

Universidad Politécnica de Cartagena

TRABAJO FIN DE MÁSTER

INGENIERÍA EN SISTEMAS DE TELECOMUNICACIÓN

Análisis de sistemas de radio sobre fibra basados en arrays de antenas moduladas en el tiempo (TMA)

Autor

Luis Fernández Álvarez

Directores

Dr.Ing. Jacopo Nanni

Prof. Giovanni Tartarini

Prof. David Cañete Rebenaque



Agradecimientos

Ha sido un largo camino y sobre todo un difícil año 2020 hasta llegar aquí. Después de muchas horas de estudio y de muchas situaciones excepcionales, por fin llegó el día de redactar el que será mi último trabajo antes de completar este prestigioso máster.

Quiero agradecer especialmente la ayuda de mis compañeros, ya que sin ellos hubiera sido mucho más difícil llegar hasta aquí. En especial se me vienen a la mente los nombres de Antonio Manuel Huescar, Joaquín García o Alejandro Gil.

Por otro lado, también quiero agradecer a mi familia y a los profesores de la Universidad de Bolonia, los doctores Jacopo Nanni y Giovanni Tartarini, por el gran apoyo y ayuda proporcionada.

Por último, agradecer tanto a mi coordinador de Erasmus-Prácticas como a mi director de TFM, los profesores Rafael Toledo Moreo y David Cañete Rebenaque respectivamente, por haberse ofrecido a ayudarme desinteresadamente y facilitarme en gran medida el trabajo realizado.

Dicho todo esto solo quiero volver a decir: gracias.

RESUMEN

El trabajo se centra en el área de procesado de la señal aplicado a sistemas de radio sobre fibra (RoF). A partir de un software desarrollado en MATLAB se realiza la simulación del diagrama de radiación generado por un sistema de RoF que transmite mediante un array de antenas moduladas en el tiempo (TMA). A partir de dicho software se trata de encontrar los parámetros necesarios para poder generar las señales que alimentan el TMA.

Una vez obtenidos todos los parámetros necesarios, se procederá al montaje físico real en un laboratorio. En él, se realizarán todas las medidas necesarias para comprobar que los resultados obtenidos en las simulaciones se corresponden con los resultados reales obtenidos.

El presente trabajo abarcará en todo momento dos casos de estudio. Un primer caso se basará en un array de $N_a = 2$ antenas y buscará objetivos de localización. El segundo, y en el que se ha llevado a cabo en mayor profundidad, consta de un array de $N_a = 4$ antenas y el objetivo buscado será el apuntamiento simultáneo en 5 direcciones del diagrama de radiación del sistema.

El trabajo se basa tanto en una parte teórica relacionada con la generación y procesado de la señal, como en otra experimental relacionada con la estimación de la calidad de la transmisión en función de los parámetros obtenidos en las simulaciones.

Palabras Clave: RoF, MATLAB, TMA.

ÍNDICE

1. Introducción	9 - 12
1.1. Sistemas de radio sobre fibra	9 - 10
1.2. Arrays modulados en el tiempo	10 - 12
1.3. Objetivos	13 - 14
2. Desarrollo teórico previo	15 - 26
2.1. Sistema de RoF sobre TMA con $N_a = 2$ elementos	13 - 17
2.2. Sistema de RoF sobre TMA con $N_a = 4$ elementos	17 - 26
3. Desarrollo experimental mediante MATLAB	27 - 34
3.1. El código	27 - 28
3.2. Sistema de RoF sobre TMA con $N_a = 2$ elementos en MATLAB	26 - 33
3.2.1. Especificaciones sobre el código implementado	28 - 29
3.2.2. Resultados	29 - 32
3.3. Sistema de RoF sobre TMA con $N_a = 4$ elementos en MATLAB	26 - 33
3.3.1. Especificaciones sobre el código implementado	28 - 29
3.3.2. Resultados	29 - 32
4. Desarrollo experimental en el laboratorio	35 - 59
4.1. Sistema de RoF sobre TMA con $N_a = 2$ elementos en el laboratorio	35 - 50
4.2. Sistema de RoF sobre TMA con $N_a = 4$ elementos en el laboratorio	50 - 52
5. Conclusiones y líneas futuras	59
6. Bibliografía	61 - 62

LISTADO DE FIGURAS

- Figura 1.** Esquema general de un sistema de RoF.
- Figura 2.** A la izquierda diagrama de bloques general de un TMA con N_a elementos.
A la derecha el tren de pulsos que sirve de alimentación del TMA.
- Figura 3.** Sistema de RoF a considerar con N_a elementos.
- Figura 4.** Trenes de pulsos de polarización.
- Figura 5.** Diagrama de radiación obtenido para un sistema con $N_a = 2$ elementos y parámetros $c_1 = 0.5$, $c_2 = 0.5$, $c_{i,1} = 0$ y $c_{i,2} = 0.5$.
- Figura 6.** Paneles de selección.
- Figura 7.** Selección por teclado del desfase de cada uno de los elementos del TMA.
- Figura 8.** Simulación para la recreación del escenario del problema original.
- Figura 9.** Compensación del desfase disminuyendo el umbral (threshold = 10^{-2}).
- Figura 10.** Compensación del desfase modificando la condición (OR condition).
- Figura 11.** Solución óptima de la compensación del desfase modificando la condición (OR condition).
- Figura 12.** Diagrama de radiación óptimo para el apuntamiento simultáneo en 5 direcciones.
- Figura 13.** Lugar de trabajo en el laboratorio (1).
- Figura 14.** Lugar de trabajo en el laboratorio (2).
- Figura 15.** Lugar de trabajo en el laboratorio (3).
- Figura 16.** Calibración del sistema en el laboratorio.
- Figura 17.** Resultados de la secuencia óptima puesta a prueba en el laboratorio ($N_a = 2$).
- Figura 18.** Simulación de la inversión de modos.
- Figura 19.** Resultados de la secuencia óptima puesta a prueba en el laboratorio ($N_a = 4$).
- Figura 20.** Beam-steering (1).
- Figura 21.** Beam-steering (2).
- Figura 22.** Beam-steering (3).
- Figura 23.** Beam-steering (4).
- Figura 24.** Secuencia óptima sin incluir desfase adicional.
- Figura 25.** Secuencia óptima original con desfase añadido.
- Figura 26.** Nueva secuencia óptima para la compensación del desfase.
- Figura 27.** Reconstrucción del modo $f_0 + f_M$.

