

Técnicas de síntesis de agrupaciones para aplicaciones radar y aeronáuticas

Vicente Nova Giménez

Tutor: Alejandro Valero Nogueira

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2013-2014
Valencia, 5 de Julio de 2014

«Dios dirige el electromagnetismo los lunes, miércoles y viernes mediante la teoría ondulatoria, y el demonio lo dirige mediante la teoría cuántica el martes jueves y sábado»

William Lawrence Bragg

Capítulo 1

Resumen

En la actualidad son numerosas las aplicaciones que precisan de un diagrama de radiación con una forma muy específica y/o unos requerimientos muy altos. Existen infinidad de métodos para la síntesis de agrupaciones que consiguen proporcionar soluciones a estas aplicaciones, pero muchas de éstas, difícilmente pueden ser implementadas de manera sencilla en la vida real. De la necesidad de conseguir soluciones de fácil implementación en tecnologías asequibles y actuales nace este trabajo. Con el desarrollo en MATLAB del método de síntesis *alternating projection method* podemos *pulir* los resultados obtenidos por otros métodos de síntesis para facilitar su implementación práctica imponiendo límites a las corrientes o, directamente, realizar la síntesis del array sin imponer restricciones, si así se desea.

Resum

En la actualitat son nombroses les aplicacions a les que li fa falta un diagrama de radiació amb una forma molt específica i/o uns requeriments molt alts. Existeixen infinitat de mètod per a la síntesi d'agrupacions que conseqüeixen proporcionar solucions a aquestes aplicacions, però moltes d'aquestes, difícilment poden ser implementades de manera sencilla en la vida real. De la necessitat d'aconseguir solcions de facil implementació amb tecnologies assequibles y acutals naix aquest treball. Amb el desenvolupament en MATLAB del mètod de síntesi *alternating projection method* podem *pulir* els resultats obtessos per uns altres mètods de síntesi, per a facilitar la seva implementació pràctica, imposant límits a les intensitats o, directament, realitzar la síntesi del ".array" sense imposar restriccions, si ho desitja així.

Abstract

Nowadays a lot of applications require a complex phased array with high requirements. There are many methods of synthesis of arrays that perform well, but the solutions of these are usually too complex to be implemented in a current technology. The project aim is provide simple solutions to complex problems. Alternating projection method can perfect the solutions of other method and provide a simpler solution for implementation in real life.

Índice general

1. Resumen	2
2. Introducción	6
3. Objetivos	8
3.1. Objetivos personales	8
4. Metodología	9
5. Desarrollo.	10
5.1. Principios básicos de Agrupaciones de antenas.	10
5.1.1. Arrays lineales.	11
5.2. Principios básicos de ranuras como elemento radiante:	12
5.2.1. Agrupaciones lineales de ranuras en cavidades resonantes	13
5.3. Principios del algoritmo <i>alternating projection method</i> :	15
5.3.1. Funcionamiento del algoritmo <i>alternating projection method</i> :	16
5.4. Consideraciones prácticas del algoritmo:	17
5.5. Consideraciones del código:	21
5.6. Introducción a Matlab.	22
5.7. Síntesis de arrays realizadas con el algoritmo:	23
5.7.1. Síntesis de un array con lóbulo principal plano:	23
5.7.2. Síntesis de un array de lóbulo principal inclinado con cierta pendiente:	28
5.7.3. Síntesis de un array con límites en el desfase de corrientes:	33
6. Conclusión	39
7. Anexo	41
7.0.4. Función principal:	41
7.0.5. Límites del Factor de array:	43
7.0.6. Límites de las corrientes:	44
7.0.7. Funciones auxiliares:	46

Índice de figuras

2.1. Satélite Scatterometer para la medida del viento en alta mar (SeaWinds)	7
2.2. Medida de la humedad en una inundación en Camboya	7
5.1. Esquema desfase	12
5.2. Circuitos equivalentes de ranuras	13
5.3. Alimentación de la guía resonante	14
5.4. Subconjunto Bc	15
5.5. Subconjunto M	15
5.6. Solución	16
5.7. Diagrama de flujo del algoritmo desarrollado	20
5.8. Formato de entrada de los límites	21
5.9. Soluciones final e inicial	24
5.10. Máscara	24
5.11. Solución final con límites	25
5.12. Solución final con límites, lóbulo principal	25
5.13. Soluciones final y solución del artículo [1]	25
5.14. Solución propuesta en el artículo [1]	26
5.15. Límites impuestos junto a solución con límites de corrientes	27
5.16. Solución con límites de corrientes junto a solución propuesta en el artículo [1]	27
5.17. Zonas en las que se define el patrón de radiación	28
5.18. Solución con límites no estrictos	29
5.19. Máscara de inicio para síntesis solofase	30
5.20. Patrón de partida y Solución final	30
5.21. Límites impuestos al patrón de radiación	31
5.22. Solución final comparada con la solución sugerida en el artículo [1]	31
5.23. Límites impuestos al patrón de radiación	32
5.24. Patrón de partida y Solución final	32
5.25. Solución final con y sin límites en el desfase comparadas con la solución sugerida en el artículo [1]	33
5.26. Cambio de espacio desde el factor de array al espacio real	34
5.27. Especificaciones (rojo) y solución de partida (Azul)	35
5.28. Distintas soluciones del algoritmo junto al diagrama de partida	36
5.29. Diagrama de radiación del conjunto, teórico e implementación	37
5.30. Módulos de corrientes teóricas y aproximación real implementada	37
5.31. Fases de corrientes teóricas y aproximación real implementada	38
5.32. Esquemático descriptivo del posicionamiento de las ranuras en la guía resonante	38
5.33. Parámetro S_{11} del conjunto del array	38

Índice de cuadros

5.1. Desarrollo en series de Fourier realizado con una FFT	19
5.2. Especificaciones del diagrama	23
5.3. Máscara de conformación de patrón	23
5.4. Módulo de corrientes normalizadas	24
5.5. Ángulo de corrientes normalizadas	24
5.6. Módulo de corrientes normalizadas	26
5.7. Fase de corrientes normalizadas	26
5.8. Módulo de corrientes normalizadas con límites de desfase	27
5.9. Fase de corrientes normalizadas con límites de desfase	27
5.10. Módulo de corrientes normalizadas	29
5.11. Ángulo de corrientes normalizadas	30
5.12. Desfase entre corrientes normalizadas contiguas	30
5.13. Módulo de corrientes normalizadas	32
5.14. Fase de corrientes normalizadas	32
5.15. Desfase entre corrientes normalizadas de elementos contiguos	32
5.16. Corrientes normalizadas solución de partida	35
5.17. Límites impuestos a las corrientes normalizadas	35
5.18. Corrientes de la solución 1	36
5.19. Corrientes de la solución 2	36
5.20. Corrientes de la solución 3	36

Capítulo 2

Introducción

Desde la publicación de los estudios sobre el electromagnetismo de James Clerk Maxwell en 1873 el hombre ha estado *jugando* con los campos electromagnéticos, intentado *dominarlos* para su implementación en nuevas tecnologías, haciendo su vida más sencilla, o por lo menos ese era el objetivo. Esta teoría pasó muchos años escrita a tinta sin que ningún otro genio pudiera trasladarla a la vida real mediante la *creación artificial* de ondas radioeléctricas, hasta que sobre 1881 Heinrich Rudolf Hertz demostró experimentalmente lo que hasta entonces sólo era una teoría, que como veríamos más tarde iba a revolucionar el mundo. Desde entonces el testigo del electromagnetismo ha ido pasando de genio en genio, consiguiendo grandes avances como la primera comunicación por ondas radioeléctrica de larga distancia por Marconi, la invención de la antena propiamente dicha por parte de Aleksandr Popov o simplemente los múltiples inventos de Nikola Tesla en este sector.

No deja de asombrarnos la capacidad del hombre para interesarse por campos casi desconocidos. Al igual que hoy nos cuestionamos los orígenes del universo o investigamos sobre la composición de la materia mucho más allá del átomo, aquellos hombres fueron pioneros en el estudio de una rama de la física de la que se sabía muy poco. Aunque existen épocas en el que el trágico motor de la supervivencia acelera las investigaciones, un ejemplo claro es la invención y perfeccionamiento de la tecnología radar durante la segunda guerra mundial, en la que incluso se diseñaron los primeros arrays de guías ranuradas. En este punto de la historia se volvió indispensable la síntesis de arrays que pudiera aplicarse de forma sencilla y directa a la tecnología existente, para adelantar al enemigo. Ese razonamiento se trasladó al mundo empresarial iniciándose una guerra por la competitividad de las empresas, dónde continuó la necesidad de adelantar al *enemigo* que exigió a los ingenieros, no solo la aportación de ideas, sino que estas fuesen aportadas rápidamente permitiendo acelerar su implantación real y a ser posible en una tecnología ya existente, esta guerra que aun existe hoy día, trasladó la tecnología del radar militar a usos civiles tales como la detección de objetos en el subsuelo o la medida de la superficie del terreno.



Figura 2.1: Satélite Scatterometer para la medida del viento en alta mar (SeaWinds)

Tras la guerra, el afán conquistador y la curiosidad que define al hombre le llevo a ir al espacio, con el desarrollo de los satélites artificiales, la tecnología del radar sufrió un gran impulso, permitiendo la monitorización de datos de la superficie terrestre tales como la velocidad del viento a través del oleaje (*wind scatterometer*) o la detección de bloques de hielo flotant *Icebergs*. Ésta tecnología evolucionó permitiendo la medida de objetos cada vez más pequeños, como la vegetación e incluso la humedad del suelo. La evolución de estas aplicaciones aumentaron la complejidad de los diagramas de radiación necesarios, las formas eran cada vez más raras, se empezaron a exigir zonas con una pendiente muy específica y un rizado muy pequeño, esto acució la necesidad de soluciones sencillas a problemas complejos.

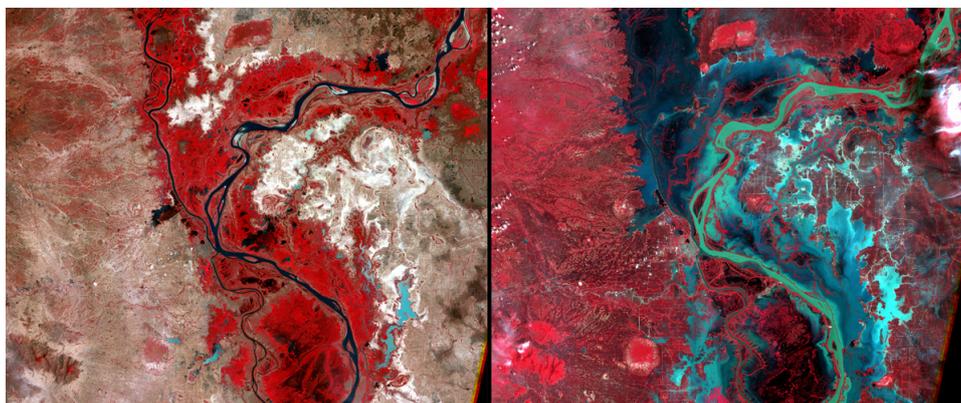


Figura 2.2: Medida de la humedad en una inundación en Camboya

El desarrollo del algoritmo de síntesis de arrays *alternating projection method* que se lleva a cabo en este trabajo pretende dar una solución a estas exigencias de la industria. El alma del proyecto reside en la capacidad del algoritmo para imponer límites a las corrientes generadoras del array, para así conseguir soluciones más sencillas de trasladar a la realidad que con otros métodos de síntesis. De esta forma se intenta hacer frente al monstruo que persigue a algunos ingenieros e investigadores, el monstruo de las ideas no acabadas, la imposibilidad de ver el resultado práctico de todo tu trabajo.

Capítulo 3

Objetivos

El objetivo principal de este trabajo final de grado es la implantación en código del método de síntesis de agrupaciones de antenas *alternating projection method*.

Para la consecución de dicho objetivo principal se han de conseguir los siguientes objetivos secundarios:

1. Estudio e investigación bibliográfica sobre la síntesis de agrupaciones de antenas, ya que no se nos ha ilustrado sobre este tema en ningún momento durante el grado.
2. Estudio bibliográfico sobre el método de síntesis en cuestión, para su posterior programación mediante la herramienta MATLAB.
3. Estudio e investigación profunda sobre las bases matemáticas del método en cuestión tales como la *Fast Fourier Transform* y el desarrollo en series de Fourier, así como su implantación práctica en la teoría de arrays.
4. Estudio superficial de la tecnología de arrays de ranuras en guías de onda resonantes, con el objetivo de comprender mejor la implantación real de los resultados teóricos obtenidos durante el trabajo.

