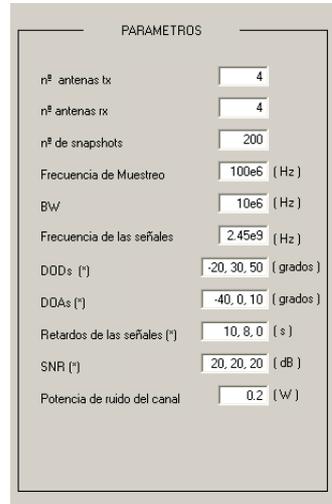


Figura 3.1- Pantalla de Entorno Simulado

Podemos distinguir tres áreas fundamentales en función de su funcionalidad.

La primera ellas es la correspondiente a la introducción de parámetros:



PARAMETROS	
n° antenas tx	4
n° antenas rx	4
n° de snapshots	200
Frecuencia de Muestreo	100e6 (Hz)
BW	10e6 (Hz)
Frecuencia de las señales	2.45e9 (Hz)
DODs (*)	-20, 30, 50 (grados)
DOAs (*)	-40, 0, 10 (grados)
Retardos de las señales (*)	10, 8, 0 (s)
SNR (*)	20, 20, 20 (dB)
Potencia de ruido del canal	0.2 (W)

Figura 3.2 - Área de Introducción de Parámetros

Como se puede ver se permite modificar cada uno de los parámetros que constituyen un sistema MIMO:

- N° Antenas Tx - número de sensores que constituyen el array de transmisión.
- N° antenas Rx - número de sensores que constituyen el array de recepción.
- N° de snapshots - número de muestras o instantes de señal que vamos a tomar de las señales presentes en el canal
- Frecuencia de Muestreo – la frecuencia, en Hertzios a la que deseemos muestrear.
- BW – ancho de banda de la señal, el mismo para todas las señales, se introduce en ‘Hz’
- DODs (*) – los ángulos de salida de las señales que introducimos en el canal.
- DOAs (*) – los ángulos de llegada de las señales que introducimos en el canal.
- Retardos de las señales (*) – retardos (en segundos) que deseamos asignar a cada una de las señales que introducimos en el sistema
- SNR (*) – relación señal a ruido en dBs de las señales
- Potencia de ruido del canal – introducimos la potencia del ruido en watos.

Como podemos observar cuatro de los parámetros aparecen destacados por ‘(*)’, esto es debido a que estos parámetros introducen además el número de señales presentes en el canal y por ello deben coincidir en número de valores introducidos para que no se produzca ningún error de ejecución.

Se ha de tener en cuenta que la señal introducida en el canal no se puede modificar, ésta será en todos los casos una señal exponencial decreciente.



Una vez introducidos los parámetros pasaríamos a la segunda parte de la interfaz:



Figura 3.3 – Área de Cálculo de H y Aplicación de Algoritmos

Para comenzar es necesario pulsar el botón ‘Calcular H’, si no se produce ningún problema en el cálculo de la misma, se mostrará un elemento emergente que indicará que ‘La señal se ha generado Correctamente’ y en caso contrario mostrará un mensaje de error (‘No se ha generado H’). Una vez generada la señal se mostrará en el tercer espacio de la interfaz (espacio de representación de resultados) la señal generada:

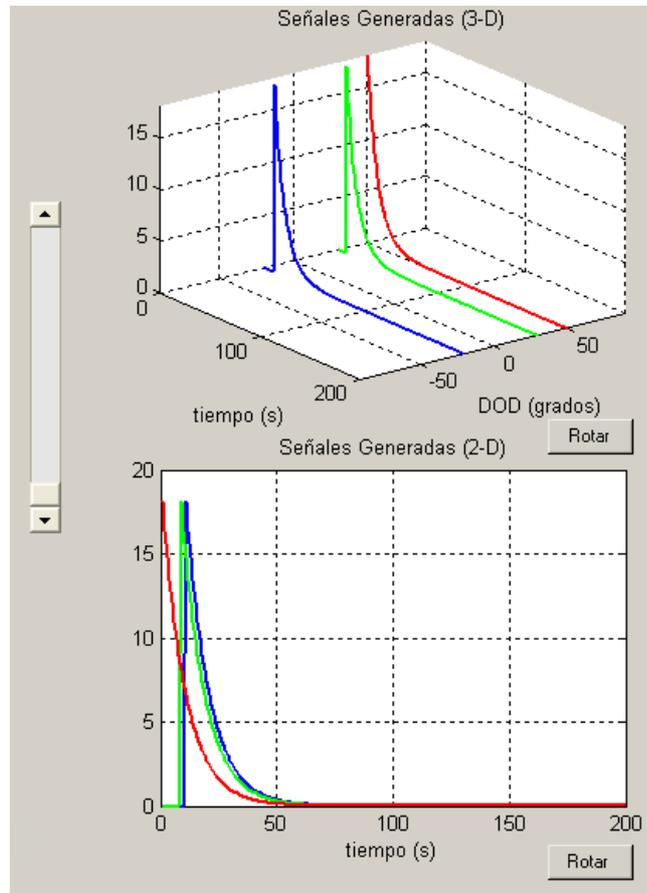


Figura 3.4 – Área de Representación gráfica

Finalmente se seleccionarán los algoritmos que se desea aplicar para la estimación de parámetros. En caso de que uno de los algoritmos seleccionados sea Beamforming o MUSIC, se solicitará que se introduzca la resolución que se desea que se utilice en el algoritmo.

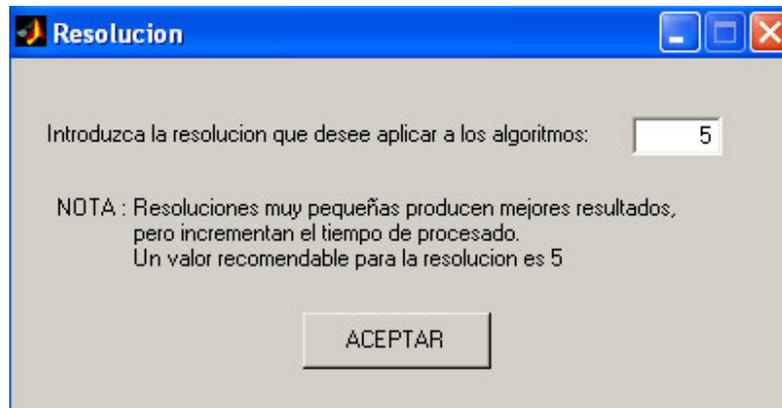


Figura 3.5 – Ventana de Resolución

Una vez estimados los parámetros se representarán los resultados en el tercer espacio de la interfaz, mencionado anteriormente, permitiendo la barra de desplazamiento la visualización de todas las gráficas en un desplazamiento de dos en dos gráficas.



Capítulo 4. Resultados Entorno Sintético

En este capítulo se analizarán los resultados que se obtienen con la estimación de parámetros de cada uno de los algoritmos sobre una señal generada mediante la simulación del canal y se llevará a cabo una comparación entre los resultados obtenidos con cada uno de ellos. Para ello se irán realizando modificaciones en las condiciones del canal para ver el comportamiento de cada uno de los algoritmos cuando se produce una modificación en las condiciones.

Para ello primero llevaremos a cabo un análisis teórico que nos permita confirmar si los resultados obtenidos son los correctos.

4.1 Extracción de Parámetros en Entorno Sintético

Como se explicó en el capítulo anterior (*Herramienta*), este entorno nos permite simular un sistema MIMO completo, con lo que vamos a poder observar y comparar el comportamiento de los distintos algoritmos variando características tanto de los transmisores y receptores como del canal.

Para ello, vamos a introducir en el canal tres contribuciones. Las contribuciones cumplen la condición de que dos de ellas se encuentran en ángulos muy próximos entre sí. De esta forma veremos la capacidad que poseen los algoritmos para detectarlas.

Variaremos también los siguientes parámetros de los sistemas:

- Número de snapshots.
- Número de antenas en los arrays de transmisión y recepción.
- La potencia de las señales.
- Resolución de los algoritmos

Para distinguir cada una de las simulaciones, al inicio de cada una de ellas se presentarán los parámetros introducidos en una tabla que presentará el siguiente aspecto:

Nº Antenas Tx	Nº Antenas Rx	Nº Snapshots	Frec. Muestreo (Hz)	BW (Hz)	Frec. Señal (Hz)
DOD (grados)	DOA (grados)	Retardo (seg.)	SNR (dB)	Potencia de Ruido (W)	Resolución

Tabla 4.1.1: Tabla de parámetros estándar

Se irán variando los valores de la simulación para comprobar el funcionamiento de los diferentes algoritmos ante las diferentes características del canal. Los parámetros que se



variarán para ver como afectan al funcionamiento de los diferentes algoritmos serán: el número de sensores en recepción y transmisión, el número de muestras de señal, la relación señal a ruido y la resolución.

La resolución es la habilidad de resolver dos fuentes muy próximas. Para los métodos basados en el espectro es sencillo de representar puesto que al obtener los picos del espectro si se obtienen menos picos de los esperados la resolución es inferior del valor necesario.

4.1.1 Primera simulación

Para comenzar se mostrará la tabla con los valores de la simulación para comparar la precisión de cada uno de los algoritmos en función de los resultados obtenidos en relación a las características introducidas:

Nº Antenas Tx	Nº Antenas Rx	Nº Snapshots	Frec. Muestreo (Hz)	BW (Hz)	Frec. Señal (Hz)
4	4	200	100e6	10e6	2.45e9
DOD (grados)	DOA (grados)	Retardo (seg.)	SNR (dB)	Potencia de Ruido (W)	Resolución
[-20,30,50]	[-40,0,10]	[10,8,0]	[20,20,20]	0.2	5

Tabla 4.1.2: Tabla de parámetros estándar con la información de la primera simulación.

La señal generada que se obtiene es:

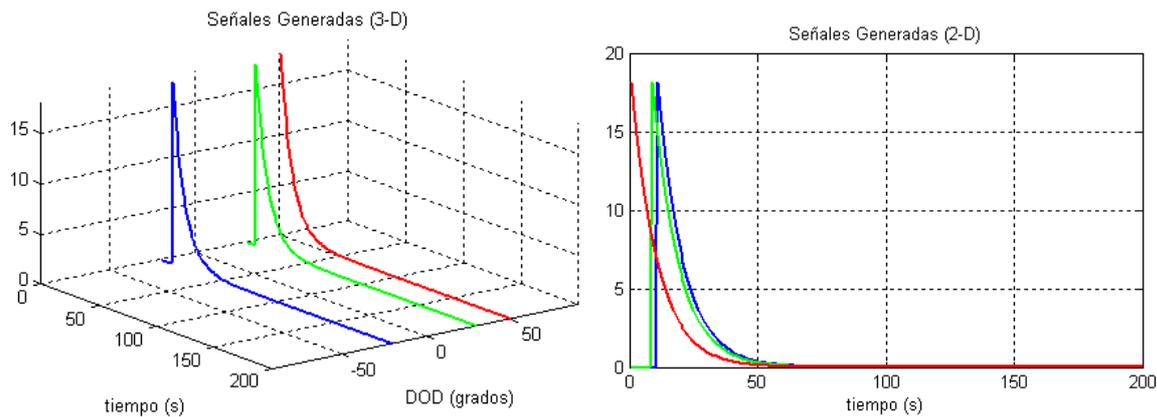


Figura 4.1.1: Señal generada a partir de los valores por defecto de los parámetros.

Tras mostrar la señal generada por el generador de señales se mostrarán los resultados obtenidos por cada uno de los algoritmos:

– Beamforming:

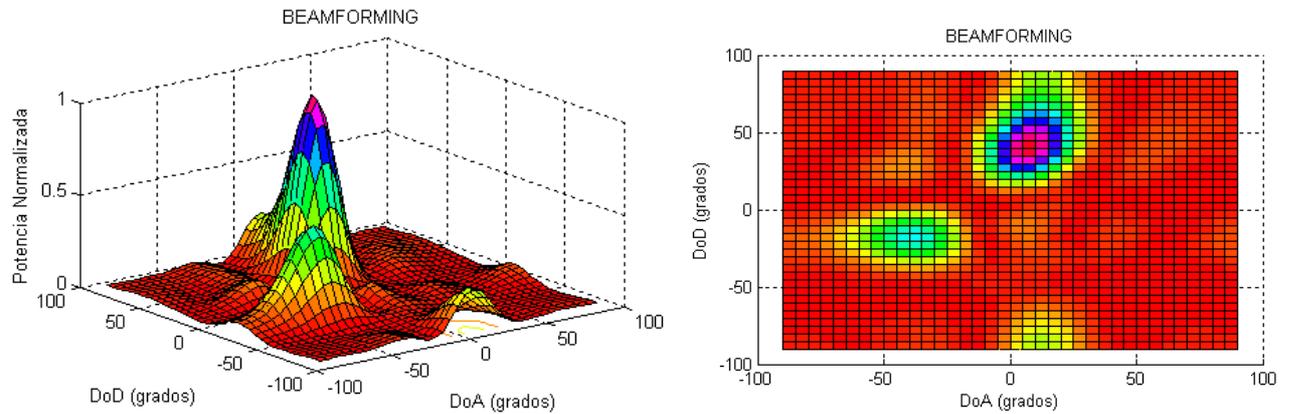


Figura 4.1.2: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

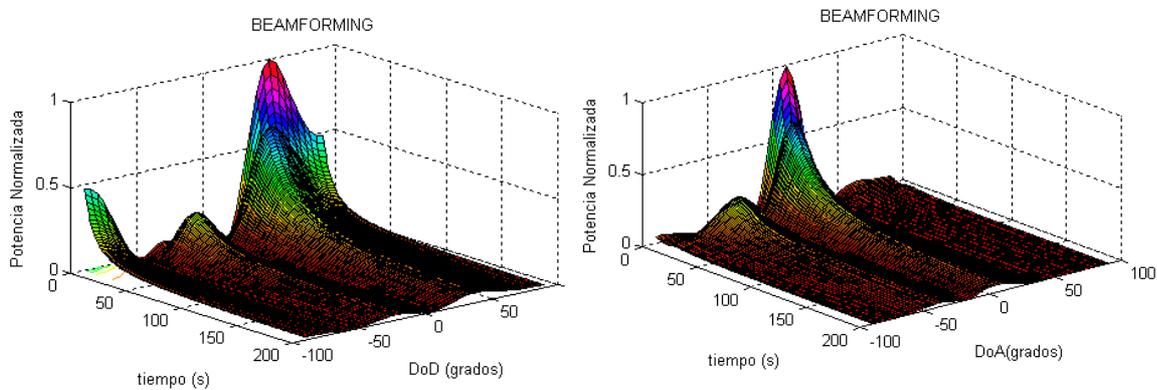


Figura 4.1.3: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 9.19 segundos.



– MUSIC:

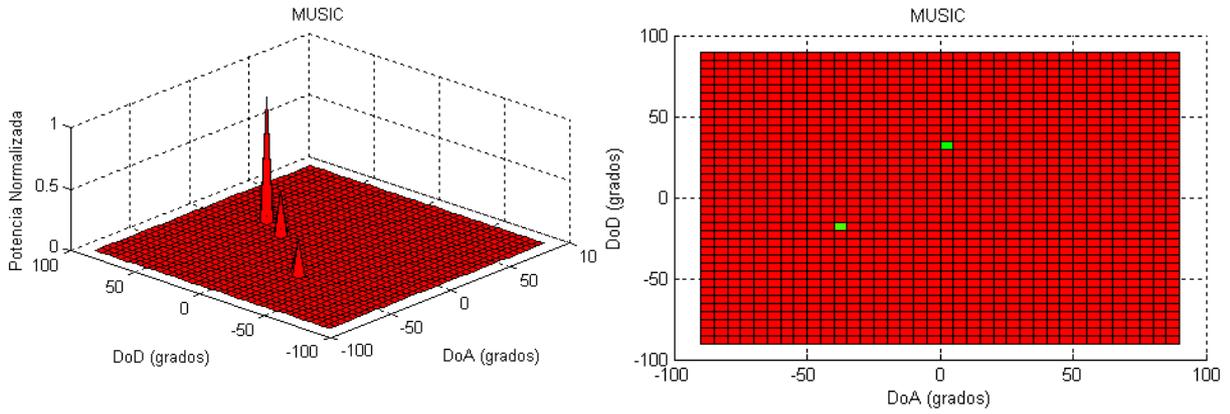


Figura 4.1.4: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

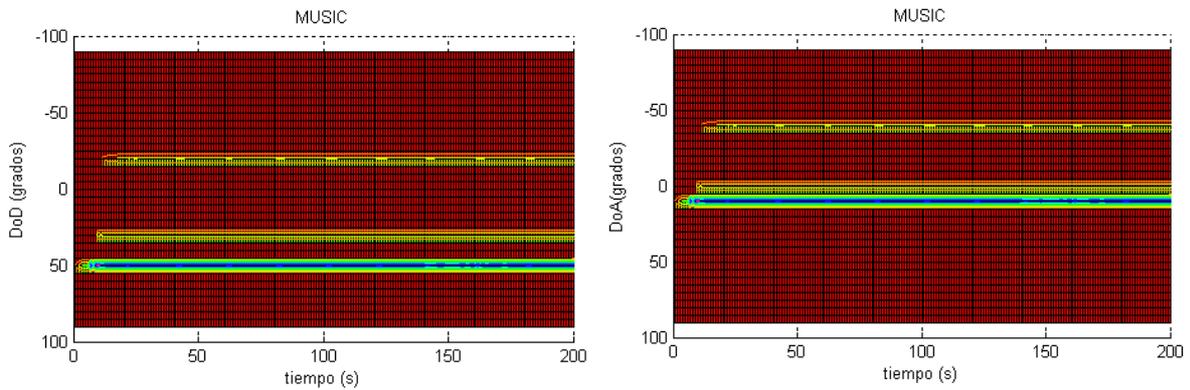


Figura 4.1.5: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 10.64 segundos.