1. INTRODUCCIÓN

1.1. SISTEMAS DE RADIO SOBRE FIBRA

Los sistemas de radio sobre fibra (RoF) son considerados una solución viable para redes de datos de corto alcance utilizadas tanto para aplicaciones profesionales como orientadas al consumidor.

En estos sistemas las portadoras de radiofrecuencia (RF) se transportan como subportadoras del canal óptico sobre un enlace en conexiones punto-punto o punto-multipunto. Estas señales modulan un láser y la señal modulada es transmitida a través de la fibra óptica. Finalmente, se recupera la señal eléctrica a través de un fotodiodo.

Los sistemas de RoF se consideran como la piedra angular, bajo la cual se sustentarán las redes de nueva generación, al permitir que una única plataforma física soporte la transmisión de información proveniente de usuarios fijos y móviles. Con estos sistemas es posible disminuir la atenuación sufrida por el medio guiado y aumentar de manera considerable el ancho de banda, velocidad de transmisión y número máximo de usuarios a considerar.

Son muy utilizados en sistemas de televisión (CATV), redes de área local (LAN), aplicaciones militares etc.

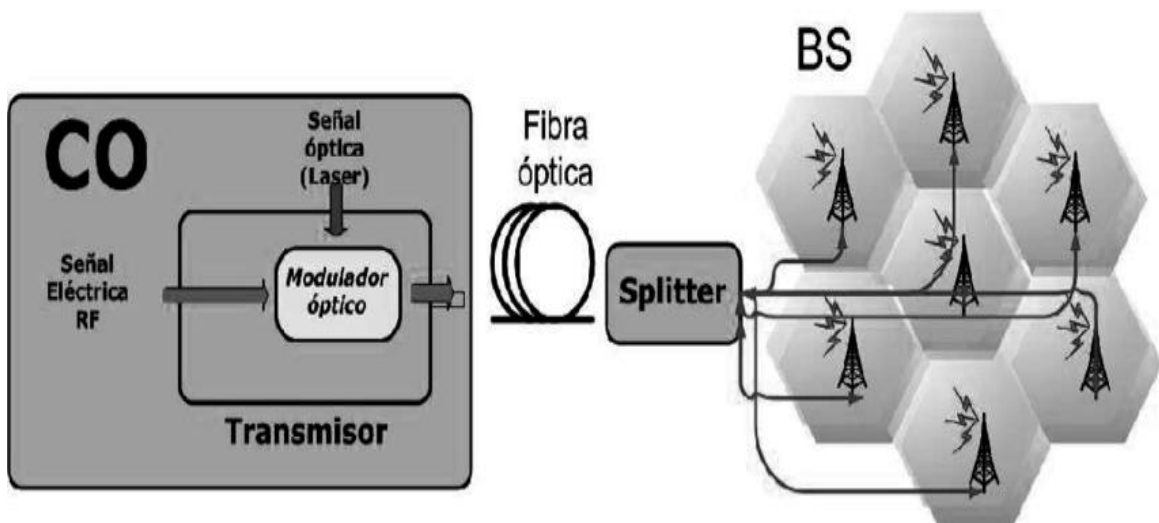


Figura 1. Esquema general de un sistema de RoF.

1.2. ARRAYS MODULADOS EN EL TIEMPO

Los arrays modulados en el tiempo o arrays de antenas modulados en el tiempo (TMA) son sistemas electromagnéticos cuyo patrón de potencia radiada se controla mediante la aplicación de pulsos periódicos de ancho variable a los elementos individuales del propio array. La naturaleza no lineal de la operación de los TMA provoca la aparición de patrones de radiación en las frecuencias armónicas de tales pulsos periódicos.

En otras palabras, los TMA en su configuración más simple, son arrays de antenas cuyos patrones de potencia radiada se controlan habilitando y deshabilitando periódicamente las excitaciones de los elementos individuales del array, como se ilustra en la Figura 2.

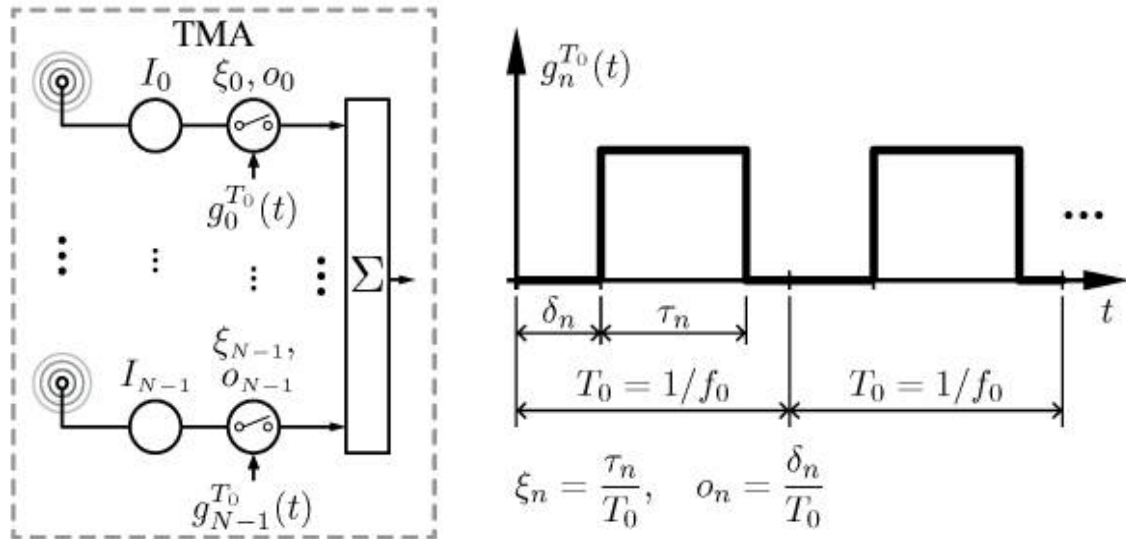


Figura 2. A la izquierda diagrama de bloques general de un TMA con N_a elementos. A la derecha el tren de pulsos que sirve de alimentación del TMA.