3.1. Objetivos personales

Durante la realización de este trabajo final de grado me he planteado el objetivo de aumentar mi entendimiento en el área de antenas y arrays, sobre todo en el ámbito práctico o de aplicación de estos en la industria, ya que durante estos 4 años han sido pocas las asignaturas con aplicación directa en la vida real.

Capítulo 4

Metodología

En este apartado de proyecto se explica la organización con la que se ha abordado la realización del trabajo.

Durante los 4 años de grado no se nos ha explicado ningún método de síntesis de arrays, por lo que en primer lugar se realizó una amplia documentación sobre el método de síntesis en cuestión y sobre otros métodos. Posteriormente se empezó a plantear el desarrollo del algoritmo, llegados aquí caí en un punto muerto debido a las peculiaridades de los operadores de MATLAB, tales como la FFT, y su relación con el factor de array, debido a esto se realizó otra documentación sobre los principios matemáticos de la teoría de antenas y del método. Inevitablemente la documentación matemática se encontraba ligada a la de métodos de síntesis de agrupaciones de antenas, al ser un tema muy específico la búsqueda de material bibliográfica fue una ardua tarea.

Después de la documentación pertinente y una vez aclarados los conceptos se procedió a la síntesis del algoritmo. Una vez implementado se realizaron sendos estudios del algoritmo sintetizando agrupaciones de uso común en radiodeterminación, para así, comprobar su funcionamiento y poder realizar funciones adicionales que hiciesen más intuitivo el uso del algoritmo.

El objetivo principal del desarrollo de este método y no de otro era la posibilidad de alcanzar soluciones fáciles de implementar en las tecnologías de elementos radiantes actuales, por lo que se realizó una documentación sobre las agrupaciones de ranuras en guías de onda resonantes, para comprobar si realmente se podría alcanzar este objetivo.

Capítulo 5

Desarrollo.

5.1. Principios básicos de Agrupaciones de antenas.

Un array o agrupación de antenas está compuesto por un conjunto de elementos radiantes idénticos, ordenados regularmente, en el cual, las fases y amplitudes relativas de las corrientes que los alimentan permiten obtener un diagrama de radiación específico. Esto no solo permite la creación de infinitud de diagramas de radiación, sino que éste se puede cambiar dinámicamente realizando cambios en las corrientes de alimentación, sin cambiar los elementos radiantes ni su disposición.

El campo radiado por un array se puede obtener como la suma de los campos radiados por cada elemento aplicando el principio de superposición que deriva de las ecuaciones de Maxwell, por otro lado el campo radiado de cada elemento se puede expresar como el campo radiado por ese elemento en el origen multiplicado por un factor que incluye la diferencia de fase del campo con respecto al origen, que ha de tener en cuenta tanto el desfase de las corrientes con respecto a la de referencia como el espaciado, quedando:

$$E_{n(r\phi\theta)} = E_{0(r\phi\theta)} \frac{I_n}{I_0} e^{jKrr_0} \quad (5.1)$$

De esta forma al sumar todos los campos para obtener el patrón de radiación podemos sacar factor común el campo del elemento radiante $E_{0(r\phi\theta)}$ quedándose una suma de corrientes y desfases que corresponderá un nuevo parámetro denominado *factor de array*, este parámetro representa el diagrama de radiación que se obtendría si los elementos radiantes fuesen isotrópicos. De esta forma el patrón de radiación de un array se puede expresar de la siguiente forma:

$$E_{n(r\phi\theta)} = E_{0(r\phi\theta)} FA_{(r\phi\theta)} \quad (5.2)$$

En arrays muy grandes la variación de factor de radiación será mucho más *rápida* que la del diagrama del elemento radiante por lo que una buena aproximación del diagrama de radiación del conjunto podría ser el propio factor de radiación.

Como se ha explicado el funcionamiento del array se basa en la interacción de las ondas esféricas que radia cada elemento haciendo que la suma de éstas *interferencias* nos proporcione el patrón de radiación deseado, la creación de máximos o mínimos de radiación dependerá de la diferencia de fase existente entre los campos radiados, de esta forma definimos un nuevo espacio angular u , esta diferencia de fase se debe a dos motivos, el primero y más reconocible es el desfase producido por la diferencia de caminos, ya que las antenas están separadas, las ondas no recorren la misma distancia para llegar a un punto, el segundo es la propia diferencia de fase de la alimentación.

$$u = K d + \alpha \quad (5.3)$$

Existen tres tipos de arrays, los lineales, donde los elementos se disponen a lo largo de una recta, los arrays planos, donde los elementos se disponen sobre un plano, y los arrays conformados o tridimensionales, donde los elementos se disponen en las tres dimensiones del espacio, estos últimos suelen utilizarse para ajustar el array a la superficie donde se incorporará después, como puede ser el fuselaje de un avión. En este trabajo solo estudiaremos los arrays lineales ya que son los que se podrán sintetizar con el método de interés, aunque muchos arrays planos son separables en dos arrays lineales suponiendo que en una dimensión contamos con un array de elementos radiantes y en otra, permitidme la expresión, *un array de arrays*.

5.1.1. Arrays lineales.

Para facilitar el análisis y síntesis de arrays combine situarlos en el eje z del espacio tridimensional de este modo la simetría de revolución es fácil de plasmar matemáticamente y sólo depende de un ángulo, en concreto de θ , de esta forma el espacio u queda definido tal y como puede observarse en la expresión 5.5

$$FA_{(\theta,\phi)} = \sum_n I_n e^{j K r r_0} = \sum_n I_n e^{j K n d_z \cos(\theta)} = \sum_n I_n e^{j n u} \quad (5.4)$$

Al definir el factor de array de esta manera vemos como depende de las corrientes de alimentación y del nuevo espacio angular definido en 5.3 comprendido entre $[-\pi, \pi]$, cuya explicación física puede observarse gráficamente en 5.1.

$$u = K d \cos(\theta) \quad (5.5)$$

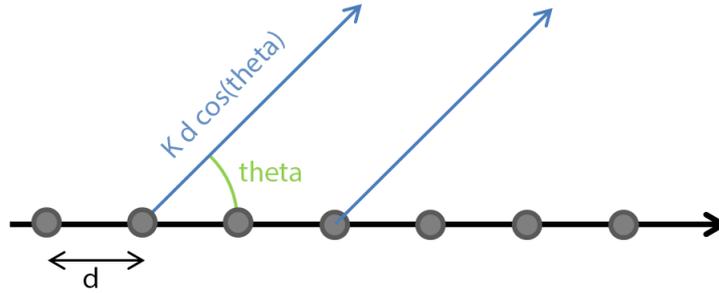


Figura 5.1: Esquema desfase

5.2. Principios básicos de ranuras como elemento radiante:

La simplicidad de su geometría, la eficiencia, su bajo coste de fabricación y su poco peso hace de los arrays de ranuras un opción muy atractiva para las empresas, si además sumamos a esto que el propio elemento radiante, en este caso la ranura, forma parte de su red de alimentación, la guía de onda, obtenemos de este tipo de agrupaciones una muy buena opción a tener en cuenta para implementar nuestro array, esto a hecho que muchas de las recientes tecnologías radar o aerotransportadas utilicen éste tipo de agrupaciones.

Solo aquellas ranuras realizadas en la pared de la guía que interrumpen las corrientes superficiales podrán radiar hacia el espacio libre, con lo cual, se ha de tener en cuenta la distribución de corrientes del modo propagado sobre la guía para la colocación de las ranuras. Para facilitar el estudio de la agrupación las ranuras se han clasificado según su forma y localización en la guía, gracias a esto el estudio se puede simplificar usando los circuitos equivalentes de cada ranura como se puede ver en la figura (5.2).

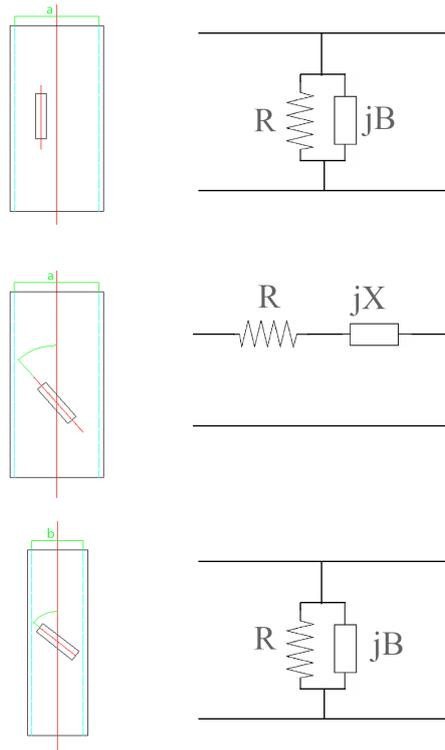


Figura 5.2: Circuitos equivalentes de ranuras

5.2.1. Agrupaciones lineales de ranuras en cavidades resonantes

Las características básicas de este tipo de agrupaciones son:

1. Todas las ranuras son resonantes en el array, lo que quiere decir que su susceptancia o reactancia equivalente es nula a la frecuencia central, al ser antenas resonantes presentan un estrecho margen de frecuencias de trabajo.
2. Las ranuras con voltaje máximo son las que se encuentran centradas en una cara, mientras que las ranuras con menor voltaje son aquellas que se encuentran desplazadas del centro, las ranuras se alternan a ambos lados del eje central para que estén en fase, de forma que dos ranuras contiguas en el mismo lado estarían a contra fase, podemos obtener varios valores de desfases aumentando o reduciendo la longitud de la ranura para que deje de ser totalmente resonante, obteniendo así una inductancia o capacitancia que desfasa la señal.

Dado que las ranuras han de ser resonantes el espaciado entre hechas ha de ser de $\lambda/2$ para encajar estas con los máximos de la onda estacionaria, si no fuese así las ranuras no serían totalmente resonantes y se tendría que realizar un estudio adicional. Para mantener la condición de resonancia y conseguir el perfecto acoplo a la frecuencia central y por tanto, conseguir que la apertura sea excitada adecuadamente y pueda radiar de forma adecuada a espacio libre se han de cumplir las siguientes condiciones:

1. La suma de todas las conductancias normalizada g_n de las ranuras ha de ser igual a 1 si la cavidad se ha alimentado por un extremo e igual a 2 si se ha alimentado por el centro, estos dos tipos de alimentación pueden observarse en la figura (5.3).
2. se ha de tener en cuenta que la conductancia de una ranura es proporcional a la potencia de radiación requerida.

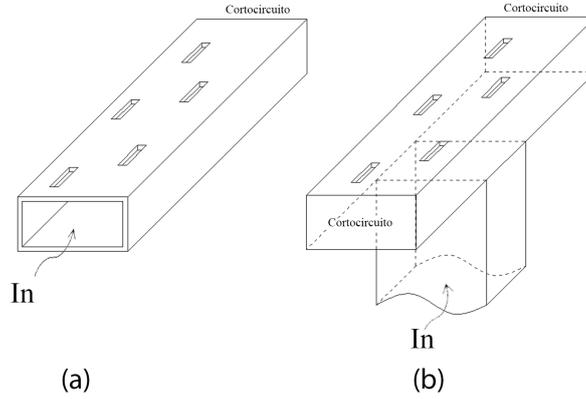


Figura 5.3: Alimentación de la guía resonante

Matemáticamente para un array de N ranuras tenemos:

$$\sum_{n=1}^N g_n = W \quad (5.6)$$

$$g_n = K A_{(n)}^2 \quad (5.7)$$

Donde g_n es la conductancia normalizada a la frecuencia de resonancia del elemento n -ésimo, $A_{(n)}$ es la distribución de voltaje de la apertura y K es la constante normalizada

Debido a la propia morfología de este tipo de array y sus elementos radiantes se nos imponen límites a la hora de implementar las corrientes teóricas, tanto en amplitud como en fase ya que las dimensiones de la guía son finitas. Es de aquí de donde subyace la necesidad de un algoritmo de síntesis que permita obtener unas corrientes fácilmente aplicables con la imposición de límites en módulo y desfases.