– 2D Unitary ESPRIT:

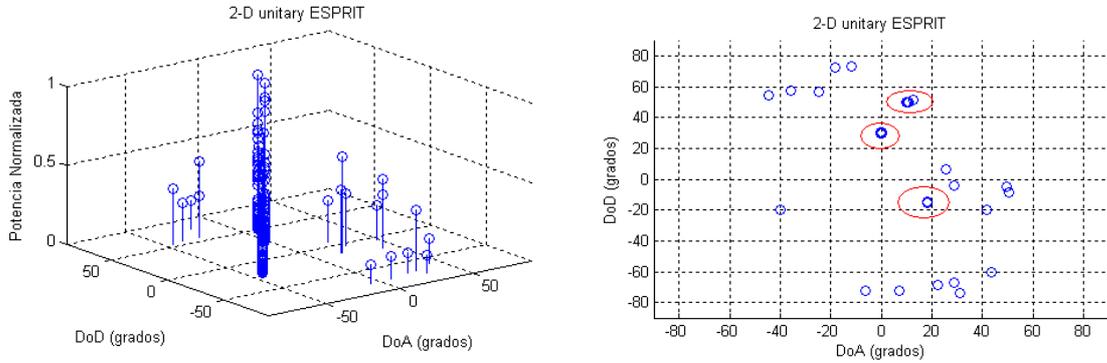


Figura 4.1.6: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

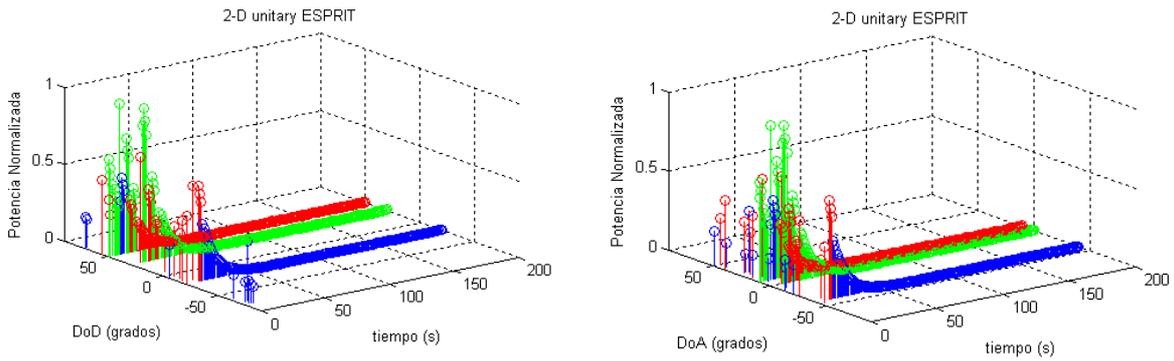


Figura 4.1.7: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 4.42 segundos.



– SAGE:

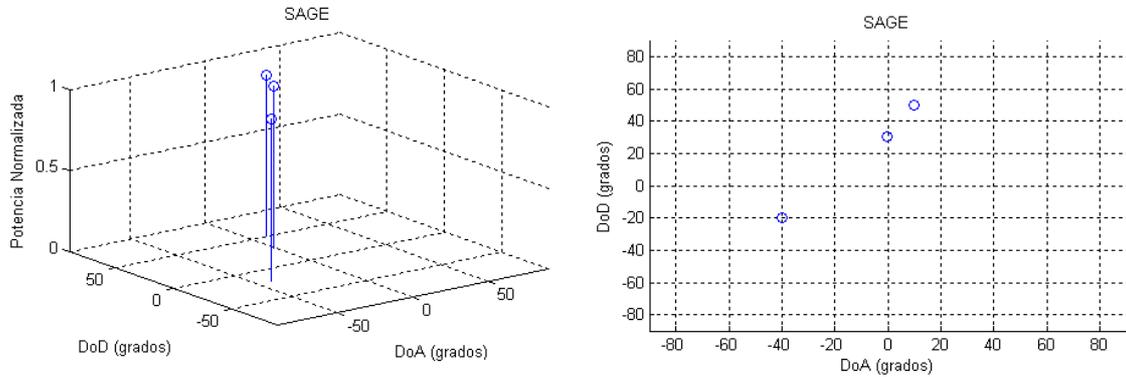


Figura 4.1.8: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

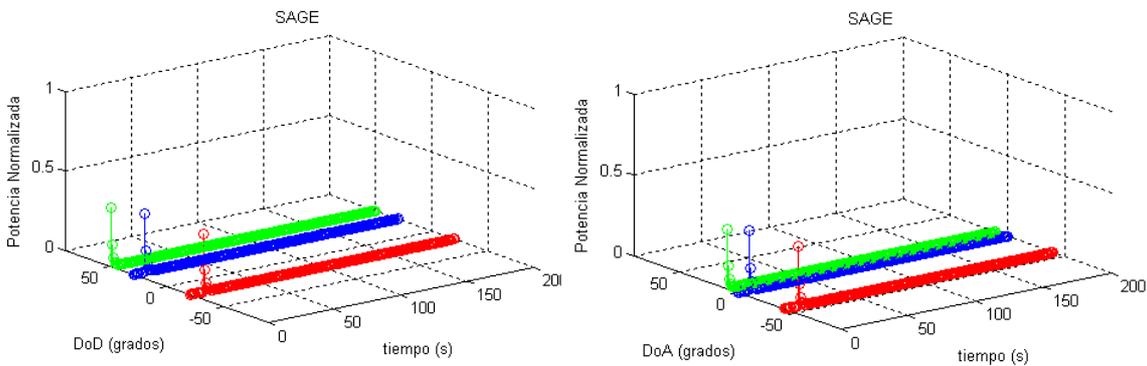


Figura 4.1.9: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 10.31 segundos.

Bajo las condiciones impuestas podemos observar que todos los algoritmos realizan una correcta estimación de parámetros. Sin embargo, y debido a la proximidad en el retardo de las contribuciones, el algoritmo Beamforming no es capaz de detectar las 3 contribuciones de la señal.



En cuanto a los tiempos de ejecución son muy similares entre los algoritmos aunque se puede apreciar que el tiempo de estimación del algoritmo 2D Unitary ESPRIT es aproximadamente la mitad del empleado por el resto de los algoritmos.

Algoritmo	DOD (°)	DOA (°)	Retardo (ns)	Tiempo Ejecución (s)
Beamforming	[-20,30,-90]	[-50,10,0]	[0,8,10]	9.19
MUSIC	[-20,30,50]	[-40,0,10]	[0,8,10]	10.64
2D Unitary ESPRIT	[-20,30,50]	[0,10,20]	[0,8,10]	4.42
SAGE	[-20,30,50]	[-40,0,10]	[0,8,10]	10.31

Tabla 4.1.3: Tabla de parámetros estimados tras la primera simulación.

4.1.2 Segunda Simulación: Aumentamos el número de sensores de los arrays

Para comenzar se mostrará la tabla con los valores de la simulación para comparar la precisión de cada uno de los algoritmos en función de los resultados obtenidos en relación a las características introducidas:

Nº Antenas Tx	Nº Antenas Rx	Nº Snapshots	Frec. Muestreo (Hz)	BW (Hz)	Frec. Señal (Hz)
12	12	200	100e6	10e6	2.45e9
DOD (grados)	DOA (grados)	Retardo (seg.)	SNR (dB)	Potencia de Ruido (W)	Resolución
[-20,30,50]	[-40,0,10]	[10,8,0]	[20,20,20]	0.2	5

Tabla 4.1.4: Tabla de parámetros estándar con la información de la segunda simulación.

La señal generada que se obtiene es:

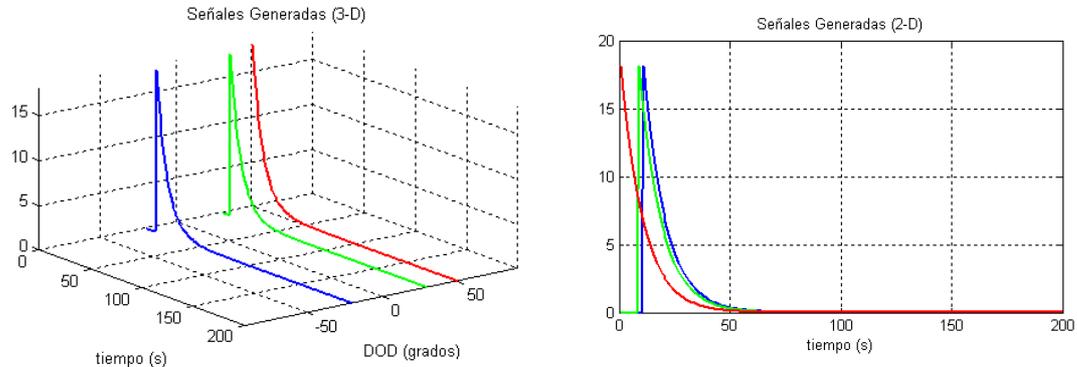


Figura 4.1.10: Señal generada al aumentar el número de sensores de los valores por defecto

Tras mostrar la señal generada por el generador de señales se mostrarán los resultados obtenidos por cada uno de los algoritmos:

– Beamforming:

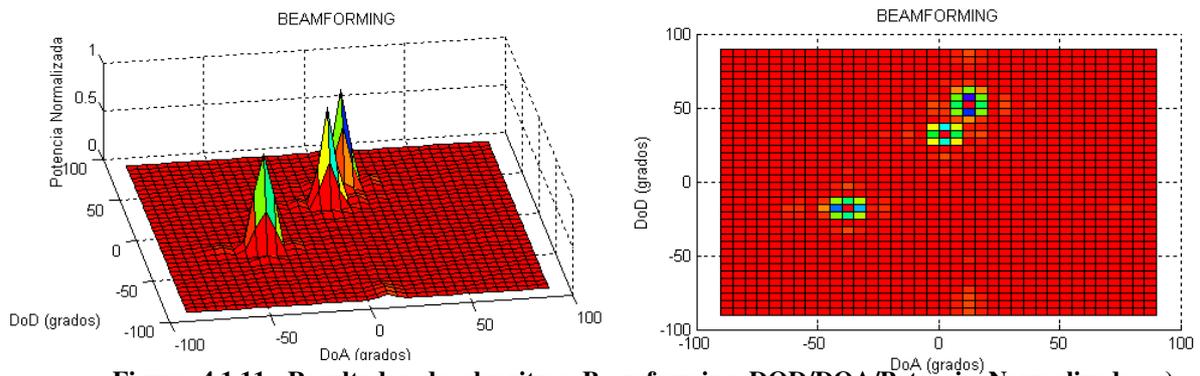


Figura 4.1.11: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

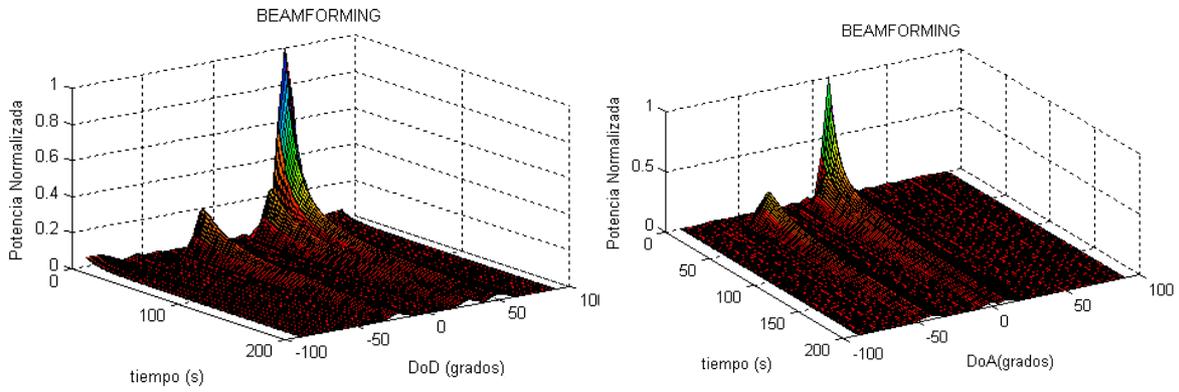


Figura 4.1.12: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 53.42 segundos.

— MUSIC:

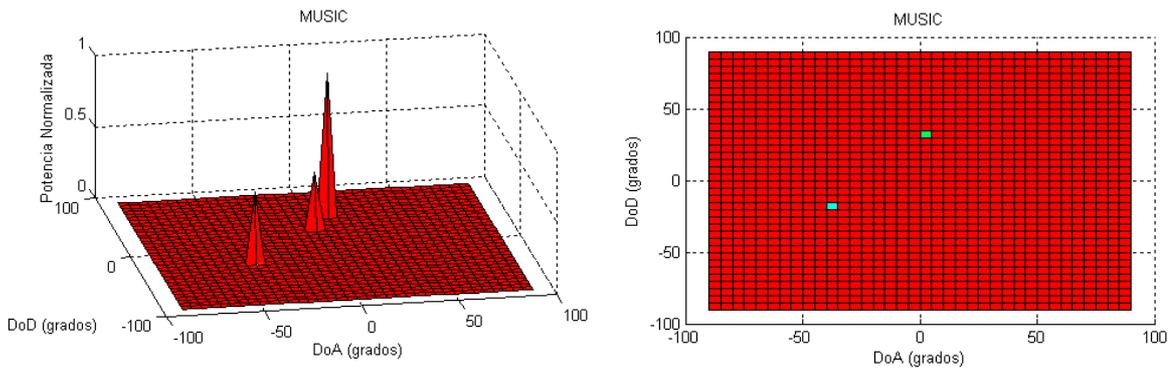


Figura 4.1.13: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

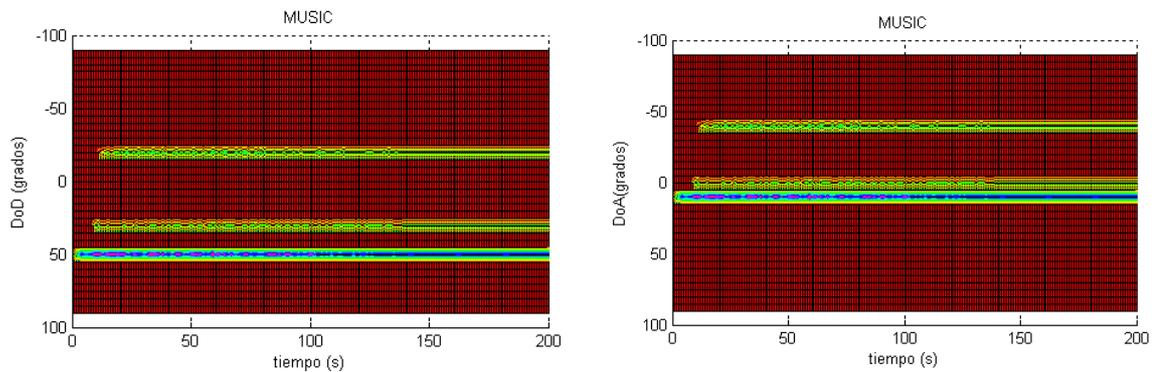


Figura 4.1.14: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 180.66 segundos.

– 2-D Unitary ESPRIT:

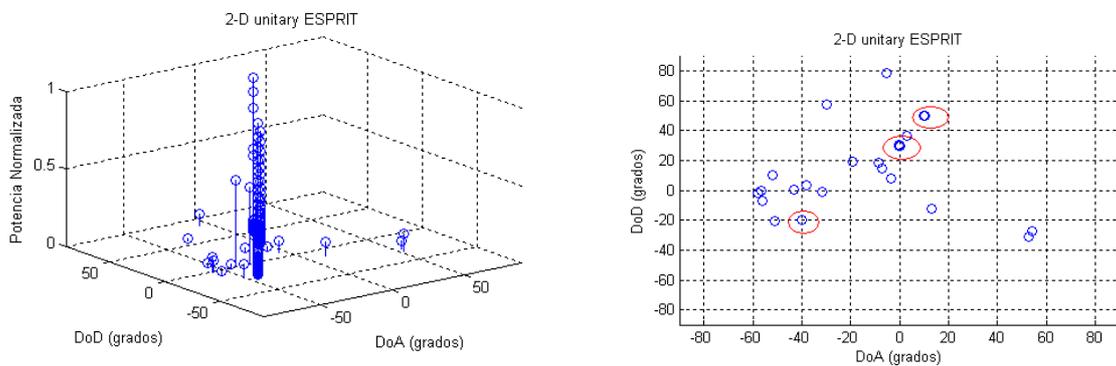


Figura 4.1.15: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

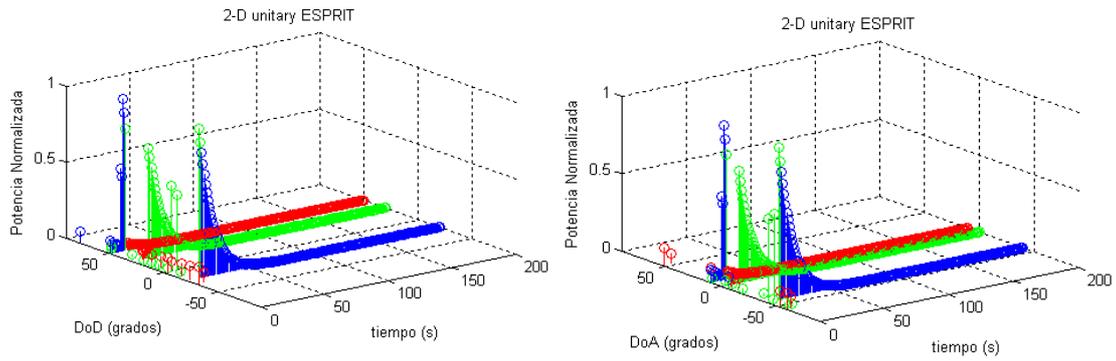


Figura 4.1.16: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 71.86 segundos.