La modulación periódica de un TMA es una operación no lineal que genera señales de banda lateral radiadas a frecuencias desplazadas desde la frecuencia portadora en múltiplos de la tasa de modulación. Los principales parámetros de tiempo que gobiernan los TMA son los retardos (δ_n en la Figura 2, aunque más adelante será renombrado como $c_{i,n}$) y ciclos de trabajo (τ_n en la Figura 2, que también será renombrado, en este caso, como c_n) de los trenes de pulsos que alimentan los TMA. Adaptando los patrones armónicos al canal inalámbrico y combinando sus salidas correspondientes, los TMA pueden realizar la formación de haces adaptativa (mediante las interferencias constructivas y destructivas de los patrones de radiación de los diferentes elementos que componen el array) [1].

La capacidad de los TMA se ha aprovechado en muchas aplicaciones, como radiogoniometría, transferencia de energía inalámbrica o comunicación segura. En todos estos casos, la fuente de RF siempre está prevista o realizada muy cerca del propio array. Sin embargo, en el presente trabajo, será deseable explotar la capacidad de reconfiguración de TMA para optimizar la cobertura inalámbrica a través de una distribución de los arrays lejos de la fuente de RF, realizando de esta manera una formación de haz eficiente.

Además, los TMA presentan muchas ventajas como la simplicidad de su hardware (lo que influye directamente en su tamaño y coste) o un reducido consumo de potencia entre otras.

1.3. OBJETIVOS

En el presente trabajo, la viabilidad de un sistema de RoF basado en TMA será estudiada y se planteará como una solución prometedora para métodos de localización (basada en la combinación de tanto la suma como de la diferencia de los patrones de radiación para hacerlos muy sensibles a las variaciones del ángulo de apuntamiento) y *'beamforming'* (donde se buscará crear patrones 'multi-harmónicos' para apuntar simultáneamente en varias direcciones).

Para ello se llevará a cabo el desarrollo de un programa para la simulación de los patrones de radiación de los sistemas mediante MATLAB y se estudiarán los escenarios reales físicos en un laboratorio mediante el uso de arrays de $N_a = 2$ elementos para los objetivos de localización y de $N_a = 4$ elementos para la creación de los patrones multi-harmónicos.

2. DESARROLLO TEÓRICO PREVIO

2.1. SISTEMA DE RoF SOBRE TMA CON $N_a = 2$ ELEMENTOS

El primer sistema que ha sido considerado es el que aparece en la Figura 3. Está formado por un primer segmento de RoF que alimenta el TMA con la señal que va a ser transmitida y por el propio array de $N_a = 2$ antenas que, junto con los trenes de pulsos que polarizan los fotodiodos y activan o no los diferentes elementos (véase Figura 4), conforman el TMA. Las antenas son de tipo parche en todo momento.

En el primer segmento la señal de RF, con frecuencia $f_0 = 2.45$ GHz, es modulada por un láser DFB en tercera ventana de transmisión (1550 nm) y después dividida por un divisor óptico en N_a ramas que desembocan en un fotodiodo PIN.

Por otro lado, la tensión de polarización de los fotodiodos viene dada por trenes de pulsos de frecuencia $f_M = 10$ KHz ($f_M = \frac{1}{T_M}$) y ciclo de trabajo (c_n o c_k) y retardo inicial ($c_{i,n}$ o $c_{i,k}$) variables. Como se verá más adelante, todos estos valores estarán normalizados con respecto al periodo por lo que sus valores estarán comprendidos en el intervalo $[0,1]$. Mediante la variación de estos parámetros se podrá conseguir modificar el diagrama de radiación del TMA.

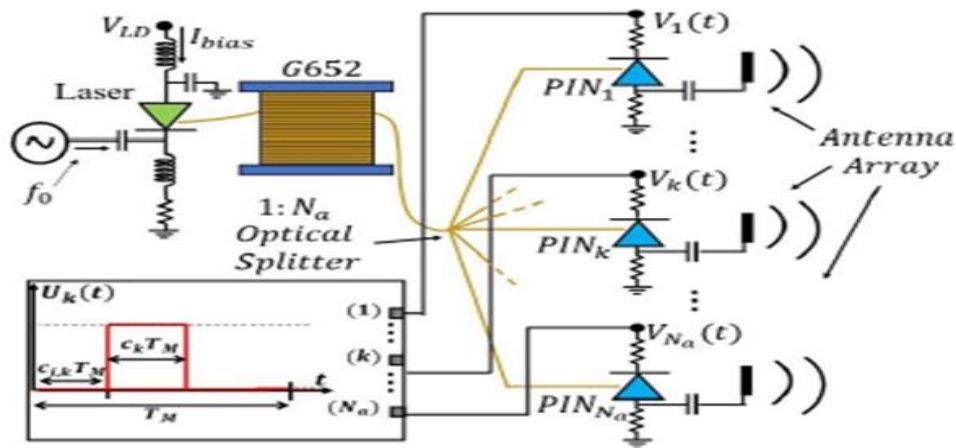


Figura 3. Sistema de RoF a considerar con N_a elementos.