5.3. Principios del algoritmo *alternating projection method*:

El método *alternating projection* también conocido como *intersection approach* es un potente método para la síntesis de arrays basado en la teoría de intersección de conjuntos. El procedimiento considera que los patrones o factores de array radiados por una variedad específica de array (en este caso consideraremos solo arrays de antenas lineales y equiespacados) pueden agruparse en dos conjuntos. Un conjunto (B) engloba todos los posibles patrones que puede radiar el array, el subconjunto de (B) denominado (Bc) que engloba los patrones que cumplen con los límites impuestos a las corrientes y el subconjunto de (B) denominado (M) que engloba los patrones que cumplen con los límites impuestos al diagrama de radiación.

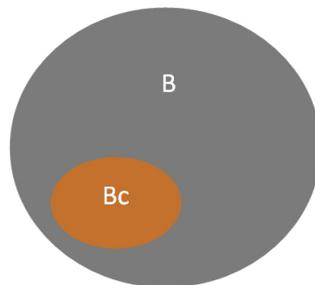


Figura 5.4: Subconjunto Bc

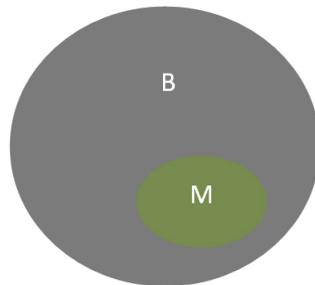


Figura 5.5: Subconjunto M

Las restricciones sobre el patrón de radiación se imponen mediante dos máscaras, una superior (Mu) que fuerza al diagrama a estar debajo de ella y otra inferior (Ml) que lo fuerza a estar por encima, aunque en el algoritmo creado en Matlab también se puede forzar mediante una máscara superior que nos marca el inicio de una recta a seguir y una pendiente que nos marca el ángulo de ésta, ya que ésta es la forma habitual de entregar cierto tipo de especificaciones en el ámbito profesional. El conjunto de patrones que cumple con estas máscaras se denominará conjunto (M).

Las soluciones que se encuentren en el subconjunto formado por la intersección de (M) y (Bc) es la solución final del problema.

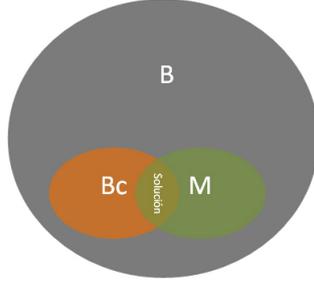


Figura 5.6: Solución

5.3.1. Funcionamiento del algoritmo *alternating projection method*:

Como su nombre indica este método realiza proyecciones para pasar alternativamente entre dos espacios: el espacio de las corrientes, donde obtenemos las corrientes de alimentación de los elementos radiantes y el espacio del factor de array donde obtenemos el patrón de radiación. Atendiendo a los conceptos teóricos, el operador de proyección que permite obtener el diagrama de radiación a partir de las corrientes de alimentación se puede observar en la expresión (5.8)

$$F(u) = \sum_{n=0}^{N-1} c_n e^{j k n d u} \quad (5.8)$$

Donde c_n es la corriente que alimenta del elemento radiante n-ésimo del array, j es el número complejo, k es el número de onda, d es la distancia de separación entre elementos radiantes contiguos y u es el espacio angular del factor de array definido en la expresión (5.5) y (5.9).

$$u \in [-\pi, \pi] \quad (5.9)$$

Realizando el cambio de variable (5.10) en la ecuación del factor de array (5.8) vemos como se obtiene el sumatorio descrito en la expresión (5.11) similar al que se realiza en el operador IDFT descrito en la ecuación (5.12).

$$p = \frac{N}{2\pi} K d u \quad (5.10)$$

$$F(p) = \sum_{n=0}^{N-1} c[n] e^{j n \frac{p 2\pi}{N}} \quad (5.11)$$

$$X[u] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{\frac{2\pi j}{N} n u} \quad (5.12)$$

De esta forma los dos operadores de proyección necesarios para movernos entre los espacios son *the Discrete Fourier Transform* (DFT) y *the Inverse Discrete Fourier Transform* (IDFT).

El algoritmo iterativo parte de una solución dada por cualquier otro método de síntesis F_i , siempre y cuando la solución resultante de éste sea lo suficiente cercana a la solución aplicable podremos imponerle límites, llegando a una solución F_{i+1} que puede ser válida o no, aplicando dos proyecciones de la siguiente forma:

$$F_{x+1} = P_c P_F F_x \quad (5.13)$$

P_F y P_c son descritas en las expresiones (5.14) y(5.15):

$$P_F = \begin{cases} Mu(u) \frac{F(u)}{|F(u)|} & \text{si } |F(u)| \geq Mu(u) \\ F(u) & \text{si } Ml(u) \leq F(u) \leq Mu(u) \\ Ml(u) \frac{F(u)}{|F(u)|} & \text{si } |F(u)| \leq Ml(u) \end{cases} \quad (5.14)$$

$$P_c = IDFT f_c f_N DFT \quad (5.15)$$

Donde P_c fuerza al factor de array a estar entre ciertos límites, f_N es un operador que restringe los términos de la serie de Fourier obtenida de la DFT del factor de array, de forma que haya tantos términos como antenas hay en el array y f_c al igual que P_c fuerza a los coeficientes de la serie a estar entre ciertos valores, tal y como muestra la expresión (5.16).

$$f_c c_n = \begin{cases} C_{MAX} \frac{c_n}{|c_n|} & \text{si } |c_n| \geq C_{MAX} \\ c_n & \text{si } C_{min}(u) \leq c_n \leq C_{MAX} \\ C_{min} \frac{c_n}{|c_n|} & \text{si } |c_n| \leq C_{min} \end{cases} \quad (5.16)$$

De una manera similar podemos forzar al desfase entre coeficientes a estar por encima de cierto valor mínimo, algo que adquiere gran importancia a la hora de implementar las soluciones en la vida real, ya que la tecnología actual nos impone ciertas restricciones en este ámbito.

Al imponer tantas restricciones es muy importante partir de una buena solución y saber dónde y cómo poner las máscaras de límites o el algoritmo no convergerá.

5.4. Consideraciones prácticas del algoritmo:

Una vez conocida la teoría profundizaremos un poco más en los cambios realizados al algoritmo para su implementación práctica en MATLAB.

En lugar de realizar transformadas discretas de Fourier usaremos transformadas rápidas de Fourier (FFT) implementada en MATLAB, que utiliza el algoritmo de Cooley–Tukey para el cálculo matricial de este operador aumentando la velocidad de la operación al reducir el número de operaciones necesarias para su cálculo, de N^2 operaciones complejas a $N \log_2(N)$.

Algoritmo Cooley–Tukey

Atendiendo a la ecuación (5.17) y realizandor el simple cambio de variable expresado en la ecuación (5.18) podemos vectorizar este operador.

$$X[k] = \sum_{n=-0}^{N-1} x[n] e^{-j 2 \pi (k n / N)} \quad (5.17)$$

$$W = e^{-j (2 \pi / N)} \quad (5.18)$$

La siguiente expresión ejemplifica la vectorización de una FFT de 4 puntos realizada por el algoritmo de Cooley–Tukey.

$$\begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{pmatrix} \times \begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{pmatrix}$$

La ecuación anterior se puede simplificar debido a que $W^{nk} = W^{\text{mod}(nk, N)}$

$$\begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^0 & W^2 \\ W^0 & W^3 & W^2 & W^1 \end{pmatrix} \times \begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{pmatrix}$$

La matriz W puede factorizarse en dos matrices, que puede ser descompuesta como el producto de 2 partes separables, obteniendose:

$$\begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{pmatrix} = \begin{pmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{pmatrix} \times \begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{pmatrix} \times \begin{pmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{pmatrix}$$

El resultado del producto de la primera matriz por el vector x da como resultado el vector x_1 .

$$\begin{pmatrix} x_1[0] \\ x_1[1] \\ x_1[2] \\ x_1[3] \end{pmatrix} = \begin{pmatrix} x[0] + W^0 x[2] \\ x[1] + W^0 x[3] \\ x[0] + W^2 x[2] \\ x[1] + W^2 x[3] \end{pmatrix}$$

Al final, la ecuación queda de la siguiente forma:

$$\begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{pmatrix} = \begin{pmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{pmatrix} \times \begin{pmatrix} x_1[0] \\ x_1[1] \\ x_1[2] \\ x_1[3] \end{pmatrix}$$

Como se puede observar se ha reducido mucho el número de operaciones aprovechando la simetría y *separabilidad* de las matrices

Este operador introduce ciertas consideraciones a la hora de implementar nuestro algoritmo en código. Las corrientes correspondientes a cierto factor de array se calcula como la FFT de éste, obteniendo así los coeficientes complejos del desarrollo en serie de fourier.

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_1 t} \quad (5.19)$$

Al realizar este desarrollo en series de Fourier hemos supuesto que la señal $f(t)$ que en nuestro caso sería el factor de array es periódica con periodo 2π , hasta el momento no existe ningún problema ya que como definimos anteriormente el factor de array se encuentra definido entre $[-\pi, \pi]$ y en realidad éste sí que es 2π periódico, lo que se ha de tener en cuenta para espaciados de más de $\lambda/2$ puesto que el margen visible $[-kd, kd]$ será mayor que un periodo de u .

Esos coeficientes que representan las corrientes se nos presentan ordenados de una determinada manera, en concreto y al aplicar una FFT de Nt puntos se obtiene:

n	1	2	3	4	...	N_{t-2}	N_{t-1}	N_t
$F(n)$	C_0	C_1	C_2	C_3	...	C_{-3}	C_{-2}	C_{-1}

Cuadro 5.1: Desarrollo en series de Fourier realizado con una FFT

Como podemos observar, las primeras posiciones del vector corresponden en primer lugar al coeficiente de continua y vienen seguido en orden ascendente de los coeficientes complejos correspondientes a las frecuencias positivas y en sentido contrario y desde el final del vector encontramos los coeficientes complejos correspondientes a las frecuencias negativas. En nuestro caso y como se ha mencionado en el apartado anterior solo podemos tener N coeficientes correspondientes a los N elementos radiantes que constituirán nuestro

array, solo tiene sentido quedarse con las primeras N posiciones del array correspondientes al coeficiente de continua y a los N-1 coeficientes de frecuencia positiva.

Por otra parte, las ecuaciones (5.16) y (5.14) no son aplicables en un estudio real dado que cuando el módulo es muy pequeño al dividir por éste, MATLAB redondea a infinito, por lo que se han implementado como muestran las expresiones (5.20) y (5.21).

$$P_{F(F(u))} = \begin{cases} Mu(u) e^{j \text{angle}(F(u))} & \text{si } |F(u)| \geq Mu(u) \\ F(u) & \text{si } Ml(u) \leq F(u) \leq Mu(u) \\ Ml(u) e^{j \text{angle}(F(u))} & \text{si } |F(u)| \leq Ml(u) \end{cases} \quad (5.20)$$

$$f_c(c_n) = \begin{cases} C_{MAX} e^{j \text{angle}(c_n)} & \text{si } |c_n| \geq C_{MAX} \\ \cdot c_n & \text{si } C_{min}(u) \leq c_n \leq C_{MAX} \\ C_{min} e^{j \text{angle}(c_n)} & \text{si } |c_n| \leq C_{min} \end{cases} \quad (5.21)$$

De esta forma cuando el módulo de la corriente o del factor de array se encuentra por debajo de la cota inferior, conservaremos la fase en ese punto y cambiaremos el módulo por el valor mínimo, evitando el problema comentado anteriormente.

El funcionamiento básico del algoritmo puede observarse en la figura (5.7), este diagrama sólo representa el funcionamiento general de código y especialmente el método de interrupción del bucle infinito ya que el funcionamiento general se ha explicado en líneas generales anteriormente.