– SAGE:

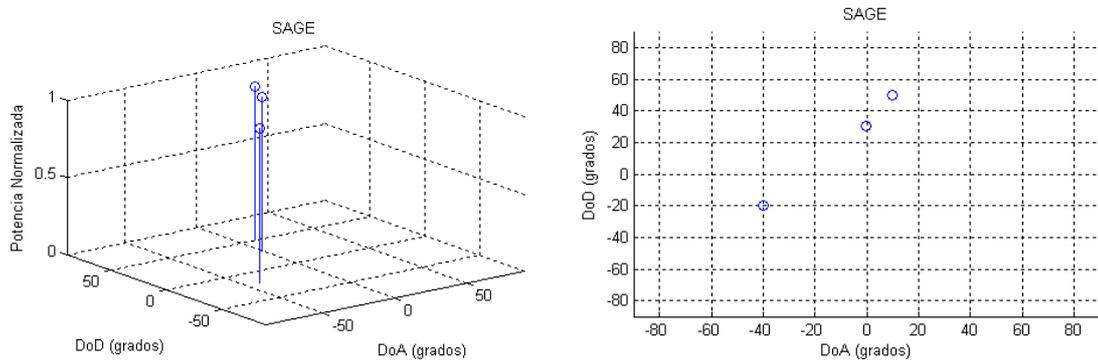


Figura 4.1.17: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

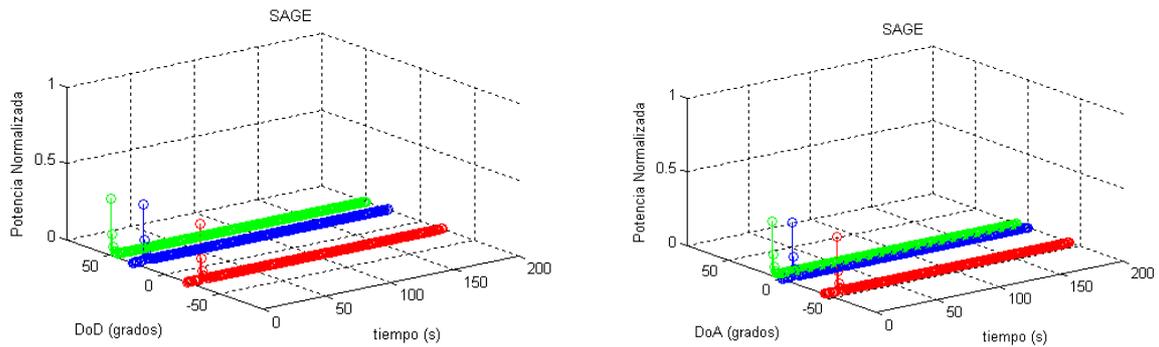


Figura 4.1.18: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 55.39 segundos.

Al aumentar el número de sensores vemos que, en caso de Beamforming, permite detectar las tres contribuciones. En el caso del resto de algoritmos no se aprecia ningún cambio notable ya que anteriormente los parámetros estimados eran correctos.

Algoritmo	DOD (°)	DOA (°)	Retardo (ns)	Tiempo Ejecución (s)
Beamforming	[-20,30,50]	[-40,0,10]	[0,8,10]	53.42
MUSIC	[-20,30,50]	[-40,0,10]	[0,8,10]	180.66
2D Unitary ESPRIT	[-20,30,50]	[-40,0,10]	[0,8,10]	71.86
SAGE	[-20,30,50]	[-40,0,10]	[0,8,10]	55.39

Tabla 4.1.5: Tabla de parámetros estimados tras la segunda simulación.



4.1.3 Tercera Simulación: Aumentamos el número de snapshots.

Para comenzar se mostrará la tabla con los valores de la simulación para comparar la precisión de cada uno de los algoritmos en función de los resultados obtenidos en relación a las características introducidas:

Nº Antenas Tx	Nº Antenas Rx	Nº Snapshots	Frec. Muestreo (Hz)	BW (Hz)	Frec. Señal (Hz)
4	4	1000	100e6	10e6	2.45e9
DOD (grados)	DOA (grados)	Retardo (seg.)	SNR (dB)	Potencia de Ruido (W)	Resolución
[-20,30,50]	[-40,0,10]	[10,8,0]	[20,20,20]	0.2	5

Tabla 4.1.6: Tabla de parámetros estándar con la información de la tercera simulación

La señal generada que se obtiene es:

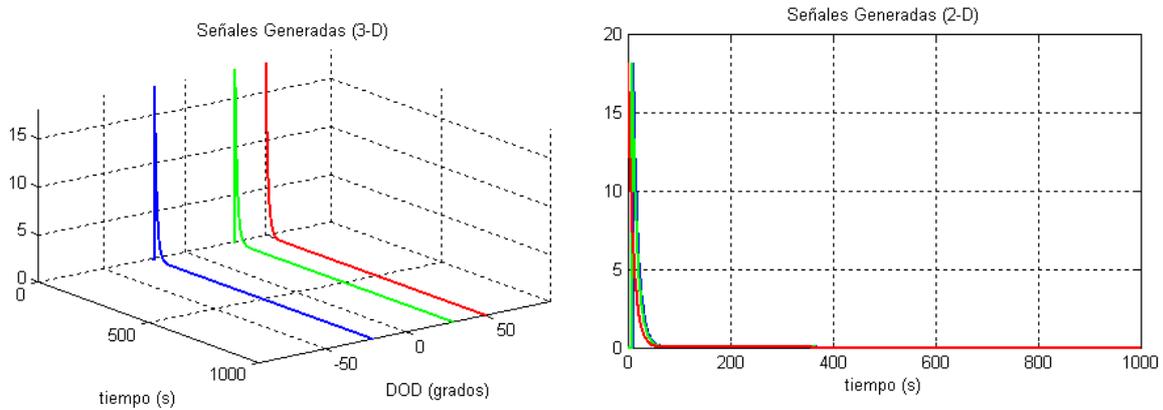


Figura 4.1.19: Señal generada tras el aumento del número de snapshots respecto de los valores por defecto de los parámetros

Tras mostrar la señal generada por el generador de señales se mostrarán los resultados obtenidos por cada uno de los algoritmos:

– Beamforming:

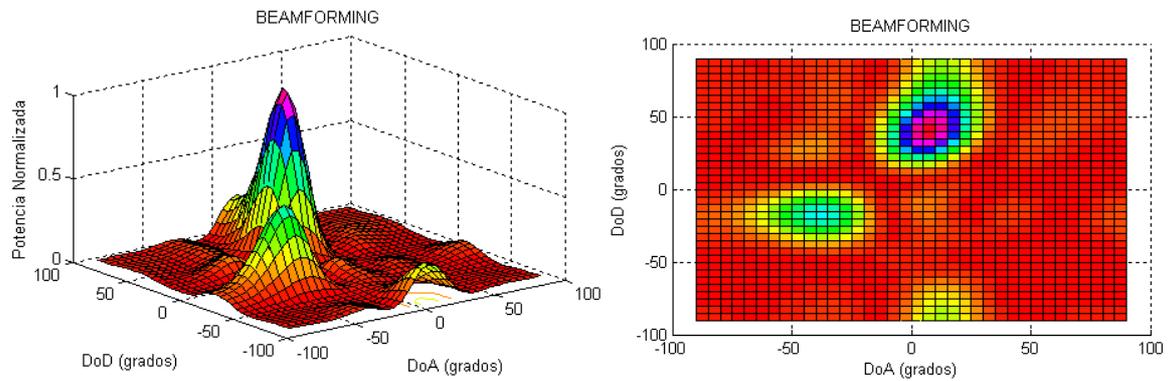


Figura 4.1.20: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

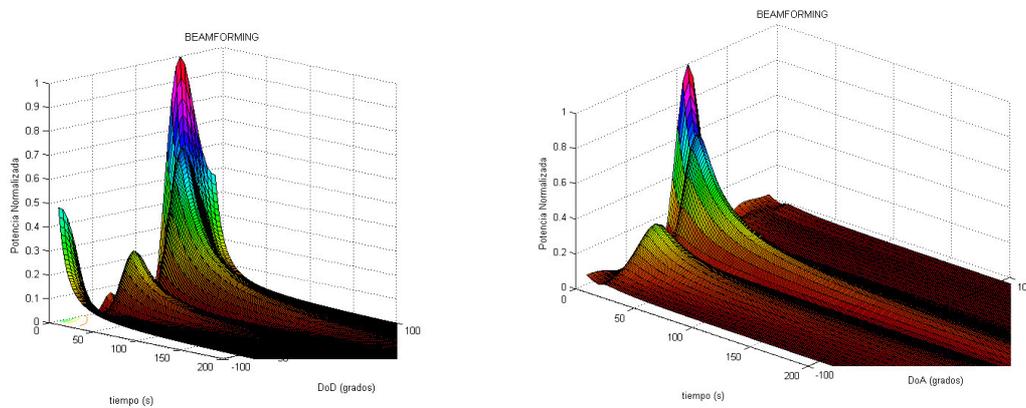


Figura 4.1.21: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 49.52 segundos.

– MUSIC:

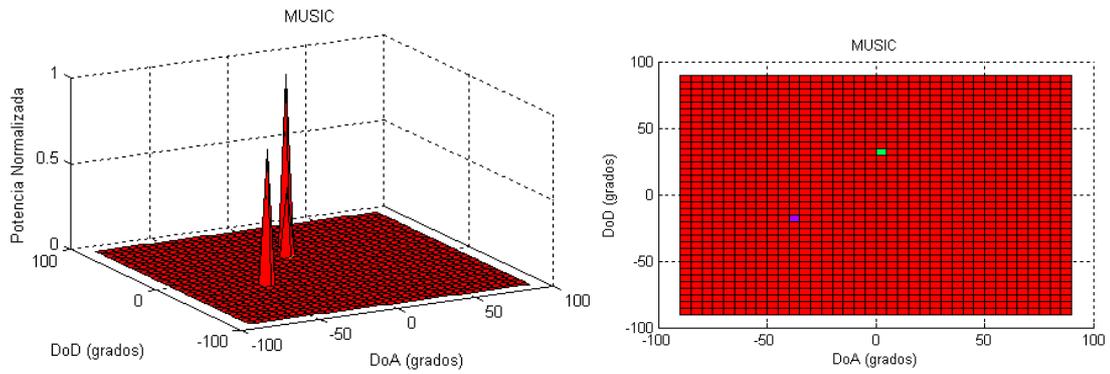


Figura 4.1.22: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

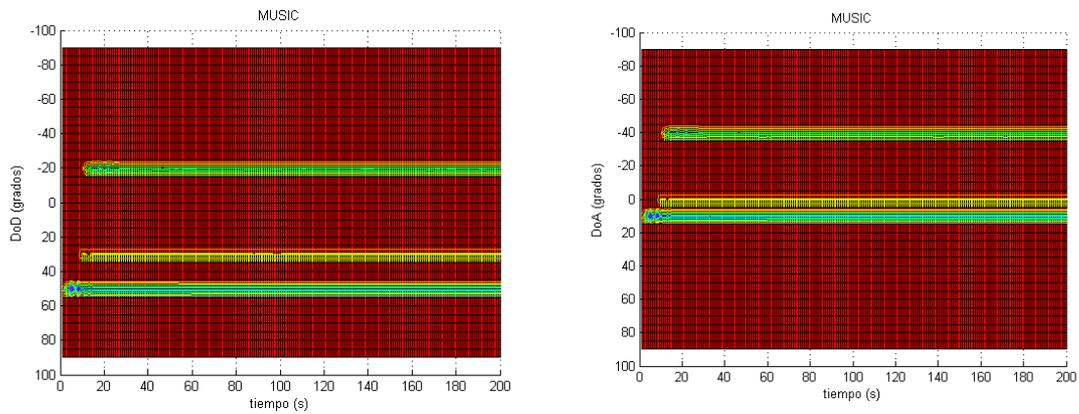


Figura 4.1.23: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 57.11 segundos.



– 2-D Unitary ESPRIT:

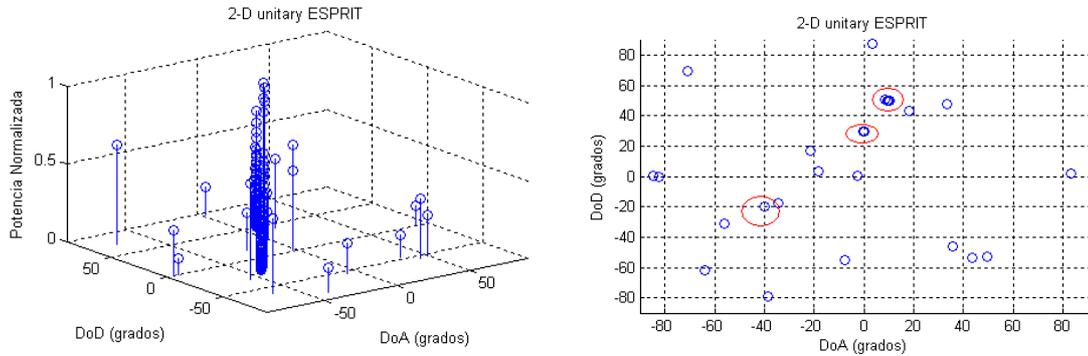


Figura 4.1.24: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

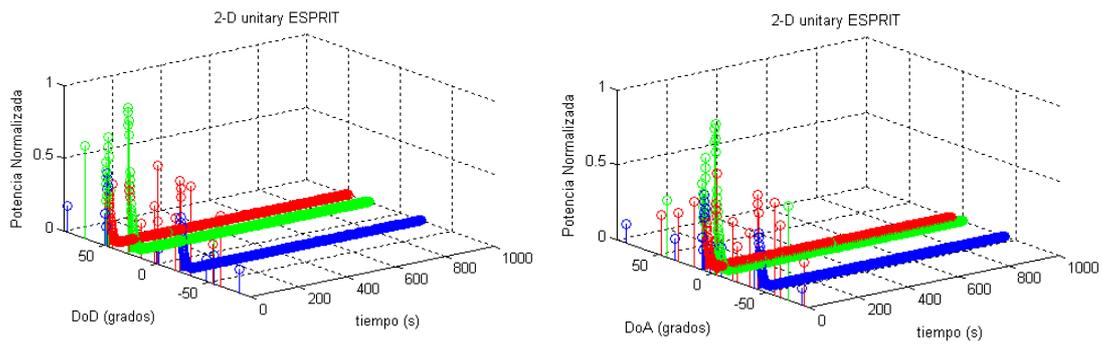


Figura 4.1.25: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 510.06 segundos.



– SAGE:

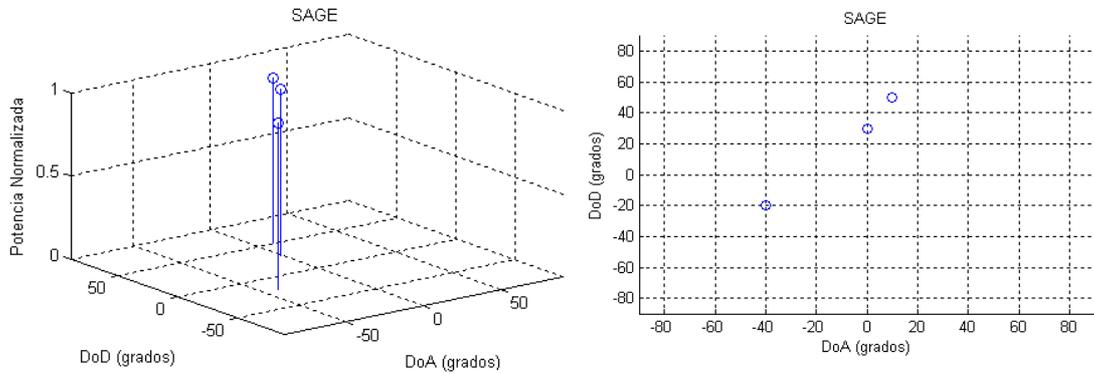


Figura 4.1.26: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

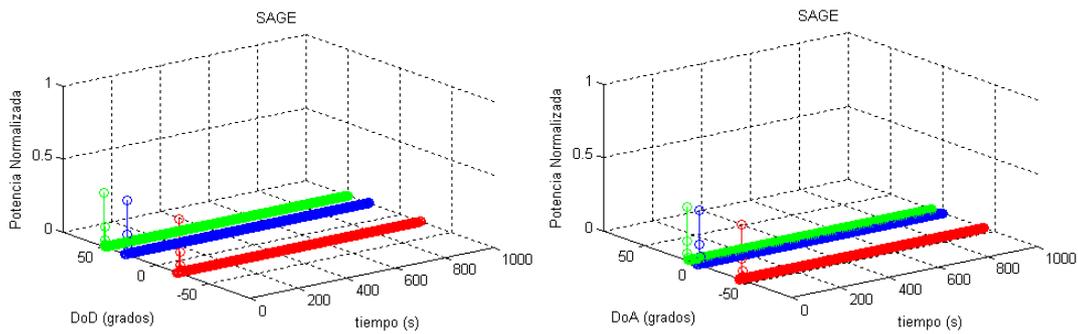


Figura 4.1.27: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 35.94 segundos.