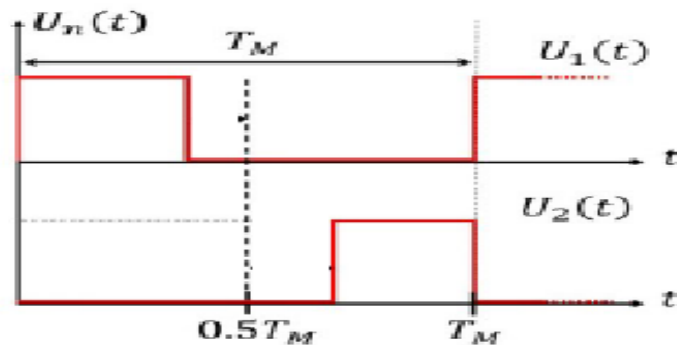


Figura 4. Trenes de pulsos de polarización.

Si situamos un array de $N_a = 2$ antenas, longitud de la fibra (del tipo G652, también conocida como fibra monomodo SMF estándar) $L_{\text{fibra}} = 1$ km, separación entre las antenas $L = \frac{\lambda}{2} = 6.12$ cm, potencia del láser $P_{\text{in,RF}} = 3$ dB y parámetros $c_1 = 0.5$, $c_2 = 0.5$, $c_{i,1} = 0$ y $c_{i,2} = 0.5$ se obtiene el diagrama de radiación que aparece en la Figura 5.

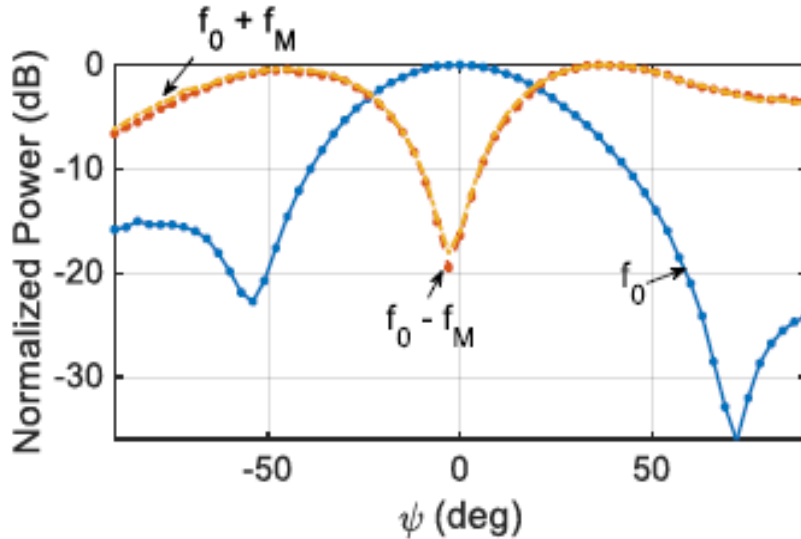


Figura 5. Diagrama de radiación obtenido para un sistema con $N_a = 2$ elementos y parámetros $c_1 = 0.5$, $c_2 = 0.5$, $c_{i,1} = 0$ y $c_{i,2} = 0.5$.

Si observamos el diagrama de radiación obtenido experimentalmente en la Figura 5 (medido mediante una antena de tipo bocina situada en campo lejano conectada a un analizador de espectros), podemos apreciar claramente dos componentes. Por un lado está la componente principal (en azul) a la frecuencia de trabajo f_0 la cual alcanza su máximo en la dirección 'broadside'. La otra componente que aparece sería la producida por los armónicos resultantes al modular el TMA con trenes de pulsos y se encuentra a frecuencias $f_0 \pm h \cdot f_M$ (donde h hace referencia al número de armónico). Si consideramos solo el primer armónico obtenemos la curva de color rojo donde ambos modos secundarios coinciden.

El resultado esperable es obtener un máximo a $\Psi = 0^\circ$ para la componente principal y simultáneamente un mínimo para las componentes $f_0 \pm f_M$ [2]. Sin embargo, esto no ocurre ya que existe una ligera diferencia de longitud entre los caminos ópticos recorridos por la señal, lo cual hace que alimenten al TMA con un desfase de alrededor de 10° . La diferencia de longitud entre los caminos ópticos se debe a que las fibras comerciales empleadas no son exactamente de la misma longitud y una pequeña diferencia en este parámetro puede producir un gran cambio en la fase. Esto se traduce en un desplazamiento del mínimo de las componentes $f_0 \pm f_M$ de aproximadamente 3° .

Uno de los objetivos que se perseguirá, será desplazar este mínimo a 0° variando los parámetros anteriormente comentados c_n y $c_{i,n}$ para aplicaciones de localización. Si conseguimos obtener el máximo de radiación a la frecuencia fundamental y el mínimo a

las frecuencias secundarias simultáneamente, el sistema podrá comportarse como un sistema de rastreo monopulso [3].

2.2. SISTEMA DE RoF SOBRE TMA CON $N_a = 4$ ELEMENTOS

En el caso de $N_a = 4$ elementos, el escenario es análogo al mostrado anteriormente. Sin embargo, al tener un mayor número de antenas, se dispone de un mucho mayor rango de libertad y por ello los objetivos perseguidos son diferentes. Sin embargo, en este caso no se dispone de un estudio previo que usar como base por lo que los resultados no podrán contrastarse ni calibrarse de la misma forma que con $N_a = 2$ antenas.

En el caso de $N_a = 2$ el objetivo principal estaba relacionado con la localización. Sin embargo, el objetivo ahora será crear patrones '*multi-harmónicos*' para apuntar a diferentes direcciones simultáneamente. Las ventajas de usar múltiples haces con fines de comunicación se encuentran en la posibilidad de gestionar más comunicaciones y servicios desde diferentes direcciones espaciales y para operar sobre múltiples bandas de frecuencia. Además, los patrones con diferentes formas pueden ser utilizados en sistemas adaptativos para mantener enlaces inalámbricos confiables en presencia de bloqueadores o señales de interferencia o para atender diferentes solicitudes de servicio [4].