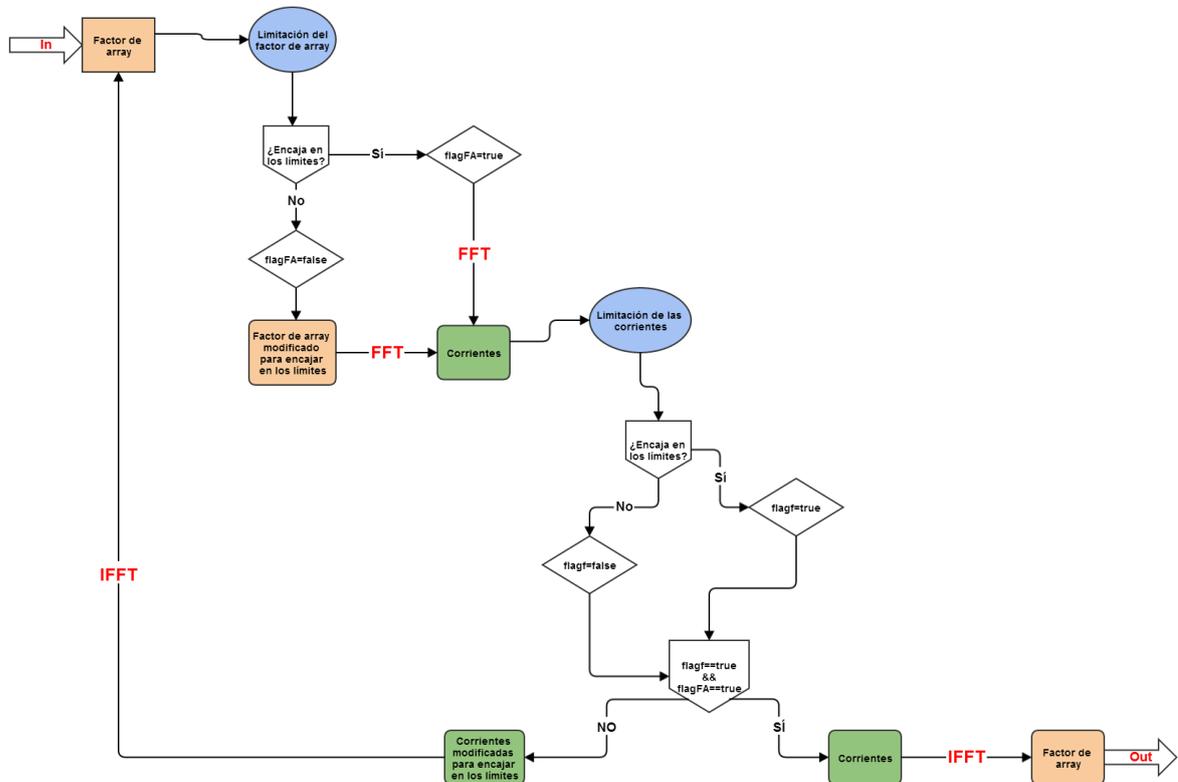


Figura 5.7: Diagrama de flujo del algoritmo desarrollado

5.5. Consideraciones del código:

En este apartado se tratará de explicar las peculiaridades del código implementado, ya que el carácter generalista del método impone una serie de dificultades en su implementación y hay que tener en cuenta ciertas características propias del código para su buen funcionamiento:

¿Cómo introducir los límites? Los límites del factor que se impondrán en el factor de array tanto superior como inferior se introducirán en unidades logarítmicas y han de formar un vector del mismo tamaño que el factor de array, de esta forma se pueden imponer tantos como se deseen, pero se presenta el primer problema ¿Que hacer si no quieres imponer cierto límite en alguna zona? Si deseas que se obvие cierto límite solo tienes que imponerle el valor $-100dB$ de esta forma el código obviara ese límite en esa zona, esta peculiaridad se explica gráficamente en la figura (5.8).

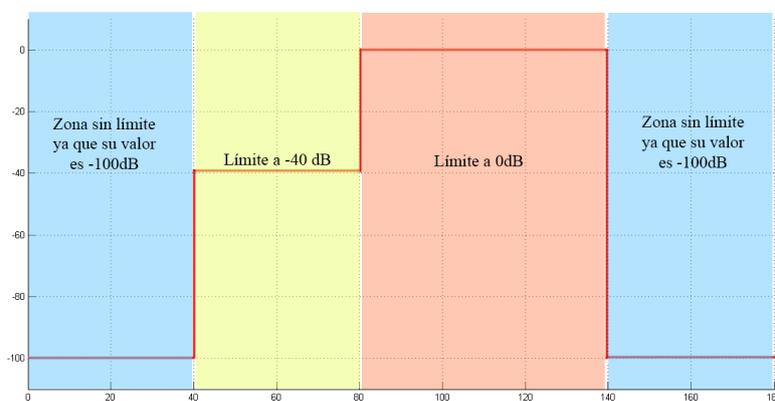


Figura 5.8: Formato de entrada de los límites

Los límites de corrientes son más simples, son tres valores: un valor de módulo máximo, otro de módulo mínimo y un valor de desfase mínimo entre elementos adyacentes.

Funciones adicionales Se han implementado dos funciones adicionales para hacer más fácil el uso del método. La primera es la función llamada: *CambioEspacio* que se encuentra en el anexo (7.0.7) y permite cambiar automáticamente del espacio angular del factor de array definido en la expresión (5.3) al espacio real comprendido entre 0 y 180 usando el parámetro kd que es el resultado del producto del número de onda por el espacio de los elementos del array, de esta forma intento hacer más fácil el uso del mi algoritmo.

Otra función que resulta útil es: *RatonAuto* situada en el anexo (7.0.7). Esta función se ha de iniciar, siempre, después de una representación en pantalla del diagrama de partida en el espacio real, representado entre 0 y 180, esta función permite dibujar gráficamente los límites superiores o inferiores del array evitando así el tedioso trabajo de buscar la correspondencia entre ángulos del espacio y posiciones de los vectores.

5.6. Introducción a Matlab.

Dado que el lenguaje seleccionado para la implementación de este algoritmo ha sido MATLAB no puedo evitar dar un pequeño paseo por su historia y sus características, ya que describen muy bien la elección de este lenguaje y GUI.

Un poco de historia:

En 1984 *Cleve Moler* crea la primera versión con la idea de emplear paquetes de subrutinas escritas en Fortran en los cursos de álgebra lineal y análisis numérico, sin necesidad de escribir programas completos en dicho lenguaje. Sería más tarde, en 1970, cuando se crearía el lenguaje M con la intención de proporcionar un acceso más sencillo al software LINPACK y EISPACK (librería para el cálculo numérica de Fortran) sin tener que usar Fortran.

MATLAB es un lenguaje de alto nivel y un entorno para el cálculo numérico, además de un entorno de desarrollo integrado (IDE). Mediante MATLAB, es posible analizar datos, desarrollar algoritmos y crear modelos o aplicaciones.

El lenguaje, las herramientas y las funciones matemáticas incorporadas permiten llegar a una solución antes que con hojas de cálculo o lenguajes de programación tradicionales.

El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, Simulink (entorno de programación y simulación visual) y GUIDE (editor de interfaces de usuario - GUI).

Conviene mencionar que MATLAB es un programa de cálculo numérico diseñado para trabajar con matrices. Por tanto, será más eficiente si se diseñan los algoritmos en términos de matrices y/o vectores o se usan sus funciones propias, ya que se implementan matricialmente. Esta cualidad lo convierte en el programa perfecto donde implementar este algoritmo iterativo, ya que se han de realizar operaciones complejas como son la FFT e IFFT reiteradas veces.

5.7. Síntesis de arrays realizadas con el algoritmo:

Una vez explicados los conceptos teóricos y su implementación en el entorno *MATLAB* realizaremos la síntesis de algunos arrays ampliamente usados en sistemas radares o de scatterometría, el objetivo de éste apartado surge del deseo de todo ingeniero de ver los resultados de su trabajo, así como la comprensión de los factores que influyen en la convergencia del algoritmo.

5.7.1. Síntesis de un array con lóbulo principal plano:

Muchas aplicaciones necesitan un patrón con ganancia plana en cierta zona del espacio y un cierto nivel de lóbulo principal a secundarios, en éste ejemplo se realiza la síntesis de un array de 16 elementos espaciados $\lambda/2$ con lo que el margen visible definido entre $[-kd, kd]$ abarcará todo el espacio del factor de array, $-\pi$ y π , éste diagrama se encuentra en el artículo [1] de la bibliografía.

Descripción	Ángulo inicio	Ángulo final	Especificaciones
Lóbulo secundario	0	50	-20 dB
Lóbulo secundario	128	160	-30 dB
Lóbulo principal	64	115	$0\text{ dB} \pm 0,5\text{ dB}$
Lóbulo secundario	167	180	-20 dB

Cuadro 5.2: Especificaciones del diagrama

Para la síntesis de éste diagrama se usó como punto de partida el factor de array obtenido de una síntesis que ofrece como solución unas corrientes de módulo constante y fase variable y a la que se le introdujo como *máscara* los siguientes valores que pueden ser observados gráficamente en la figura (5.10):

Máscara:

Ángulo inicial	Ángulo final	Valor
0	63	-20 dB
116	167	-30 dB
64	117	0 dB
167	180	-20 dB

Cuadro 5.3: Máscara de conformación de patrón

El algoritmo partió de esta solución y en éste caso no se le impuso ninguna restricción a las corrientes, mientras que al patrón de array se le impuso como límite superior los especificados en la definición del diagrama de la tabla (5.2) así como un límite inferior en la zona del lóbulo principal a $-0,5\text{ dB}$ para cumplir con el límite de rizado de $\pm 0,5\text{ dB}$ de forma holgada, tal y como se puede ver en la figura (5.11) y en la ampliación del lóbulo principal de la figura (5.12). Se puede observar cómo se le han impuesto unos límites más estrictos de lo necesario, durante los estudios realizados con éste método la práctica me ha enseñado que, en general, ha de ser así si quieres cumplir con las especificaciones.

EL resultado obtenido tras 5000 iteraciones se puede observar junto a la solución sugerida en el artículo [1] de la bibliografía en la figura (5.9) y en contraste con la solución de partida obtenida del método *solofase* en la figura (5.13), el avisado lector podrá observar como aun partiendo de una solución muy alejada de lo deseado podemos obtener un patrón que cumple prácticamente con todas las restricciones impuestas en las especificaciones, las corrientes obtenidas como solución y que por tanto generan ese patrón son:

n	1	2	3	4	5	6	7	8
abs(Cn)	1	1,37	2,27	3,26	2,78	3,22	2,48	0,51
n	9	10	11	12	13	14	15	16
abs(Cn)	0,29	2,28	3,50	3,06	3,22	2,36	1,08	0,86

Cuadro 5.4: Módulo de corrientes normalizadas

n	1	2	3	4	5	6	7	8
angle(Cn) [deg]	0	-58,04	-128,14	-163,13	144,38	89,94	75,94	123,48
n	9	10	11	12	13	14	15	16
angle(Cn) [deg]	113,90	60,84	77,76	124	172,25	-151,30	-91,11	-47,89