Al aumentar el número de snapshots no aprecia ningún cambio en las estimaciones de los algoritmos, puesto que Beamforming sigue sin detectar tres contribuciones y en caso del resto de algoritmos los resultados siguen siendo los esperados.

Por otro lado, al incrementar el número de muestras los tiempos de ejecución de los distintos algoritmos se han visto modificados, siendo en este caso el que mejor respuesta temporal presenta Beamforming. Es 2D Unitary ESPRIT el más perjudicado, desde el punto de vista de tiempo de ejecución, puesto que presenta un incremento aproximadamente diez veces superior que el resto de algoritmos.



Algoritmo	DOD (°)	DOA (°)	Retardo (ns)	Tiempo Ejecución (s)
Beamforming	[-90,-20,50]	[-40,0,10]	[0,8,10]	49.52
MUSIC	[-20,30,50]	[-40,0,10]	[0,8,10]	57.11
2D Unitary ESPRIT	[-20,30,50]	[-40,0,10]	[0,8,10]	510.06
SAGE	[-20,30,50]	[-40,0,10]	[0,8,10]	35.94

Tabla 4.1.7: Tabla de parámetros estimados tras la tercera simulación.

4.1.4 Cuarta Simulación: Variamos la relación señal a ruido (SNR).

Para comenzar se mostrará la tabla con los valores de la simulación para comparar la precisión de cada uno de los algoritmos en función de los resultados obtenidos en relación a las características introducidas:

Nº Antenas Tx	Nº Antenas Rx	Nº Snapshots	Frec. Muestreo (Hz)	BW (Hz)	Frec. Señal (Hz)
4	4	200	100e6	10e6	2.45e9
DOD (grados)	DOA (grados)	Retardo (seg.)	SNR (dB)	Potencia de Ruido (W)	Resolución
[-20,30,50]	[-40,0,10]	[10,8,0]	[20,15,5]	0.2	5

Tabla 4.1.8: Tabla de parámetros estándar con la información de la cuarta simulación.

Al variar la SNR se ha tenido en cuenta que las señales con mayor retardo al encontrarse más alejadas tienen una mayor atenuación, por tanto la relación señal a ruido deberá ser más elevada la contribución que presente mayor retardo.

La señal generada que se obtiene es:

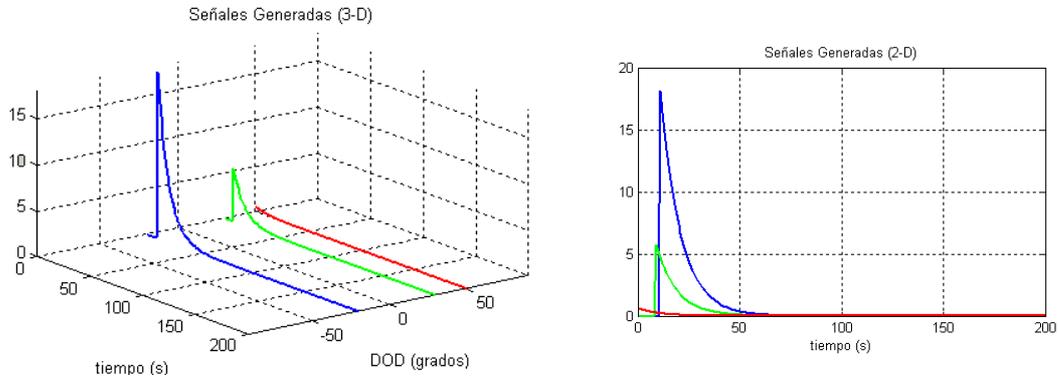


Figura 4.1.28: Señal generada a tras la modificación del SNR de las señales

Tras mostrar la señal generada por el generador de señales se mostrarán los resultados obtenidos por cada uno de los algoritmos:

– Beamforming:

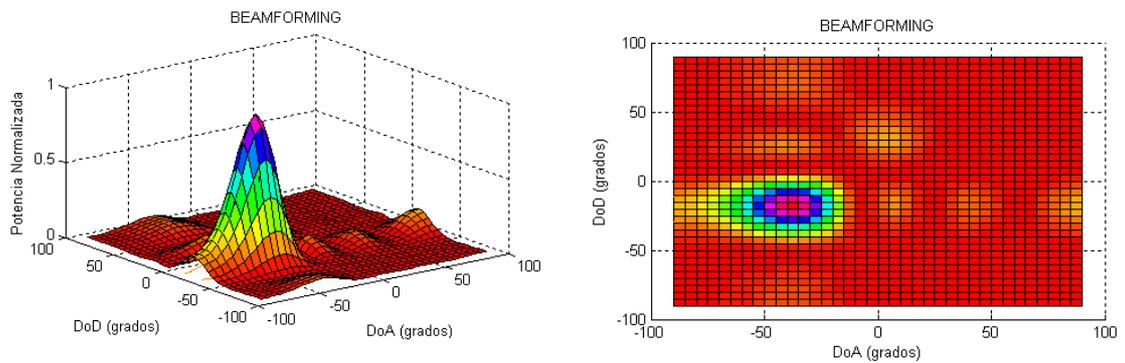


Figura 4.1.29: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

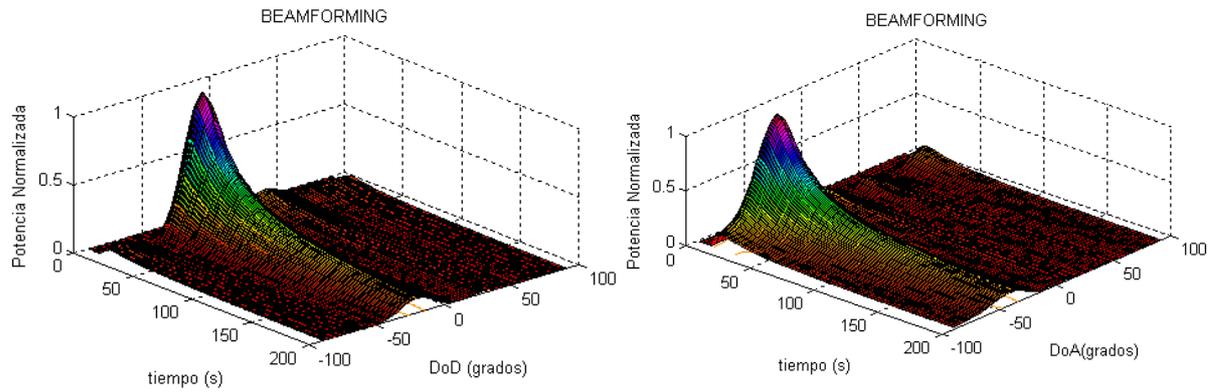


Figura 4.1.30: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 9.25 segundos.

– MUSIC:

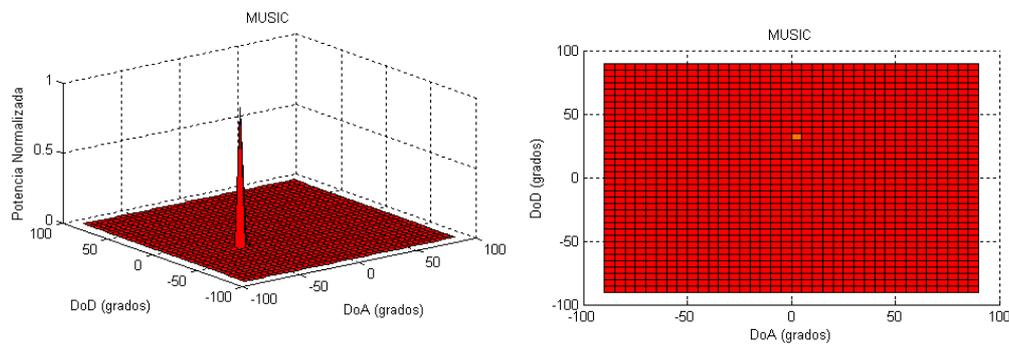


Figura 4.1.31: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

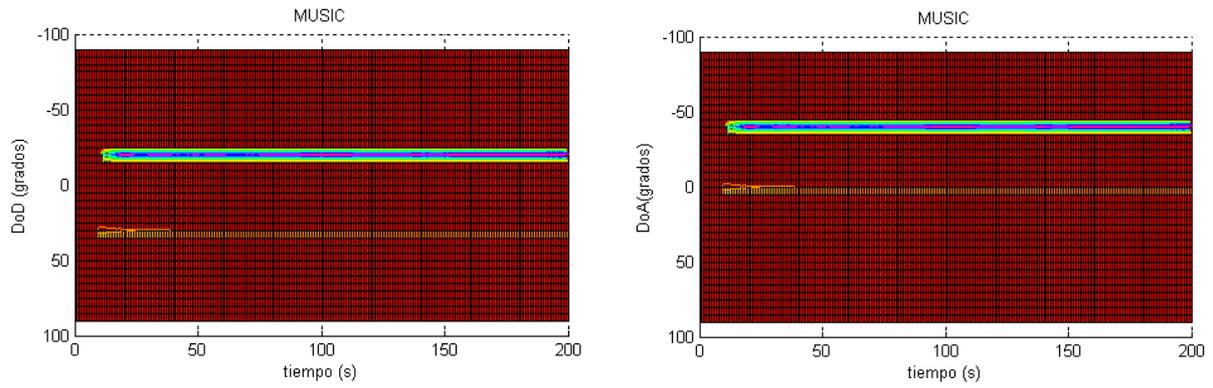


Figura 4.1.32: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 10.41 segundos.

– 2-D Unitary ESPRIT:

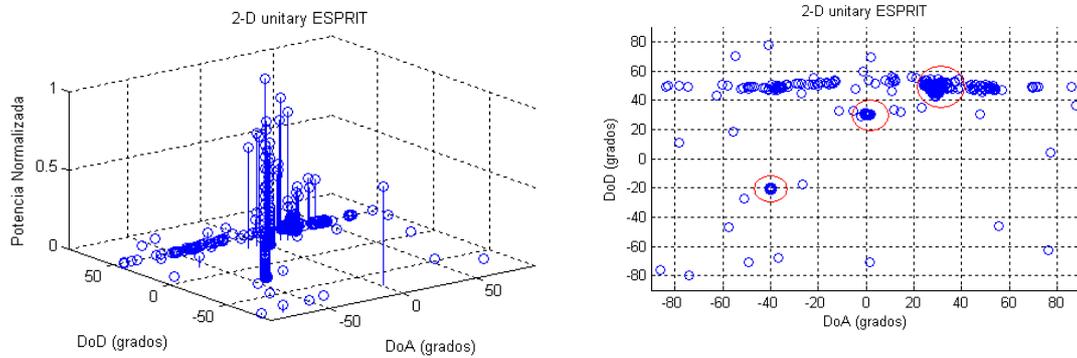


Figura 4.1.33: Resultados de algoritmo 2D Unitary ESPRIT DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

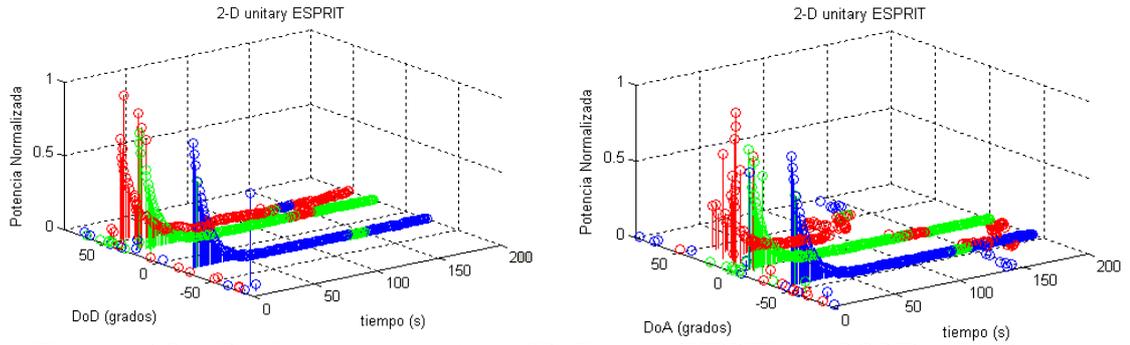


Figura 4.1.34: Resultados de algoritmo 2D Unitary ESPRIT. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 4.23 segundos.

— SAGE:

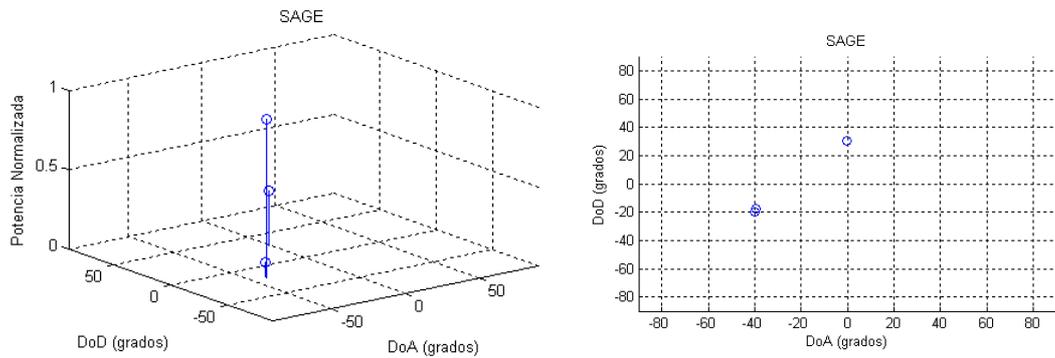


Figura 4.1.35: Resultados de algoritmo SAGE DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

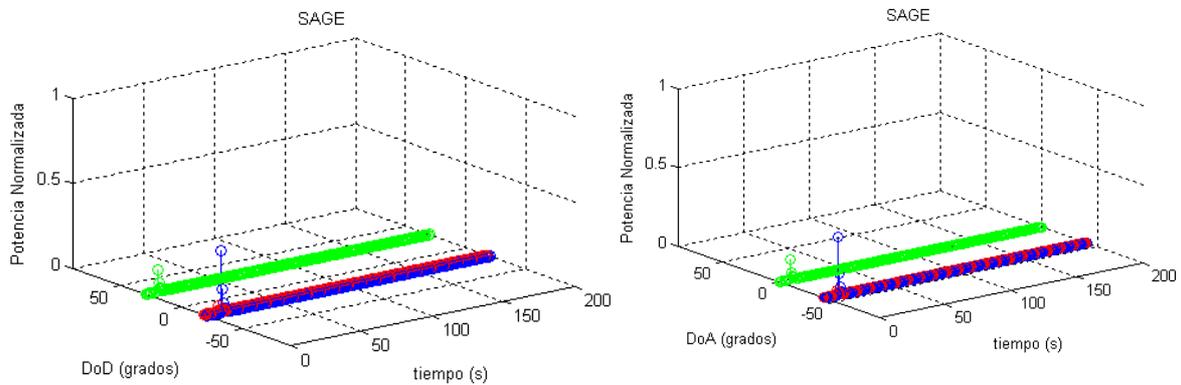


Figura 4.1.36: Resultados de algoritmo SAGE. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 102.92 segundos.

En este caso observamos que en el caso de Beamforming vuelve a aparecer el mismo problema de no detección de las tres contribuciones de señal aunque ahora esta situación es extensible al algoritmo MUSIC que tampoco es capaz de detectar las tres contribuciones. El mejor comportamiento lo presenta el algoritmo 2D Unitary ESPRIT que detecta las tres contribuciones y realiza la estimación en un tiempo muy bajo. Por otro lado el algoritmo SAGE muestra un problema de convergencia en la estimación de parámetros en una de las contribuciones lo que incrementa enormemente su tiempo de ejecución.

Algoritmo	DOD (°)	DOA (°)	Retardo (ns)	Tiempo Ejecución (s)
Beamforming	[-20,-,-]	[-40,-,-]	[-,-,10]	9.25
MUSIC	[-20,30,-]	[-40,0,-]	[-,8,10]	10.41
2D Unitary ESPRIT	[-20,30,50]	[-40,0,30]	[0,8,10]	4.23
SAGE	[-20,30,-]	[-40,0,-]	[-,8,10]	102.92

Tabla 4.1.9: Tabla de parámetros estimados tras la cuarta simulación.



4.1.5 Quinta Simulación: Variamos la resolución de los algoritmos

Para comenzar se mostrará la tabla con los valores de la simulación para comparar la precisión de cada uno de los algoritmos en función de los resultados obtenidos en relación a las características introducidas:

Nº Antenas Tx	Nº Antenas Rx	Nº Snapshots	Frec. Muestreo (Hz)	BW (Hz)	Frec. Señal (Hz)
4	4	200	100e6	10e6	2.45e9
DOD (grados)	DOA (grados)	Retardo (seg.)	SNR (dB)	Potencia de Ruido (W)	Resolución
[-20,30,50]	[-40,0,10]	[10,8,0]	[20,20,20]	0.2	2

Tabla 4.1.10: Tabla de parámetros estándar con la información de la quinta simulación.

En esta simulación no vamos a aplicar los algoritmos 2-D Unitary ESPRIT y SAGE porque la resolución no es un parámetro que varíe el comportamiento de estos algoritmos.