3. DESARROLLO EXPERIMENTAL MEDIANTE MATLAB

3.1. EL CÓDIGO

A continuación se muestra el código empleado para las simulaciones tanto para $N_a = 2$ como $N_a = 4$ elementos.

```
clear all
close all
clc
%%
%load('Luis_data_4_antennas_Larga_1.mat')
%%
Na_vector = [2 4];
posibili_valori1 = {'2','4'};
Indice_antena = listdlg('PromptString','Number of antennas:',...
    'SelectionMode','single',...
    'ListString',posibili_valori1);
Na = Na_vector(Indice_antena); %Na puede valer 2 o 4
A = zeros(1,Na); %Constante que multiplica la secuencia Uk

if(Na == 2)
    possibili_valori2 = {'Compensate phase shift of f0 + fM AND f0 -
    fM' ,...
        'Compensate phase shift of f0 + fM OR f0 - fM'};
    Indice_restriccion = listdlg('PromptString','Choose the problem to
    solve:',...
        'SelectionMode','single',...
        'ListString',posibili_valori2);
end

for i = 1:Na
    str = ['For the antenna: ' , int2str(i)];
    disp(str)
    deltha(i) = input('Type the phase shift you want to compensate of
    the antenna in degrees: ');
    deltha(i) = deltha(i)*(pi/180); %En radianes
    %deltha = [0 -27.4 -58.6 7.4]*(pi/180);
    %deltha = [0 0 0 0]*(pi/180);
    %deltha = [0 -27.4 -58.6 -68.4]*(pi/180);
    %deltha = [0 48.4 -58.6 -68.4]*(pi/180);

    if(Na==2)
        A(1) = 0.7;
        A(2) = sqrt(1 - A(1)^2);
    end
    if(Na==4)
        A(i) = 0.5; %Todas iguales
    end
end

f0 = 2.45e9; %Frecuencia de RF
fM = 10e3; %Frecuencia de los pulsos del TMA (modulating frequency)
c = 3e8; % m/s
lambda = c/f0; %Longitud de onda
L =lambda/2; %Distancia entre las antenas
btheta = 2*pi/lambda; %Constante de propagacion
if (Na==2)
```

```

    phi = linspace((-90*2*pi)/360, (90*2*pi)/360, 1000); %Phi es la
variable
    c_sw = linspace(0, 1, 101);
    haux = 3; %Numero de coeficientes de Fourier a considerar
else
    phi = linspace((-90*2*pi)/360, (90*2*pi)/360, 100); %Phi es la
variable
    c_sw = linspace(0, 0.25, 6);
    haux = 5; %Numero de coeficientes de Fourier a considerar
end
u_hk = zeros(haux, Na);
d = 0;
num = 1;
a = 1;

if(Na == 2)
    for cont = 1:length(c_sw)
        for cont2 = 1:length(c_sw)
            for caso = 1:2

                if (caso == 1)
                    if(1-c_sw(cont2) <= 1-c_sw(cont))
                        %Cuando un pulso acaba debe empezar el de la
otra antena (sincronismo)
                        cin = [0 c_sw(cont)]; %Delay normalizado de
las antenas. Valor a elegir.
                        %cin = [0 0.5];
                        cn = [c_sw(cont) 1-c_sw(cont2)]; %DC de las
antenas (independiente del delay). Valor a elegir.
                        %cn = [0.5 0.5];

                    end
                else
                    if(c_sw(cont2) <= c_sw(cont))
                        cin = [0 c_sw(cont2)];
                        cn = [c_sw(cont) c_sw(cont)-c_sw(cont2)];
                    end
                end
            end

            for i=1:haux
                h = (i-ceil(haux/2));
                for j=1:Na
                    if ((cin(j)>=0) && (cin(j)<= 1-cn(j)))
                        u_hk(i, j) = cn(j)*sinc(cn(j)*h)*exp(-
1i*h*pi*(cn(j)+2*cin(j)));
                    else
                        u_hk(i, j) = (sin(h*pi*(1-cin(j)))*exp(-
1i*h*pi*(1+cin(j)))...
+ sin(h*pi*(cin(j)+cn(j)-1))*exp(-
1i*h*pi*(cin(j)+cn(j)-1)))/h*pi;
                    end
                end
            end

            %El factor de la antena tiene infinitos terminos.
Nuestra idea es calcular
            %Solo los terminos referidos a -1,0 y 1 (f0 y f0+-fM)

            F = zeros(length(phi), length(u_hk(:, 1)), Na);
            F_tot = zeros(length(phi), length(u_hk(:, 1)));

```

```

        for i=1:length(phi)
            for j=1:length(u_hk(:,1))
                for k=1:Na
                    F(i,j,k) =
A(k)*u_hk(j,k)*exp(1i*bheta*k*L*sin(phi(i))+1i*deltha(k)); %OJO es sin
y no cos (Jacopo)
                    end
                end
            end

            F_tot = sum(F,3);
            threshold = 1e-3;
            if (Indice_restriccion == 1)

if(abs(F_tot(length(phi)/2,1)).^2/max(abs(F_tot(:,1)).^2) <= threshold
&&...

abs(F_tot(length(phi)/2,3)).^2/max(abs(F_tot(:,3)).^2) <= threshold)
            if (num == 1)
                F_tot_sol(:, :, num) = F_tot;
                cn_sol(:, :, num) = cn;
                cin_sol(:, :, num) = cin;
                num = num+1; %Para poder guardar la
próxima solución
            else
                if(cn(1,1)~=cn_sol(1,1,num-1) ||
cn(1,2)~=cn_sol(1,2,num-1) || ...
                cin(1,1)~=cin_sol(1,1,num-1) ||
cin(1,2)~=cin_sol(1,2,num-1))
                    F_tot_sol(:, :, num) = F_tot;
                    cn_sol(:, :, num) = cn;
                    cin_sol(:, :, num) = cin;
                    num = num+1; %Para poder guardar la
próxima solución
                end
            end
        else
            if(abs(F_tot(length(phi)/2,1)).^2/max(abs(F_tot(:,1)).^2) <= threshold
||...

abs(F_tot(length(phi)/2,3)).^2/max(abs(F_tot(:,1)).^2) <= threshold)

                if (num == 1)
                    F_tot_sol(:, :, num) = F_tot;
                    cn_sol(:, :, num) = cn;
                    cin_sol(:, :, num) = cin;
                    num = num+1; %Para poder guardar la
próxima solución
                else
                    if(cn(1,1)~=cn_sol(1,1,num-1) ||
cn(1,2)~=cn_sol(1,2,num-1) || ...
                    cin(1,1)~=cin_sol(1,1,num-1) ||
cin(1,2)~=cin_sol(1,2,num-1))
                        F_tot_sol(:, :, num) = F_tot;
                        cn_sol(:, :, num) = cn;
                        cin_sol(:, :, num) = cin;
                        num = num+1; %Para poder guardar la
próxima solución
                    end
                end
            end
        end
    end
end