Cuadro 5.5: Ángulo de corrientes normalizadas

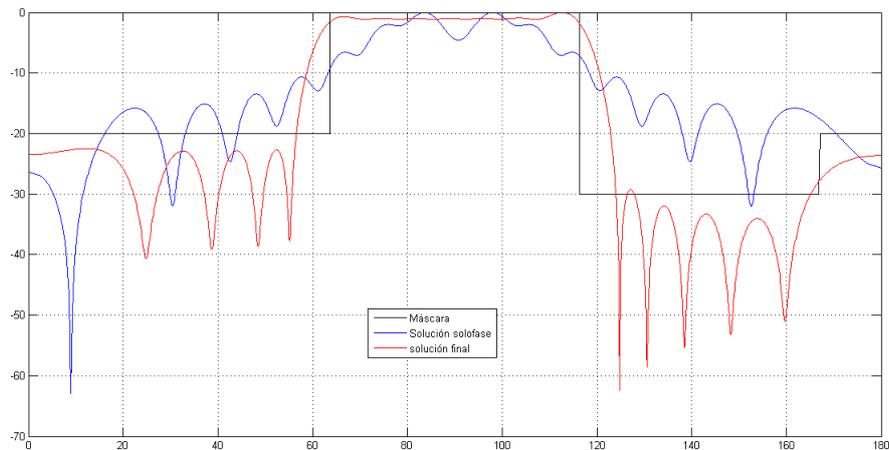


Figura 5.9: Soluciones final e inicial

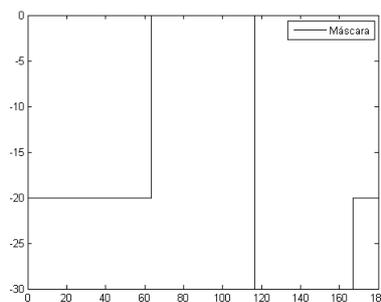


Figura 5.10: Máscara

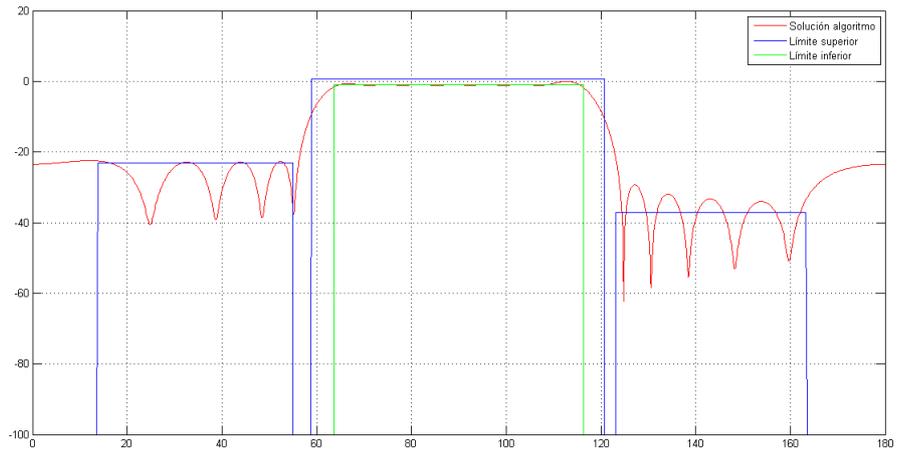


Figura 5.11: Solución final con límites

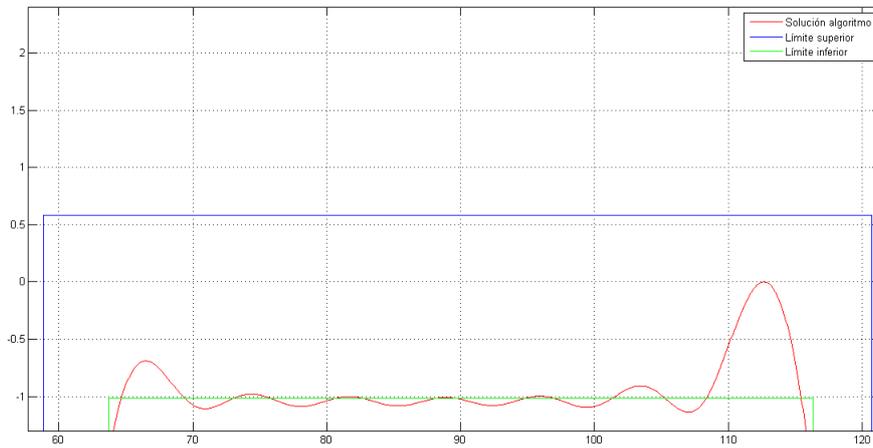


Figura 5.12: Solución final con límites, lóbulo principal

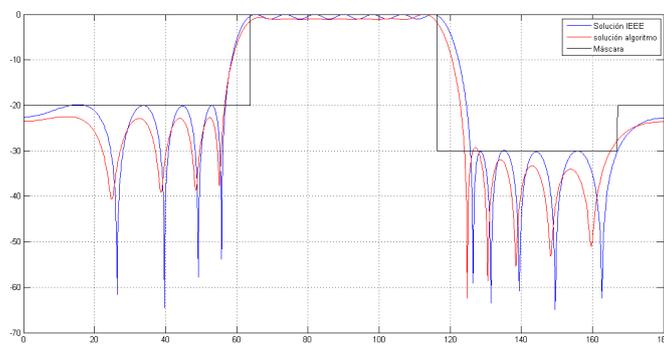


Figura 5.13: Soluciones final y solución del artículo [1]

Ahora se procederá a realizar la síntesis del mismo array anteriormente descrito en

la tabla (5.2) partiendo de una solución que cumple ya con las especificaciones, para intentar alcanzar una solución con unos desfases entre elementos adyacentes mucho más fáciles de implementar en la práctica. La solución se encuentra en el artículo [1] de la bibliografía. Las corrientes de partida pueden observarse en la tabla (5.6) o en el artículo citado anteriormente:

n	1	2	3	4	5	6	7	8
$Abs(Cn)$	0,99	1,46	1,18	1,42	1,12	2,26	2,35	1,29
n	9	10	11	12	13	14	15	16
	1,32	2,32	2,38	1,12	1,42	1,3	1,3	1

Cuadro 5.6: Módulo de corrientes normalizadas

n	1	2	3	4	5	6	7	8
$Angle(Cn)[deg]$	-7,6	23,2	78,4	138,7	-138,7	-73,8	-54,8	-27,3
n	9	10	11	12	13	14	15	16
	-25,7	-52,9	-76,8	-141,3	134,7	76,1	13,4	0

Cuadro 5.7: Fase de corrientes normalizadas

El factor de array resultante de estos 16 elementos puede observarse en la figura (5.14).

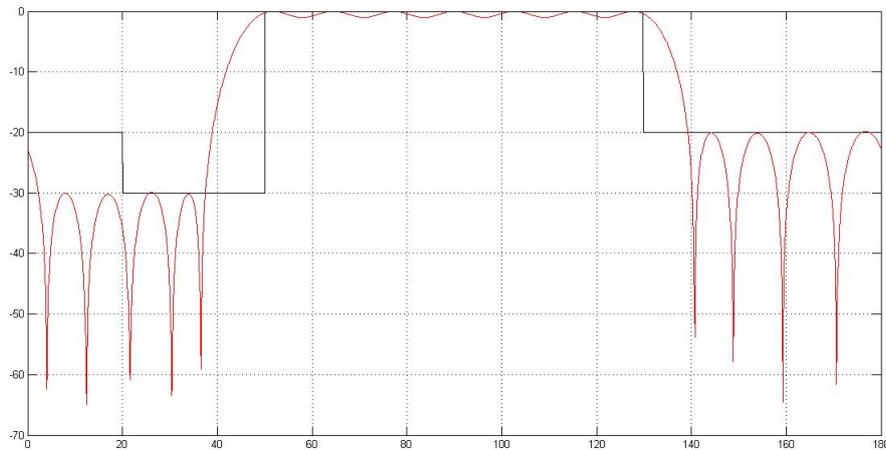


Figura 5.14: Solucion propuesta en el artículo [1]

Aunque ésta es una solución válida lo que se pretende es conseguir otra solución que cumpla con las especificaciones del factor de array y tenga unos desfases de corrientes entre elementos menores, ya que en ésta se llegan a obtener desfases de hasta 277 grados sexagesimales difícilmente implementables en la práctica. Para éste estudio puesto que vamos a limitar el desfase de corrientes a 150 grados e imponemos un módulo de corriente máxima normalizada de 5, he impuesto unos límites menos restrictivos al factor de array, tal y como se pueden observar en la figura (5.15). La solución obtenida puede observarse y compararse con el factor de array de partida (Solución propuesta en el artículo [1]) en la figura (5.16)

Las corrientes de alimentación que generan éste patrón de radiación se pueden observar en las tablas (5.8) y (5.9).

n	1	2	3	4	5	6	7	8
abs(cn)	1,00	0,67	0,53	1,33	2,90	3,21	4,68	4,90
n	9	10	11	12	13	14	15	16
abs(cn)	2,39	3,53	3,95	1,24	1,43	0,98	0,32	0,55

Cuadro 5.8: Módulo de corrientes normalizadas con límites de desfase

n	1	2	3	4	5	6	7	8
angle(cn) [deg]	0,000	-2,893	65,445	58,953	53,748	82,963	128,112	146,341
n	9	10	11	12	13	14	15	16
angle(cn) [deg]	124,178	17,756	-2,657	-32,680	-154,950	-166,975	-7,564	-3,903

Cuadro 5.9: Fase de corrientes normalizadas con límites de desfase

En éste caso el desfase máximo entre corrientes de elementos contiguos es de 160 grados sexagesimales, en un par de puntos, estos desfases son mucho más fáciles de implementar pero las amplitudes son algo mayores. En todo estudio tenemos que adquirir un compromiso entre desfase y amplitudes.

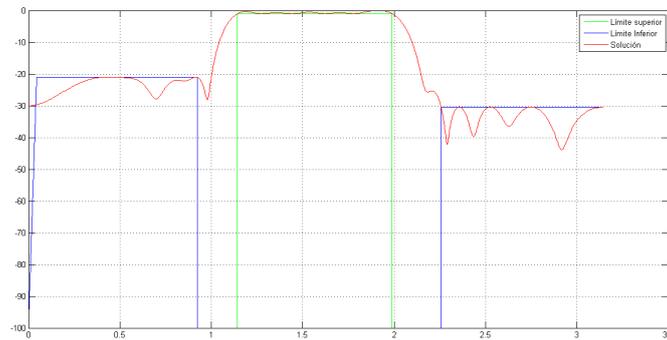


Figura 5.15: Límites impuestos junto a solución con límites de corrientes

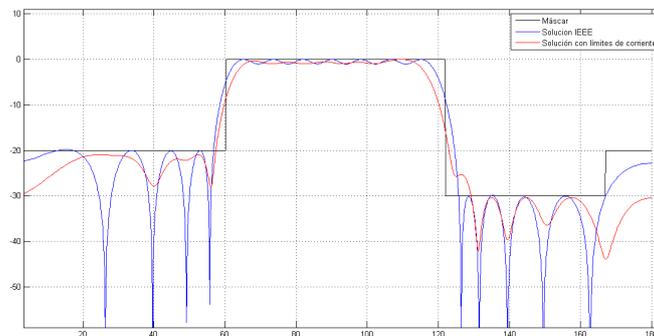


Figura 5.16: Solucion con límites de corrientes junto a solución propuesta en el artículo [1]

5.7.2. Síntesis de un array de lóbulo principal inclinado con cierta pendiente:

Una gran cantidad de aplicaciones radar, normalmente aquellas que irán montadas sobre aviones o satélites, necesitan que el lóbulo principal de patrón de radiación tenga cierta pendiente, normalmente, para suplir el efecto del ángulo de apuntamiento o la propia curvatura de la tierra. En la industria esta imposición se suele expresar con una cierta pendiente en $[dB/deg]$ sobre cierta zona del patrón y una cierta variación de esta y/o un cierto rizado, en función de las necesidades de la tecnología a implementar.

El array tal y como se puede observar en la figura (5.17) está formado por 4 zona. En la zona I se encuentran lóbulos secundarios, el cual han de estar por debajo de -20 dB en el diagrama normalizado. En la zona II se encuentran más lóbulos secundarios que, al estar más cerca del lóbulo principal, han de estar por debajo de -30 dB. En la zona III se encuentra el lóbulo principal, el cual ha de tener una pendiente de alrededor de $-0,354dB/deg$. En la zona IV se encuentran los últimos lóbulos secundarios que han de estar por debajo de -20, ésta última zona ha sido la que más problemas ha dado, ya que , como se ve en la figura (5.18), el límite impuesto está muy cerca del valor final del array en la zona III, la correspondiente al lóbulo principal, por lo que el array tiende a continuar con la pendiente, haciendo que el límite no se cumpla en una pequeña parte, la única forma de cumplir con éste límite era imponer uno mayor a la hora de sintetizar el diagrama para conseguir romper con esa tendencia, claro, a costa de aumentar la complejidad de las corrientes.