La señal generada que se obtiene es:

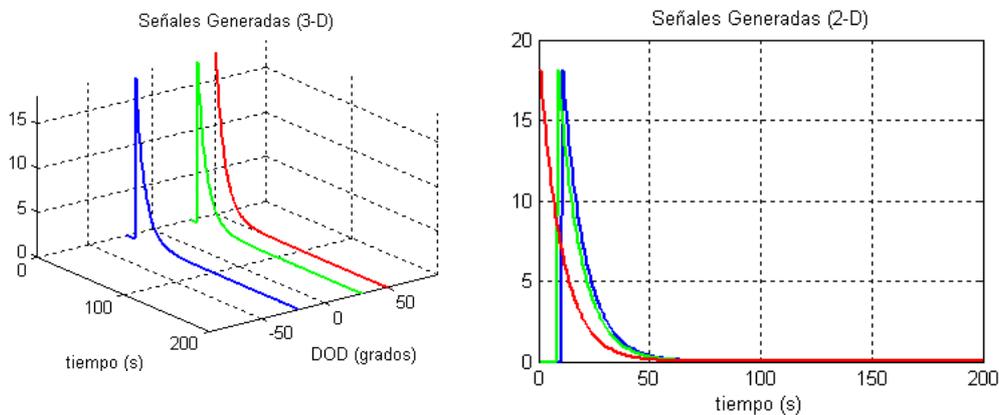


Figura 4.1.37: Señal generada a partir de los valores por defecto de los parámetros

Tras mostrar la señal generada por el generador de señales se mostrarán los resultados obtenidos por cada uno de los algoritmos:

– Beamforming:

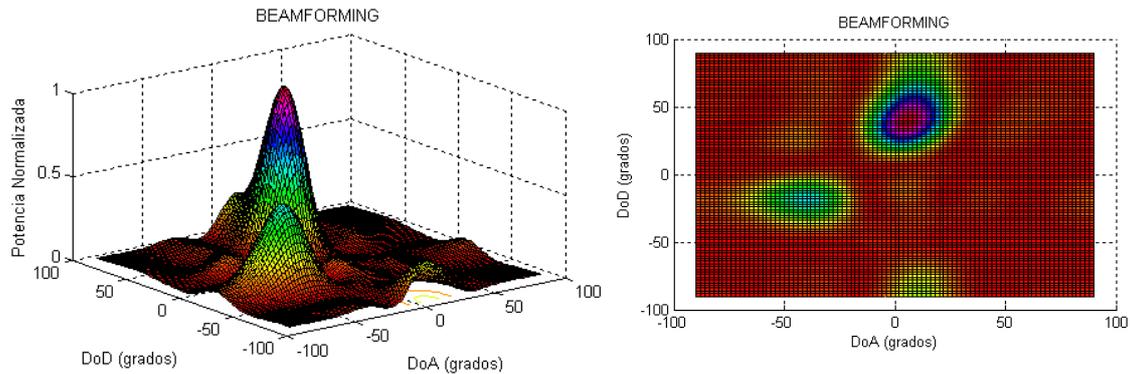


Figura 4.1.38: Resultados de algoritmo Beamforming DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

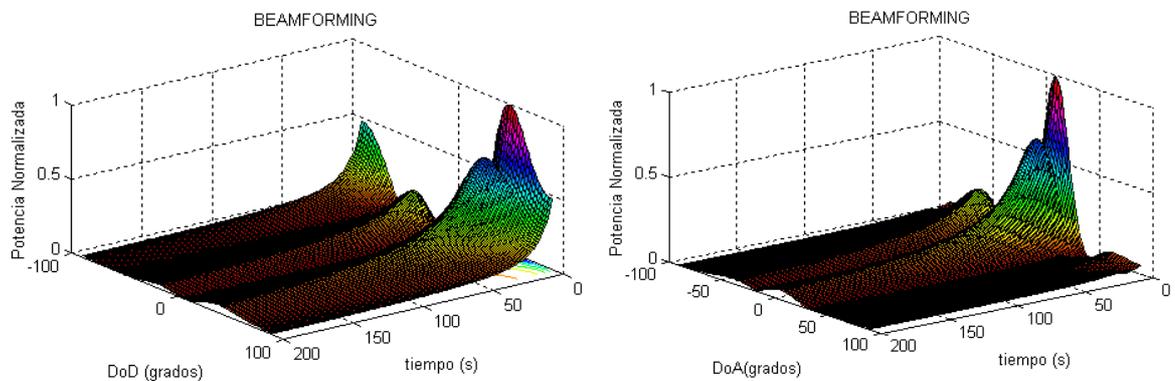


Figura 4.1.39: Resultados de algoritmo Beamforming. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 163.41 segundos.

– MUSIC:

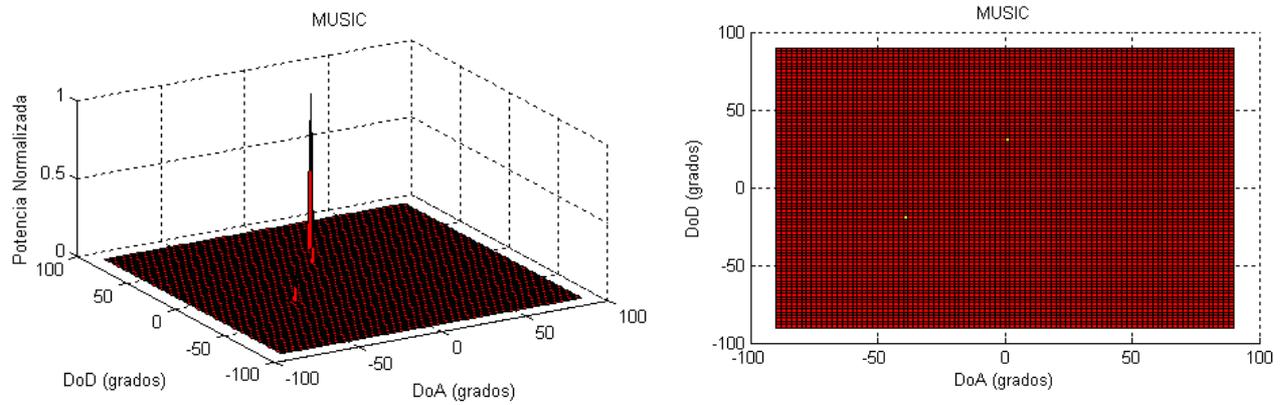


Figura 4.1.40: Resultados de algoritmo MUSIC DOD/DOA/Potencia Normalizada. a) Vista 3D. b) Vista 2D

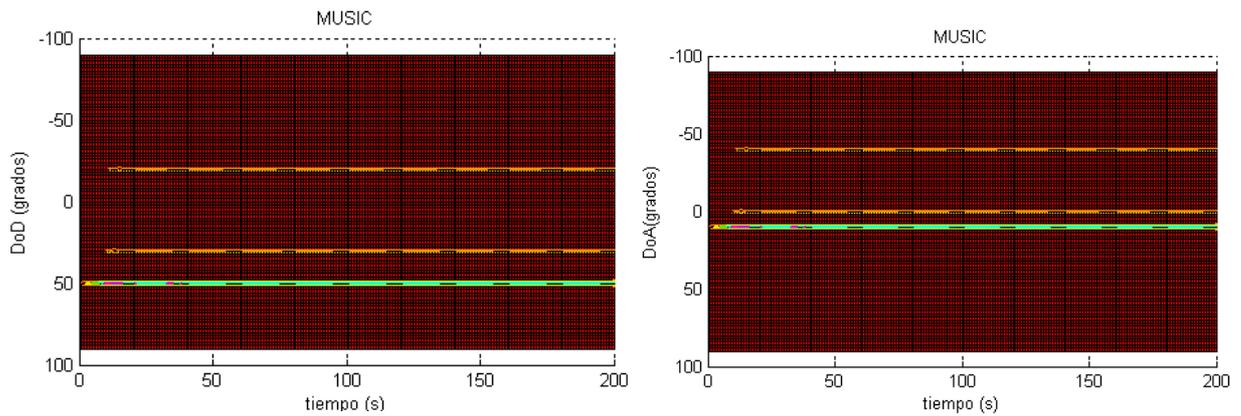


Figura 4.1.41: Resultados de algoritmo MUSIC. a) DOD/Retardo/Potencia Normalizada. b) DOA/Retardo/Potencia Normalizada

El tiempo empleado por el algoritmo para la estimación de parámetros ha sido de 187.19 segundos.



Al aumentar la resolución en ambos algoritmos vemos que se incrementa enormemente el tiempo de ejecución, sin embargo esta modificación no ayuda al algoritmo Beamforming a la detección de contribuciones que se encuentran muy próximas.

Algoritmo	DOD (°)	DOA (°)	Retardo (ns)	Tiempo Ejecución (s)
Beamforming	[-90,-20,50]	[-40,0,10]	[0,8,10]	163.41
MUSIC	[-20,30,50]	[-40,0,10]	[0,8,10]	187.19

Tabla 4.1.11: Tabla de parámetros estimados tras la quinta simulación.



Capítulo 5. Resultados Entorno Real

Tras observar el comportamiento de los algoritmos en un entorno simulado vamos a aplicarlos a un conjunto de medidas reales.

5.1 Medidas

Las medidas utilizadas se corresponden con las realizadas en un túnel de *Roux*, situado en la región de *l'Ardèche* (Zona del macizo central francés), concretamente en la población de *Saint-Cirgues-en-Montagne*. Si se representa la sección transversal del túnel semicircular de diámetro 8.6 metros y altura libre 6.1 metros se obtiene:

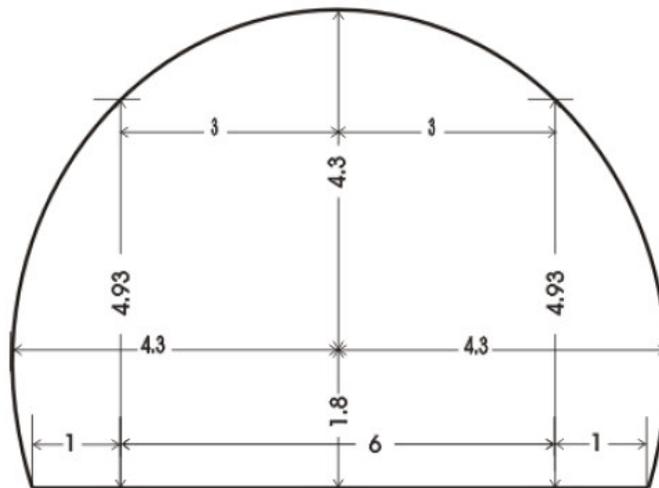


Figura 5.1.1: Sección transversal del túnel en el que se realizaron las medidas.

Para realizarlas se utilizaron dos arrays de 12 antenas separadas $\frac{\lambda}{2}$ en el caso de máxima frecuencia, para el resto de frecuencias el espaciado entre sensores cumple la relación $\frac{\lambda_{max}}{2\lambda_0}$.

El array transmisor y el receptor se colocaron como muestra la figura 5.1.2, situándolos a una distancia de 1 m de las paredes, y se fue alejando el transmisor del receptor. El número de posiciones en las que se tomaron las medidas es de 8 entre 50 y 500 metros.

La frecuencia varía de 2.8 a 5 GHz y se han tomado 1601 muestras entre ellas.

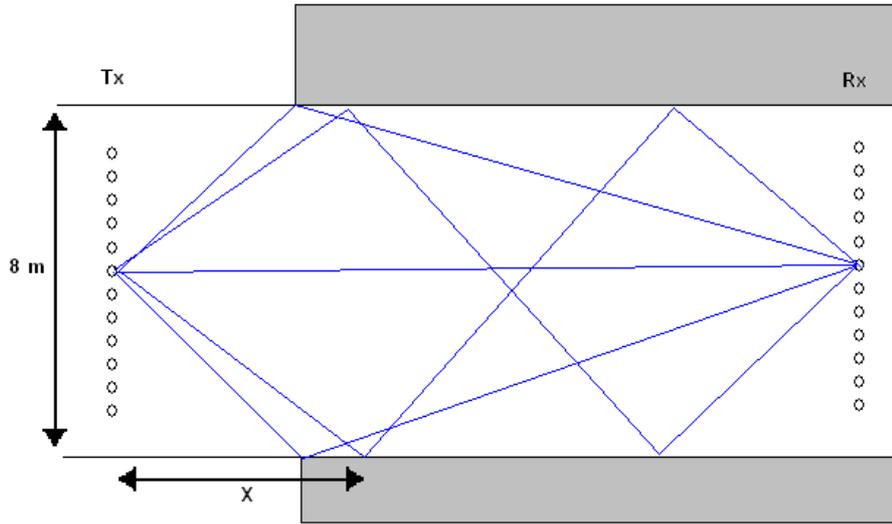


Figura 5.1.2: Disposición de arrays y contribuciones en el túnel.

Las medidas están contenidas en ocho archivos de datos, uno para cada una de las distancias entre transmisor y receptor en las que se han realizado medidas.

Las medidas se realizaron con un equipo de banda ancha. El equipo es una modificación del recogido en [19]. Se sustituyó el analizador de redes multipuerto por un analizador de redes clásico, y el conmutador por dos arrays virtuales. En la Figura 5.1.3 se muestra el esquema y componentes más importantes.

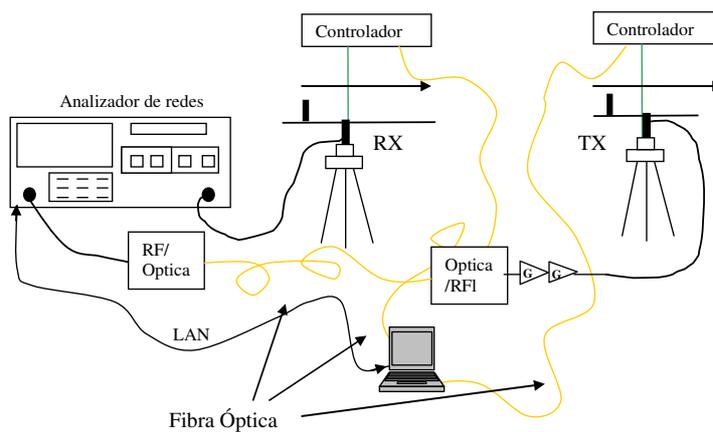


Figura 5.1.3: Esquema del equipo de medidas utilizado para MIMO-Banda Ancha.

Para la obtención de datos experimentales, se ha utilizado un analizador de redes (Agilent E5071B) que genera una señal de banda ancha (tono de 10kHz denominado ancho de banda de frecuencia intermedia que se desplaza desde los 2.8GHz hasta 5GHz en un número de puntos seleccionado en el analizador, 1601 en nuestro caso), con una potencia máxima de salida de 10dBm.

La señal generada se convierte en una señal óptica para minimizar las pérdidas en los cables coaxiales y poder tomar medidas hasta una distancia de 500m.

A continuación, la señal óptica se convierte en una señal de RF (radiofrecuencia), que es amplificada mediante dos amplificadores de 30dB de bajo ruido, y ataca a una antena de banda ancha. En recepción, otra antena recibe la señal y la entrega al analizador de redes.

La respuesta del canal radio se obtiene por el cociente entre tono recibido / tono enviado. La frecuencia inferior (2.8 GHz) viene limitada por los amplificadores utilizados, y la superior (5GHz) por los convertidores óptico-eléctricos.

Con este esquema de trabajo, se obtiene la respuesta de banda ancha para un sistema de una antena transmisora y una antena receptora.

Para obtener la respuesta del canal MIMO, se utilizan dos arrays virtuales, uno en transmisión y otro en recepción. Un sistema mecánico de posicionamiento desplaza de derecha a izquierda las antenas para tomar 8 posiciones diferentes tanto en transmisión como en recepción obteniéndose 64 medidas de banda ancha por posición para una polarización dada en cada posición.

El control del movimiento se realiza mediante unas señales electro-ópticas que se envían desde el PC. La figura 5.1.4 recoge fotos del equipo de medidas.



Figura 5.1.4: Fotos del equipo. A) Transmisor. B) Receptor.

Las medidas fueron MIMO para una combinación donde las antenas están en el mismo carril, empleando 8 horas por recorrido. En cada recorrido se tomaron 15M de puntos. En total se realizaron 4 recorridos (VV, HH, VH y HV), obteniéndose 60M de puntos en 32 horas efectivas.

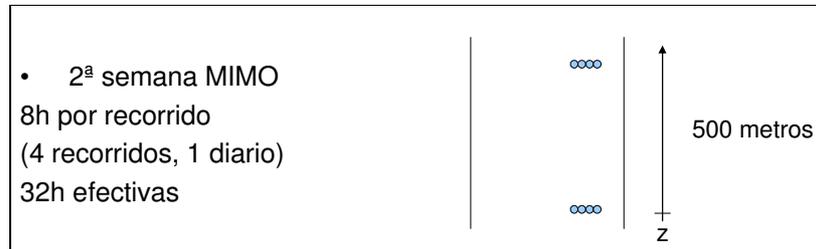


Figura 5.1.5: Configuración de las medidas.

5.2 Resultados

En cuanto a la representación de resultados obtenidos tras la ejecución de los algoritmos sobre las medidas de TELICE, para que sea más sencilla la comprensión y comparación de resultados se han obtenido los valores máximos de los parámetros estimados por cada uno de los algoritmos para obtener así un único resultado de cada uno de los parámetros para cada una de las contribuciones. En caso del algoritmo SAGE no ha sido necesario puesto que este algoritmo estima un solo valor para cada uno de los parámetros para cada una de las contribuciones. En el caso del resto de algoritmos se ha limitado el número de contribuciones a estimar a tres. Utilizando para el caso de MUSIC y Beamforming una resolución de valor 2.