```

```

end
end
end
end
end
end
num = num-1;
end

if (Na==4)
    for cont = 1:length(c_sw)
        for cont2 = 1:length(c_sw)
            for cont3 = 1:length(c_sw)
                for cont4 = 1:length(c_sw)
                    for cont5 = 1:length(c_sw)
                        for cont6 = 1:length(c_sw)
                            for cont7 = 1:length(c_sw)
                                for cont8 = 1:length(c_sw)

                                    cin = [c_sw(cont)
(0.1+c_sw(cont3)) (0.4+c_sw(cont5)) (0.7+c_sw(cont7))]; %Delay
normalizado de las antenas.
                                    cn = [(0.05 + c_sw(cont2)) (0.02
+ c_sw(cont4)) (0.02 + c_sw(cont6)) (0.05 + c_sw(cont8))]; %DC de
las antenas (independiente del delay).
                                    %cin = [0 0.15 0.3 0.45];
%Delay normalizado de las antenas.
                                    %cn = [0.05 0.07 0.07 0.05];
%DC de las antenas (independiente del delay).

                                    %CASE 1
                                    %GOOD WITH NO FIBER ADDED. BAD
WITH FIBER ADDED
                                    %cin = [0 0.18 0.35 0.53];
                                    %cn = [0.34 0.3 0.29 0.2];

                                    %GOOD WITH FIBER ADDED
                                    %cin = [0 0.28 0.35 0.52];
                                    %cn = [0.66 0.69 0.28 0.21];

                                    %CASE 2
                                    %GOOD WITH NO FIBER ADDED. BAD
WITH FIBER ADDED
                                    %cin = [0 0.13 0.28 0.42];
                                    %cn = [0.2 0.27 0.3 0.33];

                                    %GOOD WITH FIBER ADDED
                                    %cin = [0 0.17 0.2 0.27];
                                    %cn = [0.2 0.27 0.3 0.33];

                                for i=1:haux
                                    h = (i-ceil(haux/2));
                                    for j=1:Na
                                        if ((cin(j))>=0) &&
(cin(j)<= 1-cn(j)))
                                            u_hk(i,j) =
cn(j)*sinc(cn(j)*h)*exp(-1i*h*pi*(cn(j)+2*cin(j)));
                                        else
                                            u_hk(i,j) =

```

```

(sin(h*pi*(1-cin(j)))*exp(-li*h*pi*(1+cin(j)))...
+
sin(h*pi*(cin(j)+cn(j)-1))*exp(-li*h*pi*(cin(j)+cn(j)-1)))/h*pi;
end
end
end

%El factor de la antena tiene
infinitos terminos. Nuestra idea es calcular
%Solo los terminos referidos a -
1,0 y 1 (f0 y f0+-fM)

F =
zeros(length(phi),length(u_hk(:,1)),Na);
F_tot =
zeros(length(phi),length(u_hk(:,1)));

for i=1:length(phi)
for j=1:length(u_hk(:,1))
for k=1:Na
F(i,j,k) =
A(k)*u_hk(j,k)*exp(li*bheta*k*L*sin(phi(i))+li*deltha(k)); %OJO es sin
y no cos (Jacopo)
end
end
end

F_tot = sum(F,3);
threshold_2 = -3; %dB
threshold_3 = 0; %dB
threshold_4 = -5; %dB

% delta = [0 -27.4 -58.6 -68.4]
diferencia_1(num) =
20*log10(abs(F_tot(39,2))/max(abs(F_tot(:,3)))); %Normalizado (a -20°)
diferencia_2(num) =
20*log10(abs(F_tot(67,4))/max(abs(F_tot(:,3)))); %Normalizado (a 30°)
diferencia_3(num) =
20*log10(abs(F_tot(39,3))/max(abs(F_tot(:,3)))); %Normalizado (a -20°)
diferencia_4(num) =
20*log10(abs(F_tot(67,3))/max(abs(F_tot(:,3)))); %Normalizado (a 30°)
diferencia_5(num) =
20*log10(abs(F_tot(39,4))/max(abs(F_tot(:,3)))); %Normalizado (a -20°)
diferencia_6(num) =
20*log10(abs(F_tot(67,2))/max(abs(F_tot(:,3)))); %Normalizado (a 30°)

diferencia_7(num) =
20*log10(abs(F_tot(28,1))/max(abs(F_tot(:,3)))); %Normalizado (a -40°)
diferencia_8(num) =
20*log10(abs(F_tot(78,5))/max(abs(F_tot(:,3)))); %Normalizado (a 50°)
diferencia_9(num) =
20*log10(abs(F_tot(28,3))/max(abs(F_tot(:,3)))); %Normalizado (a -40°)
diferencia_10(num) =
20*log10(abs(F_tot(78,3))/max(abs(F_tot(:,3)))); %Normalizado (a 50°)
diferencia_11(num) =
20*log10(abs(F_tot(28,5))/max(abs(F_tot(:,3)))); %Normalizado (a -40°)
diferencia_12(num) =
20*log10(abs(F_tot(78,1))/max(abs(F_tot(:,3)))); %Normalizado (a 50°)

% delta = [0 -27.4 -58.6 7.4]