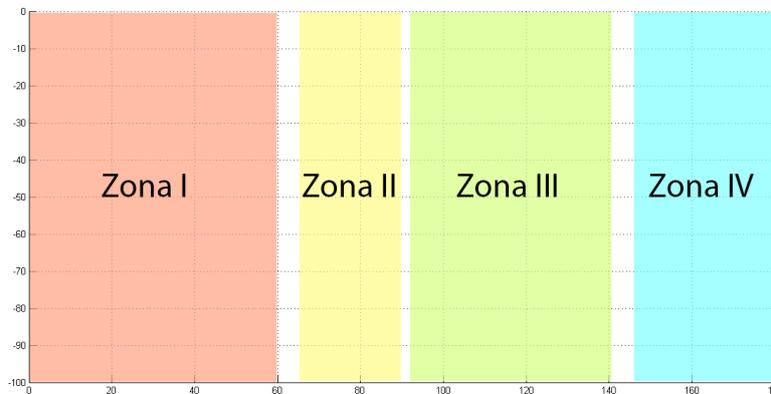


Figura 5.17: Zonas en las que se define el patrón de radiación

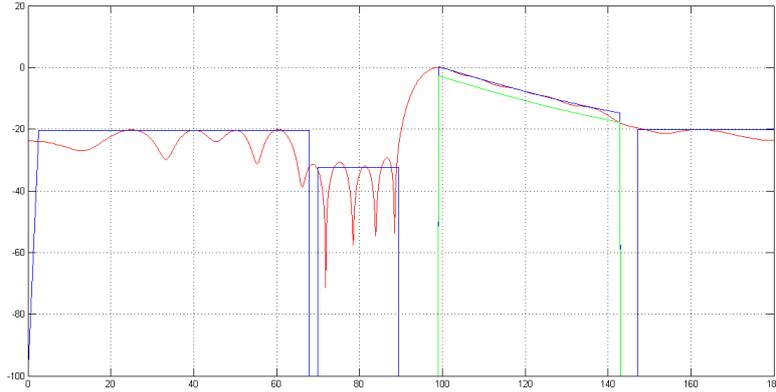


Figura 5.18: Solución con límites no estrictos

Se han realizado dos estudios de éste array, con una agrupación de 16 antenas y espaciado $\lambda/2$, a continuación muestro el análisis y los resultados de estos dos estudios, así como sus diferencias.

Solución 1:

En éste estudio se ha realizado la síntesis del diagrama descrito anteriormente partiendo del metodo de síntesis de *Solofase*, el patrón de radiación resultante de 1000 iteraciones del algoritmo puede verse en la figura (5.20), los límites impuestos al factor tal y como pueden verse en la figura (5.21) son un poco más estrictos a las especificaciones requeridas, ya que las soluciones resultantes de éste método suelen oscilar un poco alrededor de los límites, y aunque estas oscilaciones son muy pequeñas y pueden ser aceptables, he preferido exigir más para cumplir perfectamente con las especificaciones. En la figura (5.18) podemos ver el resultado con unos límites menos estrictos, se puede observar como en su mayor parte, se cumplen los objetivos, pero en las zonas iniciales y finales de los tramos el diagrama oscila un poco y no llega a cumplir el todo, debido al conocido fenómeno de Gibbs.

En las tablas (5.10) y (5.11) podemos ver el módulo y fase de las corrientes obtenidas como solución, las amplitudes son facilmente implementables, con una relación $I_{max}/I_{min} = 4$, pero las fases son más difíciles de implementar, debido a que existen grandes desfase entre corrientes. Como se puede observar en la tabla (5.12) existen desfases de hasta 300 grados sexagesimales, esto es debido a que no se le han impuesto límites de fase a las corrientes durante la síntesis, aunque sí se le ha impuesto un límite superior al módulo de 5 (valor normalizado).

n	1	2	3	4	5	6	7	8
abs(cn)	1,00	1,34	1,91	2,81	3,19	3,84	4,00	4,00
n	9	10	11	12	13	14	15	16
abs(cn)	4,00	3,35	2,36	1,84	1,86	2,72	2,96	2,04

Cuadro 5.10: Módulo de corrientes normalizadas

n	1	2	3	4	5	6	7	8
angle(cn) [deg]	0,000	21,393	71,249	87,959	127,177	165,651	-170,505	-129,944
n	9	10	11	12	13	14	15	16
angle(cn) [deg]	-89,732	-55,902	-11,469	66,472	-148,697	-54,063	16,661	114,404

Cuadro 5.11: Ángulo de corrientes normalizadas

desfas [deg]	21,39	49,86	16,71	39,22	38,47	-336,16	40,56
40,21	33,83	44,43	77,94	-215,17	94,63	70,72	97,74

Cuadro 5.12: Desfase entre corrientes normalizadas contiguas

El resultado final puede observarse en la figura (5.22) en comparación de la solución sugerida, como podemos ver el resultado es un factor de array muy similar al sugerido, incluso con mejores NLPS, debido sin duda a la imposición de límites más estrictos, también se ha conseguido un rizado mucho menor al sugerido en el lóbulo principal, aunque éste se ha ensanchado un poco, Siendo éste ensanchamiento totalmente aceptable se puede rectificar alargando el límite de la zona II, aunque seguramente se traduzca en corrientes más complejas.

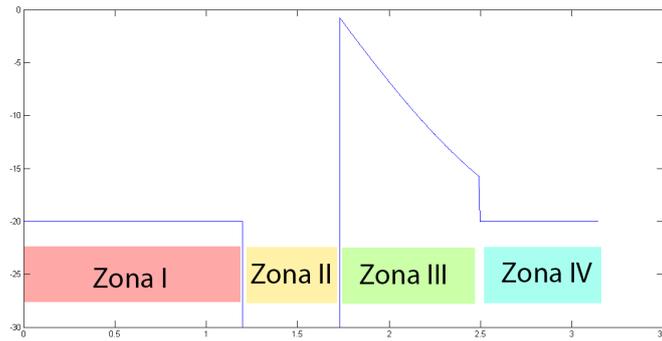


Figura 5.19: Máscara de inicio para síntesis solofase

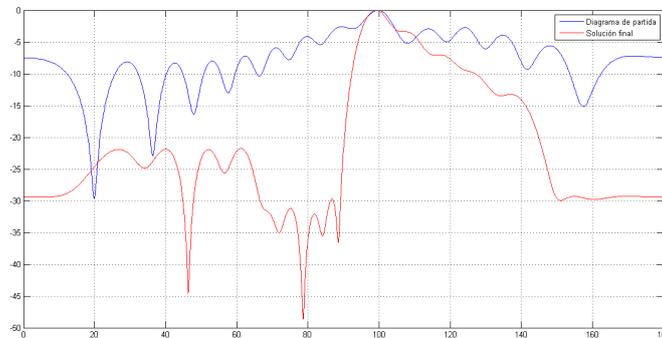


Figura 5.20: Patrón de partida y Solución final

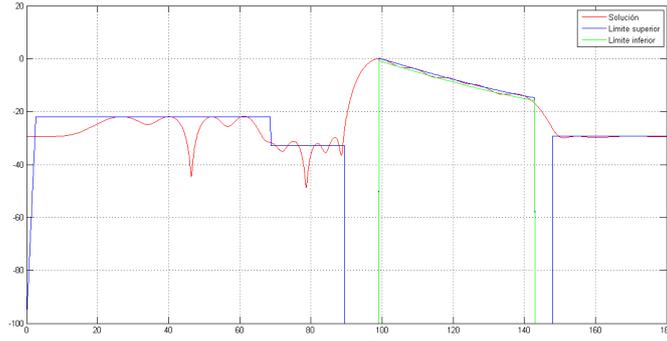


Figura 5.21: Límites impuestos al patrón de radiación

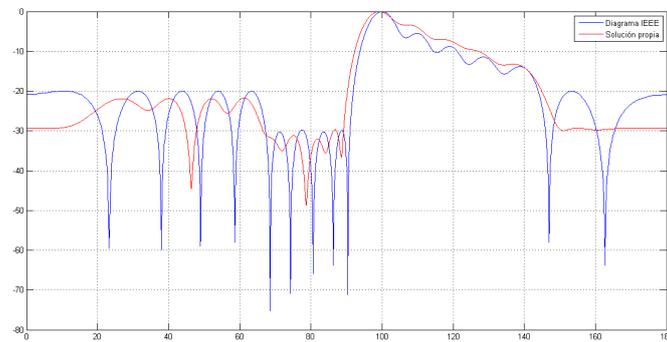


Figura 5.22: Solución final comparada con la solución sugerida en el artículo [1]

Solución 2:

Una vez visto los resultados del estudio anterior, dónde los desfases de corrientes entre elementos contiguos dificultaban su implementación práctica, se propone la solución de la siguiente síntesis, donde aunque no se han impuesto límites de corrientes, se imponen unos límites al factor de array menos estrictos, sobre todo en la zona del lóbulo principal, donde se ha permitido un mayor rizado, con el objetivo de reducir la complejidad de las corrientes sin necesidad de imponer un límite. La síntesis parte del patrón obtenido al aplicar el algoritmo (*Solofase*) del apartado anterior, por lo que el diagrama de partida es el mismo, ya que se le ha introducido la misma máscara, la solución de partida y la solución final se pueden observar en la figura (5.24).

Los límites impuestos al factor de array se pueden observar gráficamente en la figura (5.23).

Las corrientes obtenidas no son mucho mejores que las anteriores, como se pueden ver en las tablas (5.13) y (5.14), la relación I_{MAX}/I_{min} sigue siendo 4 y el desfase máximo entre corrientes de elementos contiguos se mantiene sobre 300 grados sexagesimales tal y como se puede observar en la tabla (5.15).

Con éste estudio pretendo plasmar la necesidad de imponer límites al desfase de las

corrientes en el propio método, ya que aunque seamos más flexibles en las especificaciones, la solución puede ser igual de compleja y por lo tanto difícilmente aplicable, también intento plasmar la importancia de la colocación de los límites, con unos pequeños cambios en estos el diagrama cambia mucho.

n	1	2	3	4	5	6	7	8
abs(cn)	1,00	1,72	1,95	2,82	3,58	3,80	4,00	4,00
n	9	10	11	12	13	14	15	16
abs(cn)	4,00	3,98	2,94	2,53	2,01	2,60	2,42	2,25

Cuadro 5.13: Módulo de corrientes normalizadas

n	1	2	3	4	5	6	7	8
angle(cn) [deg]	0,000	40,551	79,656	97,106	139,065	173,204	-163,380	-129,153
n	9	10	11	12	13	14	15	16
angle(cn) [deg]	-96,861	-60,543	-20,591	42,291	151,139	-80,461	-18,434	78,071

Cuadro 5.14: Fase de corrientes normalizadas

desfas [deg]	40,55	39,11	17,45	41,96	34,14	-336,58	34,23
32,29	36,32	39,95	62,88	108,85	-231,60	62,03	96,50

Cuadro 5.15: Desfase entre corrientes normalizadas de elementos contiguos

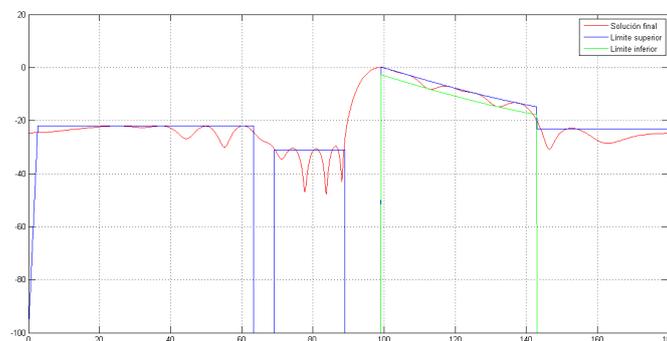


Figura 5.23: Límites impuestos al patrón de radiación

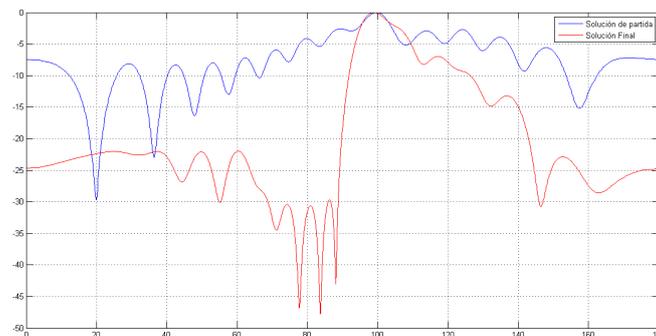


Figura 5.24: Patrón de partida y Solución final

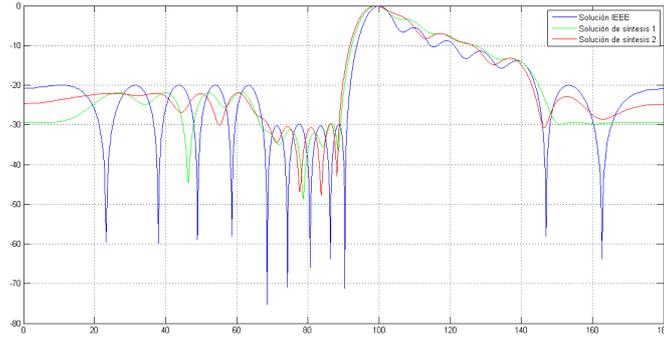


Figura 5.25: Solución final con y sin límites en el desfase comparadas con la solución sugerida en el artículo [1]

5.7.3. Síntesis de un array con límites en el desfase de corrientes:

En el siguiente estudio, se sintetizará un array con un espaciado entre elementos radiantes mayor de $\lambda/2$, en nuestro caso es de $0,67178\pi$, esto implica ciertas dificultades ya que el margen visible es mayor que el espacio del factor de array y tendríamos que aplicar la propiedad de periodicidad de éste. Este cambio de espacio se ha realizado automáticamente gracias a la función auxiliar, propia *CambioEspacio* del anexo (7.0.7), el cambio se puede observar gráficamente en un ejemplo no real en la figura (5.26).