Así para la visualización de los mismos se han representado los parámetros DOD, DOA y el retardo para cada una de las distancias, a distintas frecuencias.

Finalmente, calcularemos los la dispersión angular y temporal que presentan los resultados estimados por cada uno de los algoritmos:

$$\tau_{rms}(s) = \sqrt{\frac{\sum_i PDP'(\tau_i)\tau_i^2}{\sum_i PDP'(\tau_i)} - \left(\frac{\sum_i PDP'(\tau_i)\tau_i}{\sum_i PDP'(\tau_i)}\right)^2} \quad (5.1)$$

Donde $PDP'(\tau_i)$ es la potencia de la contribución i -ésima en unidades lineales, y τ_i es el retardo de la contribución i -ésima, en el caso de la dispersión temporal.

En el caso de la dispersión angular la expresión es:

$$\varphi_{rms} = \sqrt{\frac{\sum_i |A(\varphi)|\varphi^2}{\sum_i |A(\varphi)|} - \left(\frac{\sum_i |A(\varphi)|\varphi}{\sum_i |A(\varphi)|}\right)^2} \quad (5.2)$$



Los resultados obtenidos por cada uno de los algoritmos son:

- 3GHz:

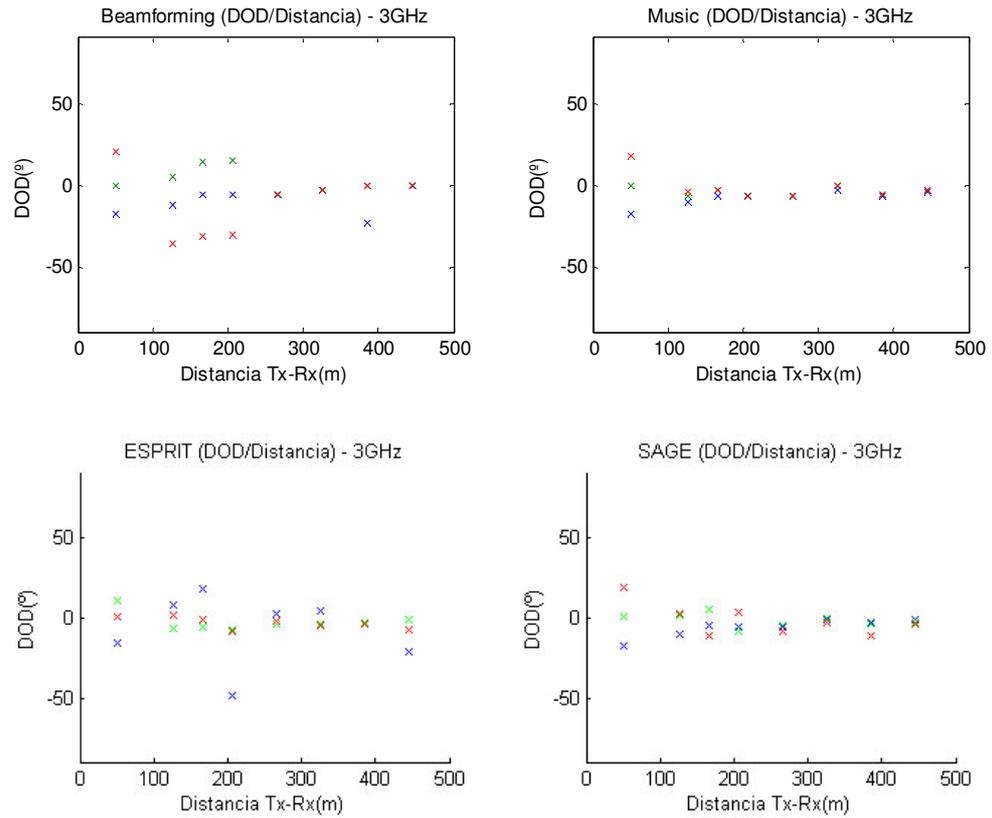


Figura 5.2.1: Resultados de los algoritmos en la estimación del parámetro DOD para cada una de las distancias de medición a una frecuencia de 3GHz.

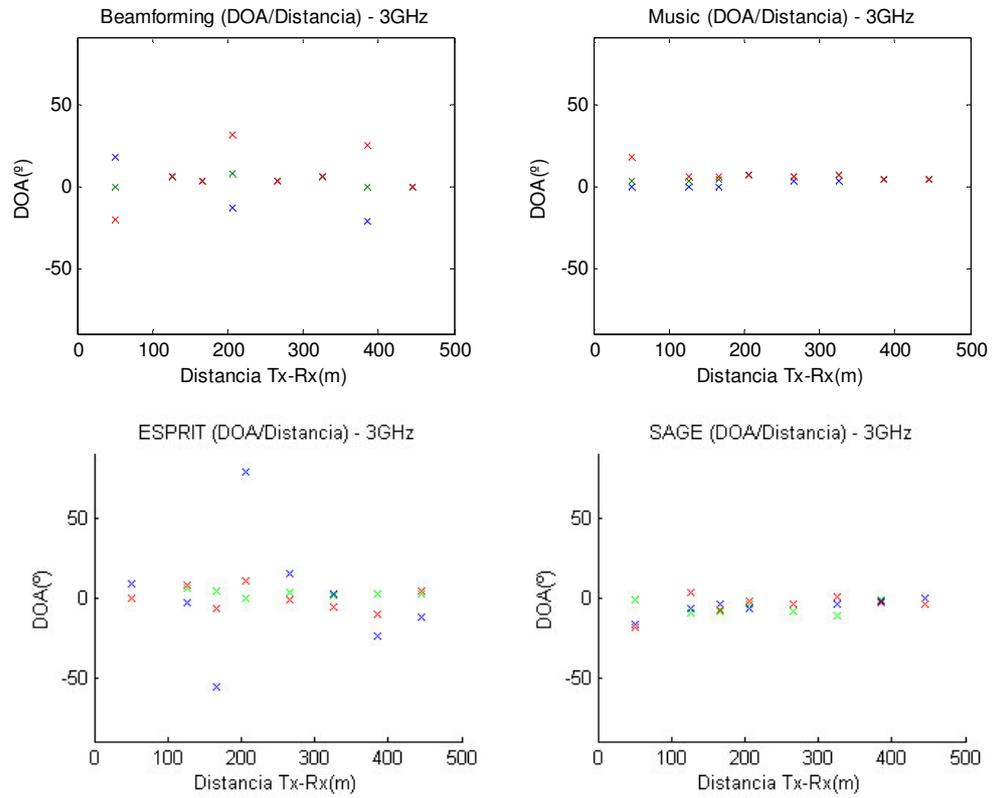


Figura 5.2.2: Resultados de los algoritmos en la estimación del parámetro DOA para cada una de las distancias de medición a una frecuencia de 3GHz.

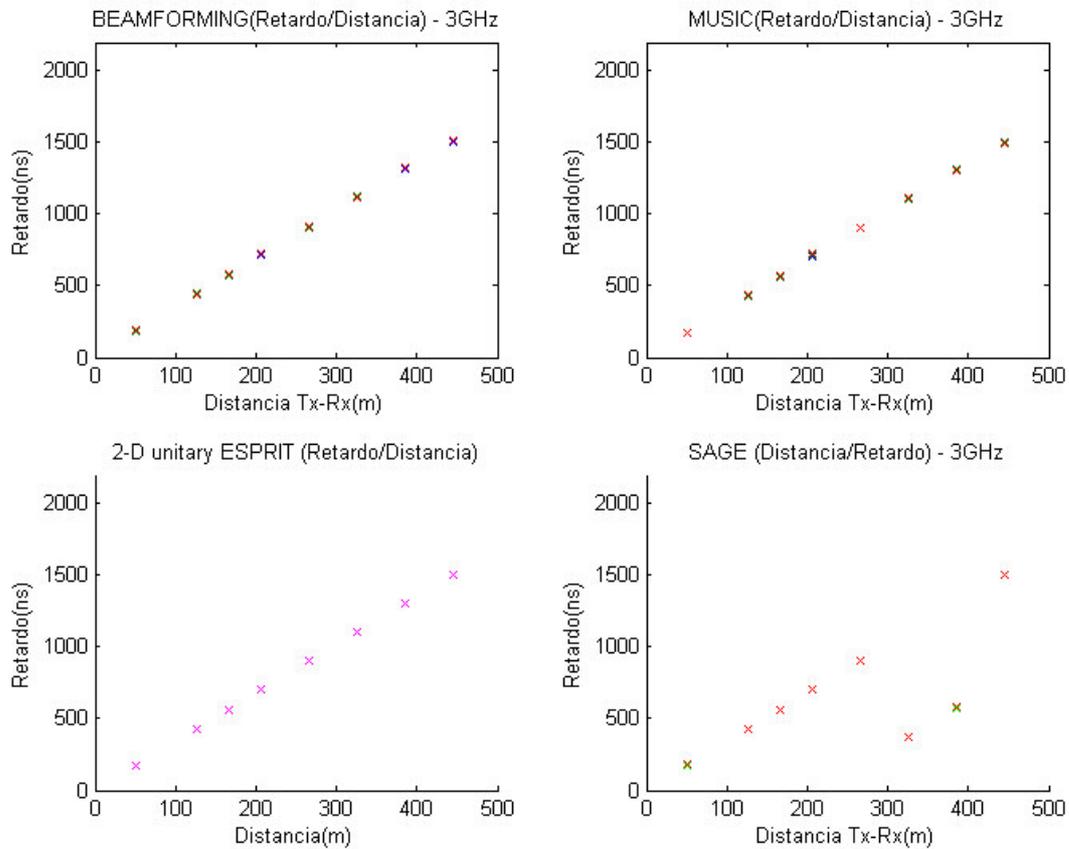


Figura 5.2.3: Resultados de los algoritmos en la estimación del retardo para cada una de las distancias de medición a una frecuencia de 3GHz.

El tiempo empleado por el algoritmo Beamforming para la estimación de parámetros a la frecuencia de 3 GHz ha sido de 1674.4 segundos.

El tiempo empleado por el algoritmo Music para la estimación de parámetros a la frecuencia de 3 GHz ha sido de 1982 segundos.

El tiempo empleado por el algoritmo 2D Unitary ESPRIT para la estimación de parámetros a la frecuencia de 3 GHz ha sido de 365.14 segundos.

El tiempo empleado por el algoritmo SAGE para la estimación de parámetros a la frecuencia de 3 GHz ha sido de 1828.2 segundos.



A continuación se muestra un resumen de los resultados obtenidos tras la aplicación del algoritmo Beamforming a una frecuencia de 3GHz para que sea más fácil su comparación con el resto de algoritmos:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-18,0,20]	[18,0,-20]	[181.24,186.28,191.31]
126	[-36,-12,5]	[6,-,-]	[438,443.03,432.97]
166	[-31,-6,14]	[3,-,-]	[568.9,573.93,578.97]
206	[-30,-6,15]	[-13,8,31]	[714.9,719.93,724.97]
266	[-6,-,-]	[3,-,-]	[911.24,906.21,916.28]
326	[-3,-,-]	[6,-,-]	[1112.6,1117.7,1107.6]
386	[-23,0,-]	[-21,0,25]	[1314,1319,1324.1]
446	[0,-,-]	[0,-,-]	[1500.3,1505.3,1510.3]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	10.701	14.329	2.7283e-009
126	1.1921e-007	18.115	2.132e-009
166	0	8.5576	2.3999e-009
206	2.3637	3.3236	2.8056e-009
266	0	0	2.012e-009
326	0	0	2.8106e-009
386	1.7192	1.1816	3.9023e-009
446	0	0	1.8952e-009

Tabla 5.1: Tabla de resultados obtenidos con el algoritmo Beamforming a 3GHZ.



En el caso del algoritmo Music los valores de los resultados obtenidos para una frecuencia de 3GHz son:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-18,0,18]	[0,3,18]	[171.17,166.14,176.21]
126	[-10,-7,-4]	[3,6,0]	[422.9,427.93,432.97]
166	[-7,-3,-]	[4,6,0]	[558.83,563.86,568.9]
206	[-7,-,-]	[7,-,-]	[704.83,714.9,719.93]
266	[-7,-,-]	[3,6,-]	[896.14,901.17,906.21]
326	[-3,0,-]	[3,7,-]	[1102.6,1097.5,1107.6]
386	[-7,-6,-]	[4,-,-]	[1298.9,1309,1303.9]
446	[-4,-3,-]	[4,-,-]	[1495.2,1500.3,1490.2]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	4.4839	6.2836	3.5233e-009
126	2.3339	1.843	3.004e-009
166	2.5954	1.5058	2.9421e-009
206	0	0	3.0314e-009
266	1.4405	0	3.2439e-009
326	1.8297	1.2927	3.2692e-009
386	0	0.4714	3.2117e-009
446	0	0.4714	2.5027e-009

Tabla 5.2: Tabla de resultados obtenidos con el algoritmo Music a 3GHZ.



Para el caso del algoritmo 2D Unitary Esprit a una frecuencia de 3GHz los resultados son:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-14.94,10.9,0]	[-0.82,-0.22,8.51]	[176.21,176.21,176.21]
126	[-5.27,-6.84,1.24]	[-3.08,6.36,7.96]	[427.93,427.93,427.93]
166	[-6.12,-1.58,18.01]	[-6.57,1.46,4.13]	[563.86, 563.86 ,563.86]
206	[-44.11,-9,-7.36]	[0.03,0.37,10.63]	[704.83 ,704.83,704.83]
266	[-1.94,-4.17, 2]	[-1.75,3.57, 14.20]	[901.17,901.17,901.17]
326	[-5.13,-3.72,4.49]	[-5.89,1.65, 1.98]	[1102.6,1102.6,1102.6]
386	[-4.3,-3.68,-2.78]	[-22.76,-10.78,2.4]	[1303.9,1303.9,1303.9]
446	[-21.2,-6.79,-1.72]	[-13.31,1.88,4.05]	[1500.3,1500.3,1500.3]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	4.3317	12.2	2.5121e-015
126	4.7833	2.1658	0
166	4.0434	10.923	1.0049e-014
206	4.5556	17.133	1.0049e-014
266	7.8296	2.0263	1.0049e-014
326	3.9045	4.6626	1.4211e-014
386	11.069	0.61518	1.4211e-014
446	7.4823	9.2278	2.0097e-014

Tabla 5.3: Tabla de resultados obtenidos con el algoritmo 2D Unitary ESPRIT a 3GHZ.



Finalmente, el algoritmo SAGE a una frecuencia de 3GHz ha obtenido la siguiente estimación de parámetros:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-19,-16.53,-1.2]	[-17.5,0.25,18.5]	[175.58,168.59,178.13]
126	[-9.72,-6.87,3]	[-10.6,1.24,2.4]	[426.38,428.03,429.16]
166	[-8.38,-7.4,-3.64]	[-11.8,-5,4.6]	[559.85,560.65,564.49]
206	[-7.13,-3.8,-2.6]	[-8.45,-5.7, 2.75]	[700.33,704.42,701.1]
266	[-9,-4,-3.8]	[-9.1,-5.1,-5.5]	[899,906.34,905.31]
326	[-11.5,-3.76, 0.5]	[-2.9,-1.3,-0.6]	[370.01,376.28,371.28]
386	[-3.5,-2.2,-1.7]	[-11.7,-4.1,-2.8]	[570.59,568.79,576.13]
446	[-4.2, -4.1,-0.8]	[-3.96,-2.83,-1.04]	[1499.2,1497.2,1499.4]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	14.301	7.8498	3.7629e-009
126	5.7968	1.9046	1.1166e-009
166	4.0575	1.9394	2.0835e-009
206	2.4428	1.5318	1.9158e-009
266	1.1049	1.2911	3.4547e-009
326	0.67537	3.0669	2.4113e-009
386	1.1664	0.29348	3.149e-009
446	1.2273	1.6063	1.0662e-009

Tabla 5.4: Tabla de resultados obtenidos con el algoritmo SAGE a 3GHZ.

Tras observar los resultados obtenidos por cada uno de los algoritmos vemos que los algoritmos que obtienen peores resultados son Beamforming y MUSIC ya que en ambos casos no se detectan todas las contribuciones y al comprobar los valores obtenidos por ESPRIT y SAGE vemos que se trata de contribuciones muy próximas. En vista a estos resultados se ha realizado una ejecución con una resolución=1 para el mejor y el peor caso (50 m. y 446 m.) y los resultados han sido los mismos, no presentan ningún cambio digno de mención.



En cuanto a la dispersión temporal que presenta cada uno de los algoritmos vemos que ésta disminuye conforme aumentamos la distancia entre transmisor y receptor, a excepción del caso de 2D Unitary ESPRIT, en el que, sin embargo, se obtienen valores muy bajos.

Respecto a la dispersión angular, presenta una disminución con la distancia, excepto en el caso del algoritmo 2D Unitary ESPRIT que presenta oscilación en el valor que en la representación de resultados se puede identificar con la aparición de unos valores atípicos.