```

```

%                                diferencia_1(num) =
20*log10(abs(F_tot(41,2))/max(abs(F_tot(:,3)))); %Normalizado (a -18°)
%                                diferencia_2(num) =
20*log10(abs(F_tot(60,4))/max(abs(F_tot(:,3)))); %Normalizado (a 15°)
%                                diferencia_3(num) =
20*log10(abs(F_tot(41,3))/max(abs(F_tot(:,3)))); %Normalizado (a -18°)
%                                diferencia_4(num) =
20*log10(abs(F_tot(60,3))/max(abs(F_tot(:,3)))); %Normalizado (a 15°)
%                                diferencia_5(num) =
20*log10(abs(F_tot(41,4))/max(abs(F_tot(:,3)))); %Normalizado (a -18°)
%                                diferencia_6(num) =
20*log10(abs(F_tot(60,2))/max(abs(F_tot(:,3)))); %Normalizado (a 15°)
%
%                                diferencia_7(num) =
20*log10(abs(F_tot(29,1))/max(abs(F_tot(:,3)))); %Normalizado (a -38°)
%                                diferencia_8(num) =
20*log10(abs(F_tot(66,5))/max(abs(F_tot(:,3)))); %Normalizado (a 28°)
%                                diferencia_9(num) =
20*log10(abs(F_tot(29,3))/max(abs(F_tot(:,3)))); %Normalizado (a -38°)
%                                diferencia_10(num) =
20*log10(abs(F_tot(66,3))/max(abs(F_tot(:,3)))); %Normalizado (a 28°)
%                                diferencia_11(num) =
20*log10(abs(F_tot(29,5))/max(abs(F_tot(:,3)))); %Normalizado (a -38°)
%                                diferencia_12(num) =
20*log10(abs(F_tot(66,1))/max(abs(F_tot(:,3)))); %Normalizado (a 28°)

                                if (num == 1)
                                    if((diferencia_3(num) <=
threshold_3 && diferencia_4(num) <= threshold_3 && ...
((diferencia_1(num) >=
threshold_2 && diferencia_2(num) >= threshold_2) || ...
(diferencia_5(num) >=
threshold_2 && diferencia_6(num) >= threshold_2))) && ...
diferencia_9(num) <=
threshold_3 && diferencia_10(num) <= threshold_3 && ...
((diferencia_7(num) >=
threshold_4 && diferencia_8(num) >= threshold_4) || ...
(diferencia_11(num) >=
threshold_4 && diferencia_12(num) >= threshold_4))

                                F_tot_sol(:, :, num) =

                                cn_sol(:, :, num) = cn;
                                cin_sol(:, :, num) = cin;
                                num = num + 1;
                                end
                                else
                                    if(cn(1,1)~=cn_sol(1,1,num-1)
|| cn(1,2)~=cn_sol(1,2,num-1) || ...
cn(1,3)~=cn_sol(1,3,num-1) || cn(1,4)~=cn_sol(1,4,num-1) || ...
cin(1,1)~=cin_sol(1,1,num-1) || cin(1,2)~=cin_sol(1,2,num-1) || ...
cin(1,3)~=cin_sol(1,3,num-1) || cin(1,4)~=cin_sol(1,4,num-1))

                                    if((diferencia_3(num) <=
threshold_3 && diferencia_4(num) <= threshold_3 && ...
((diferencia_1(num) >=

```



```

        xlabel('\psi (deg)')
        ylabel('Normalized power (dB)')
        legend('f_0 - f_M' , 'f_0' , 'f_0 + f_M')
        xlim([-90 90]);
end
end

if (Na == 4)
figure(1)
indice = 126; %121 179 265 285
%F_tot_sol = F_tot;
plot(((180/pi)*phi),20*log10(abs(F_tot_sol(:,1,indice))/max(abs(F_tot_sol(:,3,indice))))),...

((180/pi)*phi),20*log10(abs(F_tot_sol(:,2,indice))/max(abs(F_tot_sol(:,3,indice))))),...

((180/pi)*phi),20*log10(abs(F_tot_sol(:,3,indice))/max(abs(F_tot_sol(:,3,indice))))),...

((180/pi)*phi),20*log10(abs(F_tot_sol(:,4,indice))/max(abs(F_tot_sol(:,3,indice))))),...

((180/pi)*phi),20*log10(abs(F_tot_sol(:,5,indice))/max(abs(F_tot_sol(:,3,indice))))))

xlabel('\psi (deg)')
ylabel('Normalized power (dB)')
legend('f_0 - 2 * f_M' , 'f_0 - f_M' , 'f_0' , 'f_0 + f_M' , 'f_0 + 2 * f_M')
xlim([-90 90]);
ylim([-50 0]);
title('\delta = [0° -27.4° -58.6° -68.4°]')
%title('\delta = [0° 48.4° -58.6° -68.4°]')
%title('\delta = [0° 0° 0° 0°]')

%SIN NORMALIZAR
figure(2)
indice = 1; %121 179 265 285
%F_tot_sol = F_tot;
plot(((180/pi)*phi),20*log10(abs(F_tot_sol(:,1,indice))),...
((180/pi)*phi),20*log10(abs(F_tot_sol(:,2,indice))),...
((180/pi)*phi),20*log10(abs(F_tot_sol(:,3,indice))),...
((180/pi)*phi),20*log10(abs(F_tot_sol(:,4,indice))),...
((180/pi)*phi),20*log10(abs(F_tot_sol(:,5,indice))))

xlabel('\psi (deg)')
ylabel('Normalized power (dB)')
legend('f_0 - 2 * f_M' , 'f_0 - f_M' , 'f_0' , 'f_0 + f_M' , 'f_0 + 2 * f_M')
xlim([-90 90]);
ylim([-50 0]);
title('\delta = [0° -27.4° -58.6° -68.4°]')
%title('\delta = [0° 48.4° -58.6° -68.4°]')
%title('\delta = [0 0 0 0]')
end
%% Tabla resumen
if(Na == 2)
for i=1:num