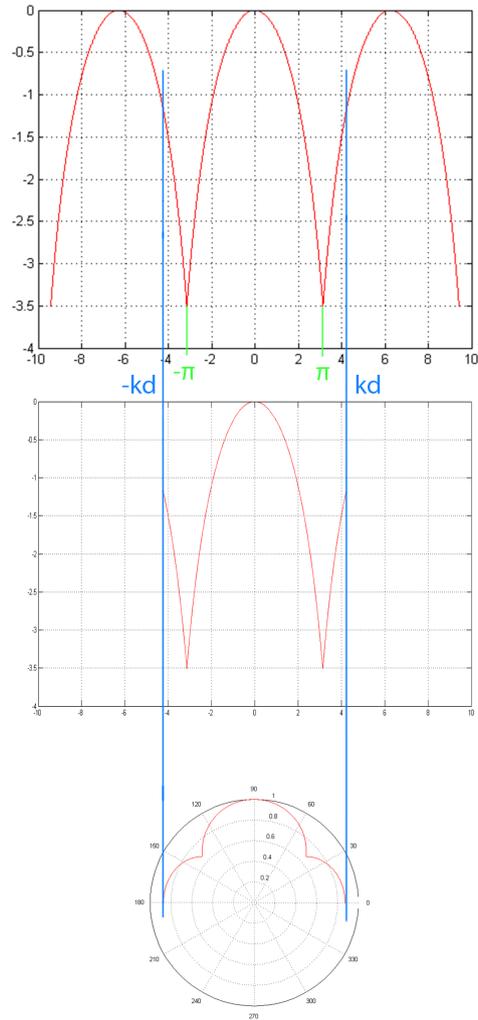


Figura 5.26: Cambio de espacio desde el factor de array al espacio real

La síntesis parte de una solución acotada perfectamente dentro de las especificaciones del factor de array pero cuyas corrientes son demasiado complejas como para realizar una implementación sencilla en una agrupación de ranuras, por lo que se recurre al algoritmo para conseguir una solución válida y fácilmente implementable.

Las corrientes normalizadas del factor de array tomadas como punto de partida se pueden observar en la tabla (5.16), estas corrientes presentan una relación I_{MAX}/I_{min} relativamente alta y un desfase entre elementos contiguos de hasta 230 grados sexagesimales, el objetivo de esa síntesis es reducir este desfase máximo manteniendo en la manera de lo posible el margen dinámico I_{MAX}/I_{min} y cumpliendo con las especificaciones del diagrama.

$abs(c_n)$	2,76	5,85	5,98	2,24	1,65	1,58	0,36	1
$angle(c_n)$	-1,351	-1,229	-1,149	-1,265	2,792	2,548	0,696	0
$angle(c_{n+1}) - angle(c_n)$	6,97	4,60	-6,62	232,42	-13,97	-106,13	-39,85	NaN

Cuadro 5.16: Corrientes normalizadas solución de partida

Los límites impuestos por las especificaciones al factor de array se pueden observar en la figura (5.27) junto a la solución de partida, que como se puede observar cumple con holgura los límites, por otro lado, los límites impuestos a las corrientes pueden observarse en la tabla (5.17)

abs(I.MAX)	7
abs(I.MIN)	0,1
Desfase máximo [deg]	100

Cuadro 5.17: Límites impuestos a las corrientes normalizadas

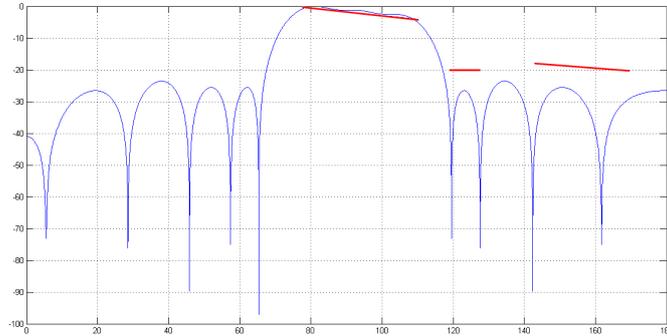


Figura 5.27: Especificaciones (rojo) y solución de partida (Azul)

Se han obtenido 3 soluciones válidas e implementables, cuyos resultados se pueden ver en la figura (5.28).

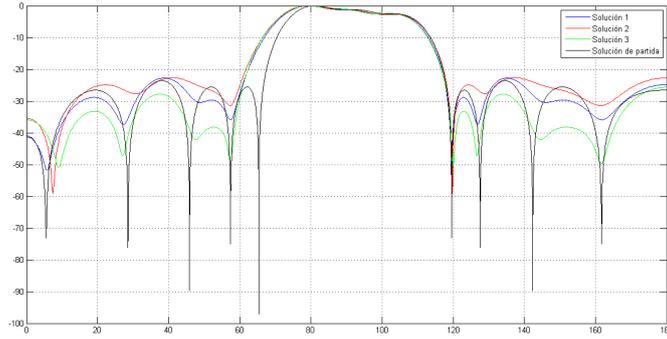


Figura 5.28: Distintas soluciones del algoritmo junto al diagrama de partida

Como el lector puede observar, en todas las soluciones, el lóbulo principal se ha ensanchado, esto se debe a que hemos limitado las corrientes y no se le han impuesto muchos límites a nuestro diagrama, se puede conseguir una lóbulo más estrecho si se le impone ese límite, pero dado que no se especifica ningún límite de ancho de haz no es preocupante.

Las corrientes obtenidas como resultado pueden verse en las tablas (5.18), (5.19) y (5.20)

elemento:	0	1	2	3	4	5	6	
$abs(c_n)$	1	2,5	2,61	0,8	0,82	0,5	0,21	0,29
$angle(c_n)$	0	1,32	-3,26	-21,43	-154,67	-175,27	4,11	47,96
$angle(c_{n+1}) - angle(c_n)$	1,32	-4,58	-18,17	-133,24	-20,6	179,38	43,85	NaN

Cuadro 5.18: Corrientes de la solución 1

elemento:	0	1	2	3	4	5	6	
$abs(c_n)$	1	2,88	3,2	1,1	0,97	0,68	0,31	0,34
$angle(c_n)$	0	3,3	0	-17,27	-149,23	-176,45	8,4	60,62
$angle(c_{n+1}) - angle(c_n)$	3,3	-3,3	-17,27	-131,96	-27,22	184,85	52,22	NaN

Cuadro 5.19: Corrientes de la solución 2

elemento:	0	1	2	3	4	5	6	
$abs(c_n)$	1	2,63	2,84	0,95	0,85	0,58	0,22	0,27
$angle(c_n)$	0	-1,7	-7,37	-25,87	-161,6	-178,92	0,87	17,99
$angle(c_{n+1}) - angle(c_n)$	-1,7	-5,67	-18,5	-135,73	-17,32	179,79	17,12	NaN

Cuadro 5.20: Corrientes de la solución 3

El lector se habrá dado cuenta de que aun exigiendo un desfase entre elementos contiguos menor de 100 grados, solo hemos conseguido desfases máximos de alrededor de los 180 y relaciones I_{MAX}/I_{min} de entorno a 10, siguen siendo corrientes algo complejas, pero se ha mejorado mucho con respecto a los resultados iniciales.

Se ha optado por implementar la solución 3 ya que es la más directiva, puesto que teniendo el mismo ancho de haz presenta un mayor NLPS, la implementación se ha realizado con una función de MATLAB ajena. Para el array anterior de 8 elementos, a una frecuencia $f = 5,3GHz$, con un espaciado de $0,67178 \pi$ tal y como se indicó anteriormente y con ranuras resonantes como elementos radiantes.

Como es evidente, el diagrama de radiación obtenido en la implementación dista de la realidad, aunque aun así se cumplen con las especificaciones en su totalidad, el resultado puede verse en la figura (5.29) donde vemos el diagrama de radiación del conjunto de la agrupación, es decir, el producto del diagrama de la ranura y el factor de array.

Las corrientes equivalentes normalizadas de las ranuras pueden observarse en las figuras (5.30) y (5.31), no se ha podido implementar ni el módulo ni la fase exacta. Se puede observar que la mayor diferencia entre el valor teórico y el implementado se encuentra, en el caso de la fase, en los puntos donde se realizan saltos de fase grandes, de ahí la importancia de poder limitar éstos saltos entre elementos, al igual que el mayor error en el módulo se produce al intentar sintetizar amplitudes muy pequeñas, esto se podría solucionar realizando más estudios de síntesis de arrays para obtener una solución teórica con unas amplitudes mínimas mayores, aunque como en nuestro caso el diagrama sigue cumpliendo los límites. El resultado implementado sobre la guía puede verse en la figura (5.32).

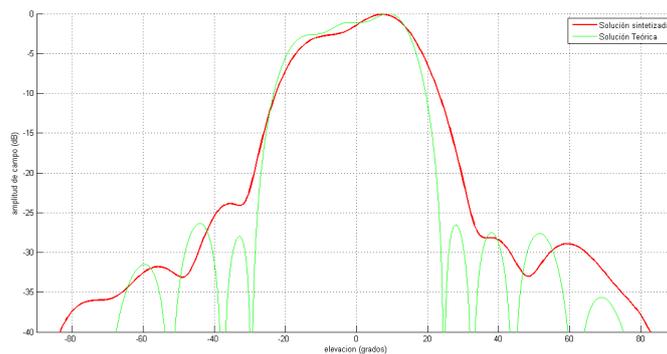


Figura 5.29: Diagrama de radiación del conjunto, teórico e implementación

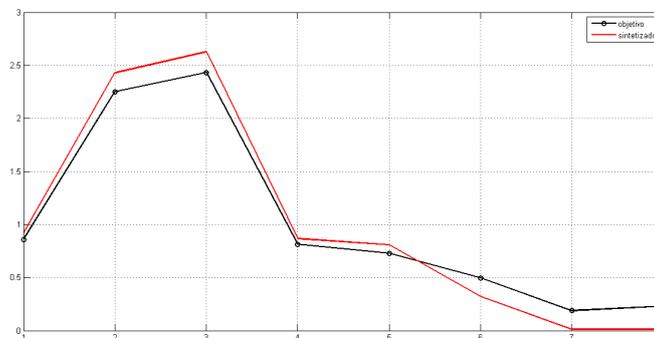


Figura 5.30: Módulos de corrientes teóricas y aproximación real implementada

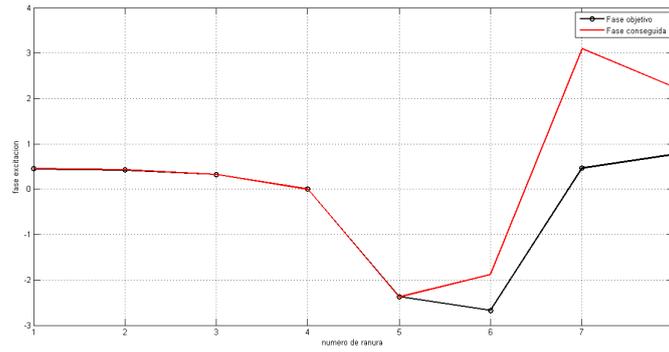


Figura 5.31: Fases de corrientes teóricas y aproximación real implementada

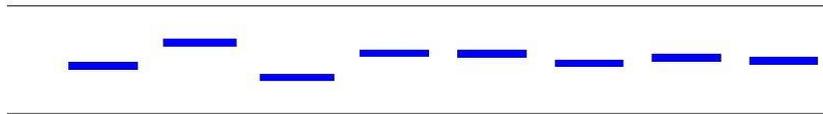


Figura 5.32: Esquemático descriptivo del posicionamiento de las ranuras en la guía resonante

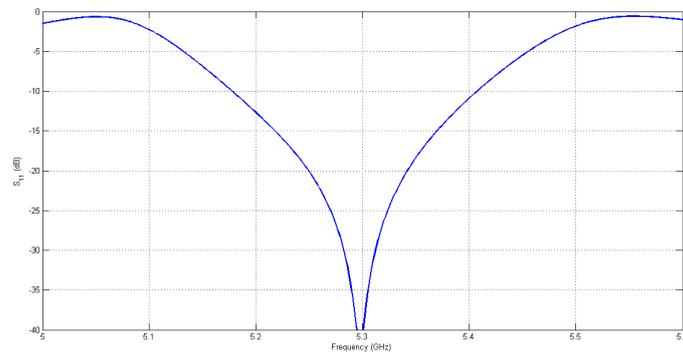


Figura 5.33: Parámetro S_{11} del conjunto del array

Capítulo 6

Conclusión

En este proyecto final de grado se ha estudiado y desarrollado en MATLAB el método *alternating projection method* para la síntesis de arrays con el principal objetivo de mejorar diagramas de radiación sintetizados por otros métodos, ya sea para la mejora de la forma de su factor de array o la simplificación de sus corrientes intentando mantener, en la medida de lo posible, las especificaciones en su forma, también se han realizado la síntesis de 3 tipos de diagramas ampliamente usados en aplicaciones de radiodeterminación y radiodetección, tras estas síntesis se han llegado a las siguientes conclusiones.