- 4GHz:

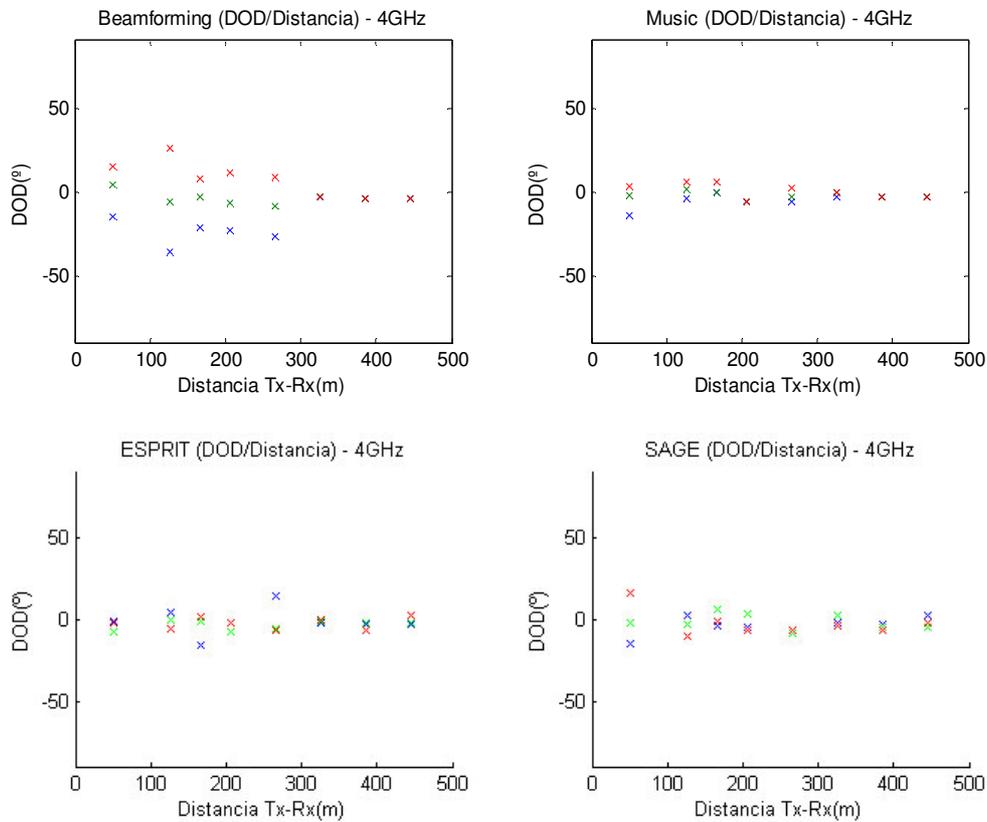


Figura 5.2.4: Resultados de los algoritmos en la estimación del parámetro DOD para cada una de las distancias de medición a una frecuencia de 4GHz.

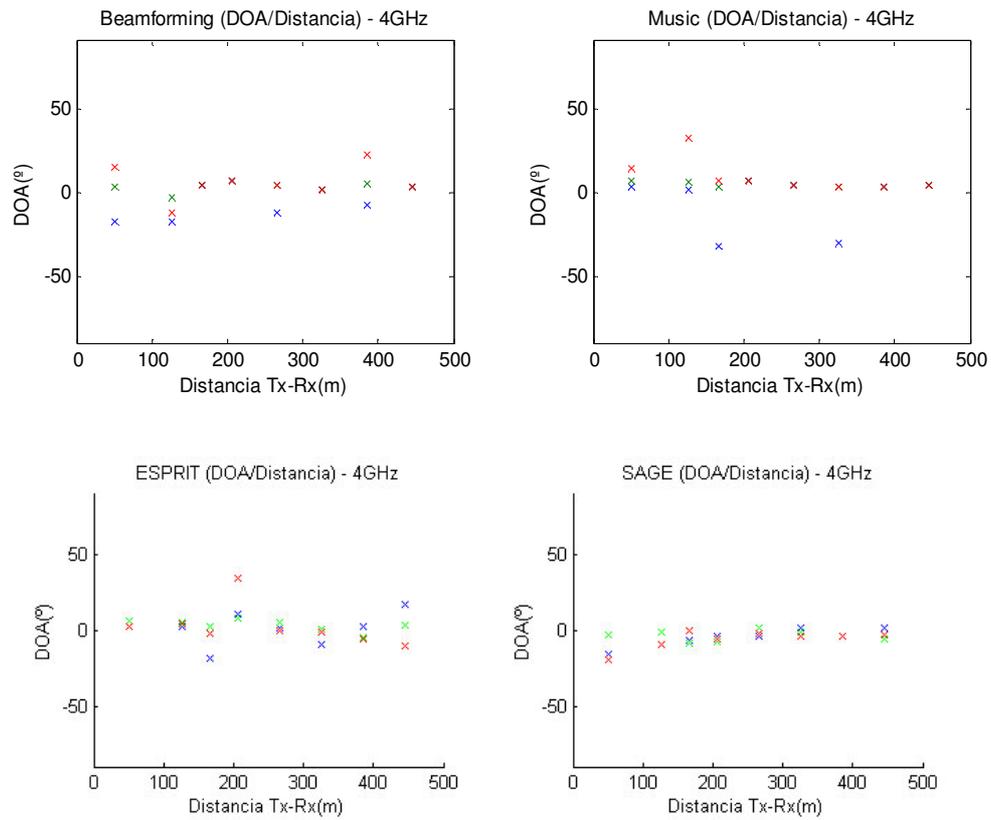


Figura 5.2.5: Resultados de los algoritmos en la estimación del parámetro DOA para cada una de las distancias de medición a una frecuencia de 4GHz.

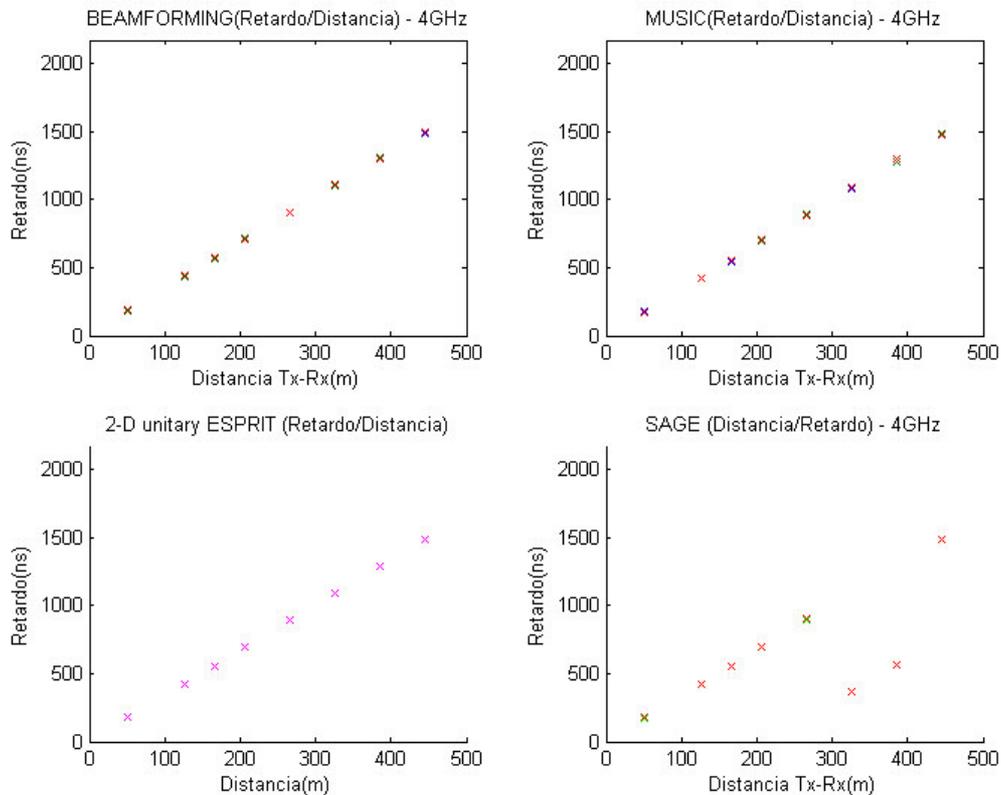


Figura 5.2.6: Resultados de los algoritmos en la estimación del retardo para cada una de las distancias de medición a una frecuencia de 4GHz.

El tiempo empleado por el algoritmo Beamforming para la estimación de parámetros a la frecuencia de 4 GHz ha sido de 1671.4 segundos.

El tiempo empleado por el algoritmo Music para la estimación de parámetros a la frecuencia de 4 GHz ha sido de 2404.8 segundos.

El tiempo empleado por el algoritmo 2D Unitary ESPRIT para la estimación de parámetros a la frecuencia de 4 GHz ha sido de 366.39 segundos.

El tiempo empleado por el algoritmo SAGE para la estimación de parámetros a la frecuencia de 4 GHz ha sido de 2405.4 segundos.



A continuación se muestra un resumen de los resultados obtenidos tras la aplicación del algoritmo Beamforming a una frecuencia de 4GHz para que sea más fácil su comparación con el resto de algoritmos:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-15,4,15]	[-18,3,15]	[176.22,171.18,166.15]
126	[-36,-6,26]	[-18,3,12]	[438.02,432.99,443.06]
166	[-21,-3,8]	[4,-,-]	[548.78,553.82,558.85]
206	[-23,-7,11]	[7,-,-]	[709.9,714.93,704.86]
266	[-27,-9,9]	[-12,4,-]	[901.22,906.25,911.28]
326	[-3,-,-]	[1,-,-]	[1102.6,1107.6,1112.7]
386	[-4,-,-]	[-8,5,22]	[1309,1314.1,1304]
446	[-4,-,-]	[3,-,-]	[1490.3,1495.3,1500.3]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	5.8812	11.456	2.3327e-009
126	4.5623	2.1873	3.4142e-009
166	0	4.8052	2.7898e-009
206	0	1.5549	1.8963e-009
266	0.84536	3.7515	2.4425e-009
326	0	0	2.7993e-009
386	3.3144	0	2.8381e-009
446	0	0	1.6664e-009

Tabla 5.5: Tabla de resultados obtenidos con el algoritmo Beamforming a 4GHZ.



Los resultados obtenidos por el algoritmo Music a una frecuencia de 4GHz son algoritmos:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-14,-2,3]	[3,7,14]	[176.22,171.18,166.15]
126	[-4,1,6]	[1,6,32]	[417.88,422.92,427.95]
166	[0,6,-]	[-32,3,7]	[548.78,553.82,558.85]
206	[-6,-,-]	[7,-,-]	[694.79,699.83,709.9]
266	[-6,-3,2]	[4,-,-]	[891.15,896.18,886.11]
326	[-3,0,-]	[-30,3,-]	[1087.5,1092.5,1097.6]
386	[-3,-,-]	[3,-,-]	[1283.9,1278.8,1299]
446	[-3,-,-]	[4,-,-]	[1485.2,1490.3,1480.2]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	2.1308	4.4838	3.6502e-009
126	3.3594	3.1681	3.1392e-009
166	4.089	1.969	2.2275e-009
206	0	0	2.5513e-009
266	0	2.3473	2.497e-009
326	1.48	1.4941	2.825e-009
386	4.2147e-008	0	3.0618e-009
446	0	0	2.8867e-009

Tabla 5.6: Tabla de resultados obtenidos con el algoritmo Music a 4GHZ.



Para el caso del algoritmo 2D Unitary Esprit a una frecuencia de 4GHz la estimación de parámetros ha dado como resultados:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-8.1,-2.7,-1.7]	[2.2,3,5.8]	[176.22, 176.22, 176.22]
126	[-6.17,-0.8,3.8]	[2.5,4,4.7]	[422.92, 422.92, 422.92]
166	[-15,-1,1.7]	[-17,-2,2.2]	[558.85, 558.85, 558.85]
206	[-7.5,-2.7, -2.3]	[34,10,7.5]	[699.83, 699.83, 699.83]
266	[-7,-6,14.4]	[-0.4,2,4.6]	[896.18, 896.18, 896.18]
326	[-1.9,-1.2,-0.2]	[-2,-9,0]	[1092.5, 1092.5, 1092.5]
386	[-6.6,-2.4,-2.3]	[-6,-5,2]	[1288.9, 1288.9, 1288.9]
446	[-3,-2,1.7]	[-10.5,3,16]	[1485.2, 1485.2, 1485.2]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	1.4418	3.0022	0
126	0.991	4.0144	0
166	9.5228	7.0025	0
206	7.6179	2.2024	0
266	1.3167	10.218	0
326	4.6546	0.38788	2.421e-009
386	3.6904	1.807	1.4211e-014
446	12.788	2.2658	2.0097e-014

Tabla 5.7: Tabla de resultados obtenidos con el algoritmo 2D Unitary ESPRIT a 4GHZ.



Finalmente se mostrarán los resultados obtenidos en la estimación de parámetros del algoritmo SAGE a una frecuencia de 4GHz:

Distancia (m)	DOD (°)	DOA (°)	Retardo (ns)
50	[-19.7,-16,-3]	[-14.7,-2.2,16.1]	[175.45,168.26,178.13]
126	[-9.7, -9.5,-1]	[-10.7,-3.6,2.6]	[422.74,418.86,421.21]
166	[-8.4,-6.9, -0.4]	[-4.5, -1,5.7]	[553.39,555.18,553.13]
206	[-7.6,-6.2,-4.5]	[-4.8,3.5,-6.4]	[695.72,696.48,695.08]
266	[-4.2, -2,1]	[-9,-8.6,-6.8]	[894.09,894.17,900.62]
326	[-4.2, -1.7,1.1]	[-4.2,-2.3,1.8]	[366.69,367.98,365.24]
386	[-4.4,-4.1,-4]	[-6.4,-5,-3.2]	[562.5,568.2,570.2]
446	[-6.1,-3,1.7]	[-4.7,-2.12,2]	[1489.2,1489,1486]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)	RMS Time Spread (s)
50	12.057	6.5735	3.6496e-009
126	5.0542	4.2109	1.7026e-009
166	4.6507	2.8996	9.6137e-010
206	1.796	0.91849	5.6434e-010
266	0.32711	1.9292	1.0213e-009
326	2.6923	1.9876	1.0601e-009
386	1.1973	0.17208	3.4356e-009
446	1.5315	1.7831	1.1083e-009

Tabla 5.8: Tabla de resultados obtenidos con el algoritmo SAGE a 4GHZ.

Al igual que en el caso anterior, vemos que los algoritmos que obtienen peores resultados son Beamforming y MUSIC ya que en ambos casos no se detectan todas las contribuciones y al comprobar los valores obtenidos por ESPRIT y SAGE vemos que se trata de contribuciones muy próximas.

En cuanto a la dispersión temporal que presenta cada uno de los algoritmos vemos que ésta disminuye conforme aumentamos la distancia entre transmisor y receptor.



Respecto a la dispersión angular, vemos que presenta un disminución con la distancia, excepto en el caso del algoritmo 2D Unitary ESPRIT que presenta oscilación en el valor que puede deberse a una mala estimación en los resultados.

- 5GHz:

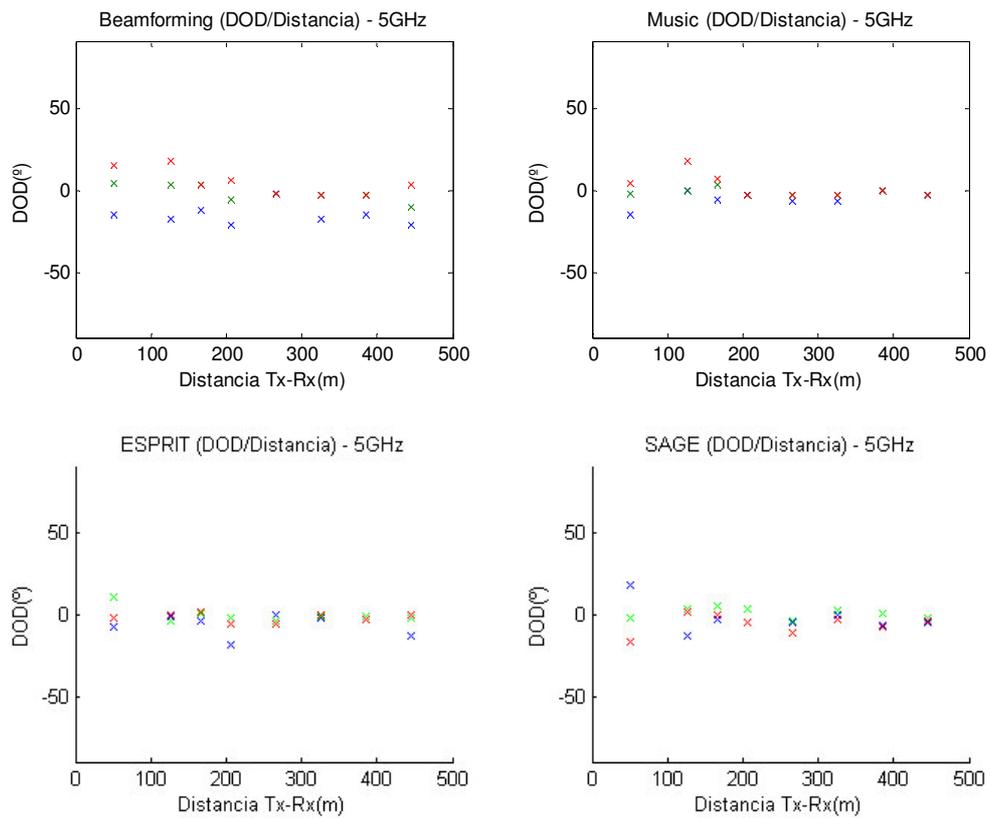


Figura 5.2.7: Resultados de los algoritmos en la estimación del parámetro DOD para cada una de las distancias de medición a una frecuencia de 5GHz.