Si se parte de una síntesis cuyo patrón está muy alejado a lo deseable es más eficiente optimizar el diagrama sin imponer límites a las corrientes para después partir de esta solución, ahora sí, imponiéndoles límites.

Si se desea sintetizar un array atendiendo solo al diagrama, sin importar las corrientes, conviene imponer límites más estrictos, alargarlos para que se cumplan en los bordes, ya que estos suelen ser problemáticos.

Al sintetizar cierta pendiente, es mejor imponer como límite superior una recta con esa pendiente y otra por debajo como límite inferior para que se cumpla la especificación de rizado que usar el método que implemente la pendiente como límite de entrada. Si usamos éste último método corremos el riesgo de perder total y absolutamente la *planitud*, ya que no se ha implementado ningún límite de rizado, las pruebas realizadas fueron tan desastrosas que simplemente se optó por no usarse, aunque si se parte de una solución cercana el método converge, sus soluciones no son mejores que añadiendo rectas como límites

Para limitar anchos de haz lo mejor es alargar los límites inferiores hasta los valores angulares que te definen el ancho o imponer unos pequeños límites inferiores en esos valores.

A la hora de limitar corrientes es importante comprender la complejidad de nuestro factor de array ya que conviene forzar el límite de desfase, porque en regla general, con arrays complejos el desfase máximo suele estar 20 grados por encima del máximo impuesto aunque si se fuerza mucho el resultado puede ser desastroso. Los límites de amplitud como norma general se cumplen, por lo que no existe la necesidad de imponer unos más estrictos.

Uno de los grandes problemas con los que se encuentra un desarrollador es hacer una aplicación o algoritmo intuitivo, por lo que mi primera recomendación es el desarrollo de una interfaz gráfica y el desarrollo de las funciones necesarias para realizar los cambios en la coordenada angular para ajustarse a los múltiples cambios de variables que se realizan en el algoritmo.

Capítulo 7

Anexo

7.0.4. Función principal:

```
function [f, F] = Trabajo(F, Mu, MI,CM, Cm, Aph,Nt,N)
% Alternating Projection Method

% Se ha omitido la acentuacion por conflictos con LaTeX

% F -> Factor de array de partida
% Mu -> Limite superior del factor de array
% MI -> Limite inferior del factor de array
% CM -> Maximo modulo de las corrientes
% Cm -> Minimo modulo de las corrientes
% Aph -> Maximo desfase entre corrientes contiguas
% Nt -> Numero de puntos del array F
% N -> Numero de elementos radientes que conformaran el array

f=zeros(1,Nt);
numinter=5000;

for j=1:numinter

    [ F,flagFA ] = Pm( F,Mu,MI );    % Limitacion del factor de array

    Aux=(fft( ifftshift(F)));    % Obtencion de corrientes.

    f(1:N)=Aux(1:N);    % Diezmado de los coeficientes al numero de elementos
    f=f/f(1);

    [ f ,flagf]= fc(f,CM,Cm,Aph,N);    % Limitacion de las corrientes

    F=ifftshift( ifft(f));    % Obtencion del nuevo diagrama
    F=F/max(abs(F));

    if(flagFA==0 && flagf==0 && j~=1)
```

```
disp('La solucion SI cumple en todo momento con las especificaciones ');
sprintf('Se ha llegado en: %d iteraciones ',j);
    break;
end

end

if(flagFA==1 || flagf==1)
disp('La solucion NO cumple en todo momento con las especificaciones ');
end

end
```

7.0.5. Límites del Factor de array:

```
function [ F,flagPm ] = Pm( F,Mu,MI )
%en esta funcion se ha implementado el "encasillado" del factor de array
%dentro de los limites

%Se ha omitido la acentuacion por conflictos con LaTeX

% F -> vector del factor de array
% Mu -> vector de limites superiores
% MI-> vector de limites inferiores

    flagu=0;
    flagl=0;

for i=1:length(F)

%    Limites superiores

        if (abs(F(i))>10^(Mu(i)/20) && Mu(i)~= -100)
            flagu=1;
            F(i)=10^(Mu(i)/20)*exp(1i*angle(F(i)));
        end

%    Limites inferiores

        if (abs(F(i))<10^(MI(i)/20) && MI(i)~= -100)
            flagl=1;
            F(i)=10^(MI(i)/20)*exp(1i*angle(F(i)));
        end

end

%Se ha cumplido con las especificaciones impuestas?
if(flagu==1 || flagl==1)
    flagPm=1; %NO
else
    flagPm=0; %SI
end

end
```

7.0.6. Límites de las corrientes:

```
function [ f, flagFc ] = fc( f,CM,Cm,Aph,N)
%en esta funcion se ha implementado el "encasillado" de las corrientes
%dentro de los limites

%Se ha omitido la acentuacion por conflictos con LaTeX

% f -> vector de corrientes
% CM -> Modulo de corriente maximo permitido
% Cm -> Modulo de corriente minimo permitido
% Aph -> Desfase maximo permitido entre corrientes contiguas

flag1=0;
flag2=0;
flag3=0;

for i=1:N

% Limites superiores de amplitud
if (abs(f(i))>CM)
    flag1=1;
    f(i)=CM.*exp(1i*angle(f(i)));
end

% Limites inferiores de amplitud
if (abs(f(i))<Cm)
    flag2=1;
    f(i)=Cm.*exp(1i*angle(f(i)));
end

end

% Limites de fase

for i=N:-1:2
    if abs(abs(angle(f(i))-angle(f(i-1)))>Aph)
        flag3=1;
        f(i)=abs(f(i)).*exp(1i*(angle(f(i-1)) ...
            +sign(angle(f(i))-angle(f(i-1)))*Aph));
    end
end

%Se ha cumplido con las especificaciones impuestas?
if(flag1==1 || flag3==1 || flag2==1)
    flagFc=1; %NO
```

```
else
    flagFc=0; %SI
end

end
```

7.0.7. Funciones auxiliares:

Dibujar gráficas del factor de array

```
function ploter( FA,u, r )
%Normaliza el factor de array y lo dibuja en logaritmico

% FA= Vector factor de array en lineal.
% u= Vector que define el eje x.
% r= Color a representar;

FA=FA/max(FA);
plot(u,20*log10(abs(FA)),r);
grid on;

end
```

Cambiar del espacio del array al espacio real:

```
function [ Fout,Np,phi ] = CambioEspacio( Fin,kd )
% Esta funcion permite pasar del espacio del factor de array al espacio
% real gracias a la variable kd

% Fin -> Diagrama de entrada
% kd= K*d
% Fout-> Diagrama de salida
% NT -> longitud del vector Fout
% phi -> vector de eje x en el espacio real [0,180]

Nt=length( Fin );
    if(kd>pi)
        Auxiliar=[Fin Fin Fin];
        Np=round(2*kd*Nt/(2*pi));

        Fout=Auxiliar((round(3*Nt/2)-round(Np/2)+1):(round(3*Nt/2)+round(Np/2)));
        up=linspace(-kd,kd,length(Fout));
        phi=acos(up/kd);

    else
        Np=round(2*kd*Nt/(2*pi));
        Fout=Fin((round(Nt/2)-round(Np/2)+1):(round(Nt/2)+round(Np/2)));
        up=linspace(-kd,kd,length(Fout));
        phi=acos(up/kd);
    end

end

end
```

Imponer límites de forma gráfica:

```
function [ M ] = RatonAuto(Nt,kd)
% Esta funcion permite crear los limites superiores o inferiores del
% factor de array a traves del raton. SOLO LIMITES HORIZONTALES.

% IMPORTANTE:
% Antes de ejecutar esta funcion se ha de haber representado el espacio
% sobre el que se definiran los limites, el eje x ha de ser
% necesariamente el correspondiente al eje angular real, definido entre 0
% y 180 grados sexagesimales.

% Nt-> numero de puntos que necesita tener el vector de limites
% kd-> Resultado del producto del numero de onda por el espaciado entre
% elementos

NUM = 2*input('Introduzca el numero de limentes que impondra: ');
[X,Y] = ginput(NUM);

X=kd*cos((X)*pi/180);

X=floor(((X+pi)/(2*pi))*Nt);

M=ones(1,Nt)*-100;

    for i=1:2:(length(X)-1)
        if (X(i+1)==0)
            X(i+1)=1;
        end

        if (i>length(X))
            break;
        end

        if (X(i)>1 && X(i)<(Nt+1))
            M(X(i+1):X(i))=Y(i);
        end
    end

end
```

Bibliografía

- [1] H. Orchard, R. Elliott, and G. Stern, “Optimising the synthesis of shaped beam antenna patterns,” *Microwaves, Antennas and Propagation, IEEE Proceedings H*, vol. 132, pp. 63–68, February 1985.
- [2] R. J. Mailloux, *Phased Array Antenna Handbook, Second Edition (Artech House Antennas and Propagation Library)*. Artech Print on Demand, 2005.
- [3] B.-K. Yeo and Y. Lu, “Array failure correction with a genetic algorithm,” *Antennas and Propagation, IEEE Transactions on*, vol. 47, pp. 823–828, May 1999.
- [4] J. Corcoles, M. Gonzalez, and J. Rubio, “Fourier synthesis of linear arrays based on the generalized scattering matrix and spherical modes,” *Antennas and Propagation, IEEE Transactions on*, vol. 57, pp. 1944–1951, July 2009.
- [5] S. Singh, G. Mahanti, and H. Pilli, “Synthesis of discrete amplitude-only linear array antenna with minimum side lobe level using iterative fast fourier transform,” in *Computers and Devices for Communication (CODEC), 2012 5th International Conference on*, pp. 1–3, Dec 2012.
- [6] M. Shimizu, “Determining the excitation coefficients of an array using genetic algorithms,” in *Antennas and Propagation Society International Symposium, 1994. AP-S. Digest*, vol. 1, pp. 530–533 vol.1, June 1994.
- [7] W. P. M. N. Keizer, “Low-sidelobe pattern synthesis using iterative fourier techniques coded in matlab [em programmer’s notebook],” *Antennas and Propagation Magazine, IEEE*, vol. 51, pp. 137–150, April 2009.
- [8] X.-K. Wang, Y.-C. Jiao, and Y.-Y. Tan, “Synthesis of large thinned planar arrays using a modified iterative fourier technique,” *Antennas and Propagation, IEEE Transactions on*, vol. 62, pp. 1564–1571, April 2014.
- [9] B. C. Mahato, G. K. Mahanti, and A. Panda, “Synthesis of linear array antenna with uniform side lobe level and fixed dynamic range ratio using iterative fast fourier transform,” in *Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on*, pp. 204–206, Sept 2011.
- [10] V. Autores, “Ideas generales sobre la tecnología wind scatterometer.” Accedido en junio 2014 a url: <http://climate.nasa.gov/>.
- [11] V. Autores, “Página online de la asignatura de antenas de la universidad politécnica de valencia.” Accedido en junio 2014 a url: <http://www.upv.es/antenas/>.
- [12] E. Tohidi, M. Sebt, and M. Nayebi, “Linear array thinning using iterative fft plus soft decision,” in *Radar Symposium (IRS), 2013 14th International*, vol. 1, pp. 301–306, June 2013.