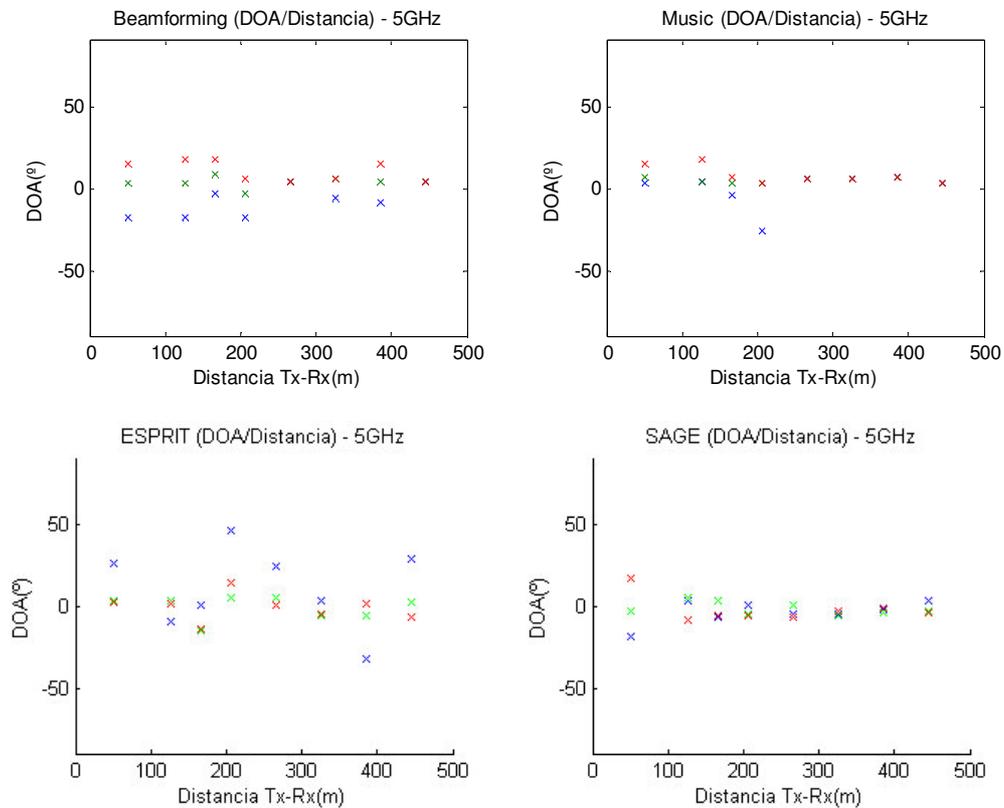


Figura 5.2.8: Resultados de los algoritmos en la estimación del parámetro DOA para cada una de las distancias de medición a una frecuencia de 5GHz.

En este caso no se han presentado los resultados obtenidos de la estimación del retardo por cada uno de los algoritmos puesto que los valores obtenidos no son los esperados.

El tiempo empleado por el algoritmo Beamforming para la estimación de parámetros a la frecuencia de 5 GHz ha sido de 553.62 segundos.

El tiempo empleado por el algoritmo Music para la estimación de parámetros a la frecuencia de 5 GHz ha sido de 1116.3 segundos.

El tiempo empleado por el algoritmo 2D Unitary ESPRIT para la estimación de parámetros a la frecuencia de 5 GHz ha sido de 185.5 segundos.

El tiempo empleado por el algoritmo SAGE para la estimación de parámetros a la frecuencia de 5 GHz ha sido de 1312.3 segundos.



A continuación se muestra un resumen de los resultados obtenidos tras la aplicación del algoritmo Beamforming a una frecuencia de 5GHz para que sea más fácil su comparación con el resto de algoritmos:

Distancia (m)	DOD (°)	DOA (°)
50	[-15,4,15]	[-18,3,15]
126	[-18,3,18]	[-18,3,18]
166	[-12,3,-]	[-3,9,18]
206	[-21,-6,6]	[-18,-3,6]
266	[-2,-,-]	[4,-,-]
326	[-18,-3,-]	[-6,6]
386	[-15,-3,-]	[-9,4,15]
446	[-21,-10,3]	[4,-,-]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)
50	5.8812	11.456
126	6.0525	5.85
166	3.3364	0.76603
206	4.8176	6.083
266	0	0
326	1.5013	0.55282
386	1.7942	0.86815
446	0	6.6787

Tabla 5.9: Tabla de resultados obtenidos con el algoritmo Beamforming a 5GHZ.



En el caso del algoritmo Music a una frecuencia de 5GHz los resultados obtenidos son:

Distancia (m)	DOD (°)	DOA (°)
50	[-15,-2,4]	[3,7,15]
126	[0,18,-]	[4,18,-]
166	[-6,3,7]	[-4,3,7]
206	[-3,-,-]	[-26,3,-]
266	[-7,-3,-]	[6,-,-]
326	[-7,-3,-]	[6,-,-]
386	[0,-,-]	[7,-,-]
446	[-3,-,-]	[3,-,-]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)
50	4.8759	4.3608
126	4.1894	4.0511
166	4.4824	2.7351
206	4.2147e-008	7.2425
266	1.8856	8.4294e-008
326	1.9957	0
386	0	0
446	4.2147e-008	0

Tabla 5.10: Tabla de resultados obtenidos con el algoritmo MUSIC a 3GHZ.



Los resultados obtenidos por el algoritmo 2D Unitary ESPRIT a una frecuencia de 5GHz son:

Distancia (m)	DOD (°)	DOA (°)
50	[-8,-2.7,10]	[2.6,3.1,25.8]
126	[-4.2,-1.3, 0]	[-9.1,3,1.5]
166	[-4.5,0,1]	[-15,-14.2,0.5]
206	[-18.3,-5.8,-1.8]	[5,13.7,46.2]
266	[-5.9,-3.8,-0.7]	[5.3,0.2,24.3]
326	[-2.4,-1.7,-0.4]	[-6,-5.1,3.1]
386	[-3.4,-3.1,-1.6]	[-32.7,-6,1.7]
446	[-13.1,-2.2,-0.3]	[-6.6,2.6,28.4]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)
50	11.338	8.9338
126	6.0047	1.4736
166	7.3692	2.7055
206	13.029	5.2378
266	9.5365	1.5857
326	4.4617	0.55694
386	15.366	0.60228
446	16.652	6.1531

Tabla 5.11: Tabla de resultados obtenidos con el algoritmo 2D Unitary ESPRIT a 5GHZ.



Finalmente, los resultados presentados por el algoritmo SAGE a una frecuencia de 5GHz son:

Distancia (m)	DOD (°)	DOA (°)
50	[-19,-3.3,16.8]	[-17,-2.1,17.5]
126	[-8.3,3.6,5.1]	[-12.8,1.2,3]
166	[-6.8,-6.2,3.2]	[-3.6,0,5.5]
206	[-5.7,-5,0.5]	[-5,-5,3.1]
266	[-7.2,-4.8,0.5]	[-11,-4.7,-4]
326	[-6.2,-4.6, -3]	[-3.2,0,2]
386	[-4.5,-2.5, -1]	[-7.4,-6.8,0.7]
446	[-4,-3,3.2]	[-5.4,-4.2,-2]
Distancia (m)	RMS Angular Spread DOA (°)	RMS Angular Spread DOD (°)
50	11.406	10.728
126	7.5602	5.0217
166	3.8322	4.3257
206	3.9951	2.1324
266	1.1117	2.0056
326	1.6316	1.0109
386	2.182	1.0621
446	1.1864	2.0473

Tabla 5.12: Tabla de resultados obtenidos con el algoritmo SAGE a 5GHZ.

Al igual que en los casos anteriores, vemos que los algoritmos que obtienen peores resultados son Beamforming y MUSIC ya que en ambos casos no se detectan todas las contribuciones y al comprobar los valores obtenidos por ESPRIT y SAGE vemos que se trata de contribuciones muy próximas.



En cuanto a la dispersión temporal, no se han mostrado los resultados obtenidos porque el retardo estimado por cada uno de los algoritmos era incorrecto. Esto se debe a que se produce un aumento de la dispersión temporal con la frecuencia.

Respecto a la dispersión angular, vemos que, de forma general, presenta un disminución con la distancia, excepto en el caso del algoritmo 2D Unitary ESPRIT que presenta oscilación en el valor que puede deberse a una mala estimación en los resultados.



Capítulo 6. Conclusiones

A lo largo de este proyecto hemos cumplido los dos propósitos que nos planteamos a su inicio.

Así, en primer lugar, hemos adaptado una herramienta, en el lenguaje de programación de MatLab, que nos permite observar y comparar el comportamiento de los algoritmos en un entorno de simulación. Este entorno de simulación el usuario puede generar un sistema MIMO completo, permitiendo variar cualquiera de los parámetros que lo constituyen.

En segundo lugar, hemos adaptado los algoritmos Beamforming Conventional, MUSIC, 2-D Unitary ESPRIT y SAGE, con el fin de poder estimar los DOD y DOA de los arrays de transmisión y recepción, así como la potencia y el retardo de las distintas contribuciones, a partir de unas medidas reales.

Tras las simulaciones en el entorno simulado podemos concluir lo siguiente respecto cada uno de los algoritmos:

- El algoritmo Beamforming presenta un buen comportamiento, en cuanto a tiempo de ejecución, sin embargo cabe remarcar que este algoritmo posee una resolución muy baja y no permite detectar contribuciones de señal que se encuentren demasiado próximos entre sí.
- El algoritmo MUSIC presenta un buen comportamiento en entornos con una SNR alta pero presenta una alta dependencia con los instantes de muestreo empleados lo que puede hacer que se pierdan contribuciones o que se confundan con el ruido del canal. Sin embargo, hay que remarcar que este algoritmo es el que presenta un el peor comportamiento en cuanto a tiempo de ejecución.
- El algoritmo 2D Unitary ESPRIT presenta un buen comportamiento en general ante todas las situaciones presentadas además de ser de los cuatro algoritmos el que mejores tiempos de ejecución presenta.
- El algoritmo SAGE presenta un buen comportamiento ante la mayoría de las situaciones planteadas, sin embargo, cuando se presentan valores muy bajos de SNR el algoritmo no consigue alcanzar la convergencia incrementando enormemente su tiempo de ejecución.

Si tenemos en cuenta los resultados obtenidos en la estimación de parámetros sobre medidas reales.

Después de observar los resultados obtenidos tras aplicar los algoritmos en las medidas reales de TELICE podemos decir que las conclusiones son prácticamente las mismas que las mencionadas anteriormente en el entorno simulado. Respecto a cada uno de los algoritmos:

- El algoritmo Beamforming presenta un problema de detección de contribuciones. Sin embargo, presenta un buen comportamiento respecto a las dispersiones



- temporales y angulares, puesto que, al obtener una sola contribución en muchos de los casos, no presenta casi dispersión.
- El algoritmo MUSIC presenta el mismo problema que el algoritmo Beamforming pero además en cuanto a tiempo de ejecución del algoritmo presenta peores resultados que Beamforming. Con lo que, si además tenemos en cuenta que es el algoritmo que emplea mayor tiempo de ejecución, podemos concluir que es el que peor comportamiento presenta.
 - El algoritmo 2D Unitary ESPRIT estima todas las contribuciones pero obtiene resultados muy oscilantes en el caso de la dispersión angular. Respecto al tiempo de ejecución del algoritmo es el que mejores resultados presenta.
 - El algoritmo SAGE es, en general, el que mejores resultados obtiene, sin embargo, cuando presenta problemas de convergencia aumenta enormemente su tiempo de ejecución.



Referencias

- [1] S. Haykin, editor. *Array Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [2] J. M. Hernández Rábanos, *Comunicaciones Móviles*, Madrid, Centro de Estudios Ramón Areces, 1997.
- [3] J.M. Hernando Rábanos, *Comunicaciones móviles de tercera generación*, Telefonica Móviles España S.A., Madrid 2001.
- [4] Frank Dellaert, *The Expectation Maximization Algorithm*, College of Computing, Georgia Institute of Technology Technical Report number GIT-GVU-02-20, February 2002.
- [5] Jeffrey A. Fessler and Alfred O. Hero, *Space-Alternating Generalized Expectation-Maximization Algorithm*, Dept. of Electrical Engineering and Computer Science, The University of Michigan, Oct. 1994.
- [6] Pablo Ródenas Vicente. *Implementación de algoritmos para la detección de la dirección de llegada en sistemas multiantena*. PFC Universidad Politécnica de Cartagena. Abril 2005.
- [7] B. H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus & K. Ingeman Pedersen, *Channel Parameter Estimation in Mobile Radio Environment Using the SAGE Algorithm*, IEEE Journal on Selected Areas in Communications, Vol. 17, No. 3, pp. 434 – 450, March 1999.
- [8] L. Ahlin and J. Zander. *Principles of Wireless Communications*. Studentlitteratur, Lund, Sweden, 1997.
- [9] <http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html>
- [10] Hamid Krim and Mats Viberg. *Two decades of array signal processing research: The parametric approach*, IEEE Signal Processing Magazine, Vol. 13, No. 4, pp. 67-94, July 1996.
- [11] Encarna Robles Arnao, *Extracción de parámetros del canal doble-direccional para sistemas MIMO*. PFC Universidad Politécnica de Cartagena. Diciembre 2005. Apéndice A.
- [12] <http://www.mathworks.com/products/matlab/>
- [13] Bernard H. Fleury, Patrick Jourdan, Andreas Stucki, *High-Resolution Channel Parameter Estimation for MIMO Applications Using the SAGE Algorithm*. 2002 International Zurich Seminar on Broadband Communications.
- [14] K. D. Laakmann and William H. Steier, *Waveguides: characteristic modes of hollow rectangular dielectric waveguides*, Applied Optics, vol. 14, no. 5, pp 1134-40, 1976.
- [15] Lal Chand Godara. *Handbook of Antennas in Wireless Communications*. University of New South Wales, Australia, 2001.
- [16] M. Haardt and J.A. Nossek. *Simultaneous Schur Decomposition of Several Nonsymmetric Matrices to Achieve Automatic Pairing in Multidimensional Harmonic Retrieval Problems*. IEEE Trans. Signal Processing 46(1): 161-169. Jan. 1998.
- [17] M. Haardt and J. A. Nossek. *Unitary ESPRIT: How to obtain increased estimation accuracy with a reduced computational burden*. IEEE Trans. Signal Processing, 43(5): 1232-1242, May 1995.



- [18] M. Kaveh and A. J. Barbell. *The Statistical Performance of the MUSIC and the Minimum-Norm Algorithms in Resolving Plane Waves in Noise*. IEEE Trans. ASSP, ASSP-34:331 – 341, Apr. 1986.
- [19] Rubén Ibernón Fernández, Jose-María Molina-García-Pardo y Leandro Juan-Llácer, “Wideband MIMO Channel Sounder Based on a Network Analyzer”, enviado al IET Communications Marzo 2007.



Apéndices

Apéndice I. Verosimilitud

Dado un conjunto de datos “completo” $X_1(t)$, el conjunto de observaciones $Y(t)$ y el

conjunto de parámetros a estimar θ_L , la expectativa o valor esperado condicional de la verosimilitud de los datos “completos” dada una observación para cada contribución se define como:

$$V_l(\theta_l; \hat{\theta}^i) = E\left\{\Lambda(\theta_l; \hat{\theta}^i; X_l(t)) \middle| y(t); \hat{\theta}^i\right\}$$

$$V_l(\theta_l; \hat{\theta}^i) = \frac{1}{N_0} \left[2 \int_D \Re \left\{ s(t; \theta_l, \hat{\theta}^i)^H \hat{x}_l(t; \hat{\theta}^i) \right\} dt - \int_D \|s(t; \theta_l, \hat{\theta}^i)\|^2 dt \right]$$

Siendo la verosimilitud logarítmica:

$$\Lambda(\theta_l; \hat{\theta}^i; X_l(t)) = \frac{1}{N_0} \left[2 \int_D \Re \left\{ s(t; \theta_l, \hat{\theta}^i)^H x_l(t) \right\} dt - \int_D \|s(t; \theta_l, \hat{\theta}^i)\|^2 dt \right]$$

Donde $[\cdot]^H$ es el operador Hermitiano.



Apéndice II.

II.1 Operador Vec

El operador vec transforma una matriz en un vector apilando sus columnas. Si X es una matriz $n \times m$, y $X_{t,i}$ es la columna i -ésima de X , el operador vec se define como:

$$\text{vec}X = \begin{bmatrix} X_{t,1} \\ X_{t,2} \\ \vdots \\ X_{t,m} \end{bmatrix}$$

Con lo que $\text{vec}X$ es un vector columna de longitud nm . Una propiedad que se deduce directamente de este operador es que $(X + Y) = \text{vec} X + \text{vec} Y$

II.2 Producto Kronecker

El producto Kronecker entre una matriz X de tamaño $n \times m$ y una matriz Y de tamaño $o \times p$ se define como:

$$X \otimes Y = \begin{bmatrix} x_{11}Y & x_{12}Y & \cdots & x_{1m}Y \\ x_{21}Y & x_{22}Y & \cdots & x_{2m}Y \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1}Y & x_{n2}Y & \cdots & x_{nm}Y \end{bmatrix}$$

Donde x_{ij} determina el elemento i -ésimo, j -ésimo de X . Por lo tanto, $X \otimes Y$ es una matriz $no \times mp$.



II.3 Teorema

La siguiente propiedad del operador vec es fundamental para derivar el modelo de datos del canal MIMO.

Considerando las matrices X, Y y Z tenemos la siguiente propiedad

$$\text{vec}(XYZ) = (Z^T \otimes Z)\text{vec}Y$$