```

```

        v1(i) = cn_sol(1,1,i);
        v2(i) = cn_sol(1,2,i);
        v3(i) = cin_sol(1,1,i);
        v4(i) = cin_sol(1,2,i);
    end
    v1 = v1';
    v2 = v2';
    v3 = v3';
    v4 = v4';
    T = table;
    T.DC_antenna_1 = v1;
    T.DC_antenna_2 = v2;
    T.Delay_antenna_1 = v3;
    T.Delay_antenna_2 = v4;
    T
end
if(Na == 4)
    for i=1:num
        v1(i) = cn_sol(1,1,i);
        v2(i) = cn_sol(1,2,i);
        v3(i) = cn_sol(1,3,i);
        v4(i) = cn_sol(1,4,i);
        v5(i) = cin_sol(1,1,i);
        v6(i) = cin_sol(1,2,i);
        v7(i) = cin_sol(1,3,i);
        v8(i) = cin_sol(1,4,i);
    end
    v1 = v1';
    v2 = v2';
    v3 = v3';
    v4 = v4';
    v5 = v5';
    v6 = v6';
    v7 = v7';
    v8 = v8';
    T = table;
    T.DC_antenna_1 = v1;
    T.DC_antenna_2 = v2;
    T.DC_antenna_3 = v3;
    T.DC_antenna_4 = v4;
    T.Delay_antenna_1 = v5;
    T.Delay_antenna_2 = v6;
    T.Delay_antenna_3 = v7;
    T.Delay_antenna_4 = v8;
    T
end
%% Guardar el Workspace
% filename = 'Luis_4_antennas_14_12_2020';
% save(filename)

```

En primer lugar, el usuario elegirá el número de elementos a considerar, tal y como se muestra en la Figura 6. En el caso de $N_a = 2$, después se elegirá el problema a solventar, algo que se explicará más adelante (véase también Figura 6). Por último, el usuario elegirá por teclado el desfase generado para cada una de las antenas que conforman el array (Figura 7).

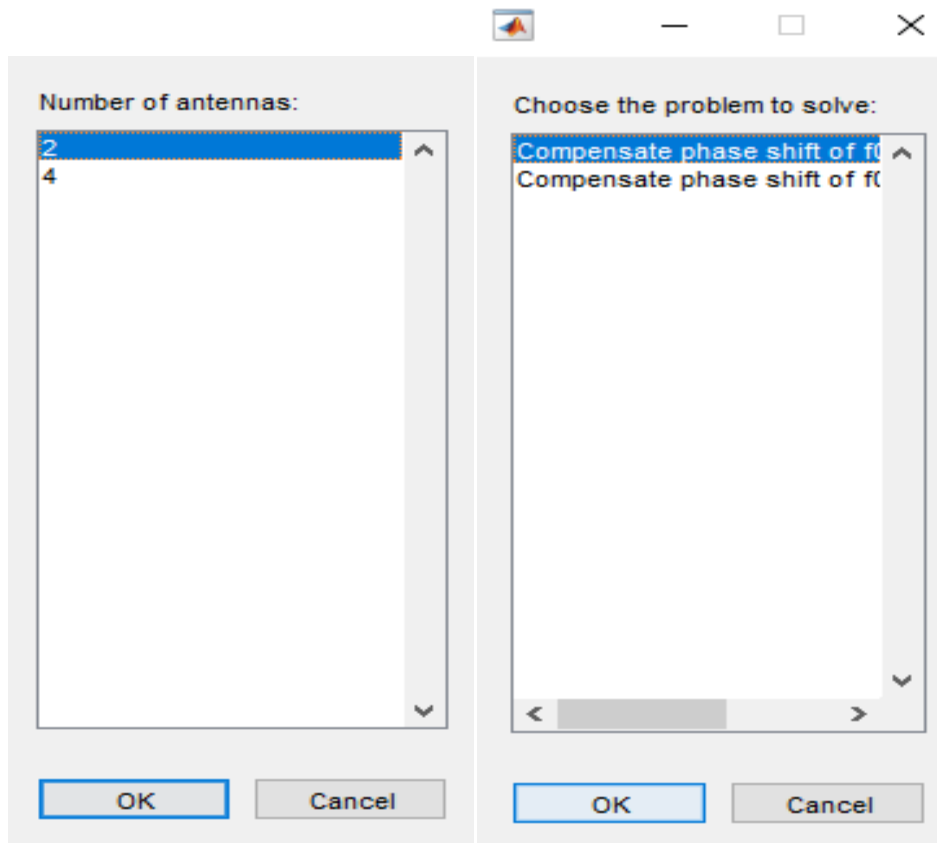


Figura 6. Paneles de selección.

```

For the antenna: 1
Type the phase shift you want to compensate of the antenna in degrees: 0
For the antenna: 2
Type the phase shift you want to compensate of the antenna in degrees: 0
For the antenna: 3
Type the phase shift you want to compensate of the antenna in degrees: 0
For the antenna: 4
Type the phase shift you want to compensate of the antenna in degrees: |

```

Figura 7. Selección por teclado del desfase de cada uno de los elementos del TMA.

Una vez elegidos los parámetros iniciales, en ambos casos se procederá de la misma forma. Para calcular el diagrama de radiación, se ha hecho uso de la ecuación (1) donde se expresa el diagrama de radiación en campo lejano como el producto del diagrama de radiación de un elemento individual, $E_0(r, \theta, \varphi)$ y el factor de array, $F(\theta, \varphi, t)$.

$$\begin{aligned}
 E(r, \theta, \varphi, t) &= E_0(r, \theta, \varphi) \sum_{k=1}^{N_a} A_k U_k(t) e^{j\beta k L \cos \psi} \\
 &= E_0(r, \theta, \varphi) F(\theta, \varphi, t)
 \end{aligned} \quad (1)$$

Las antenas empleadas son de tipo parche y tienen un diagrama de radiación omnidireccional en azimut. Por ello, para calcular el diagrama de radiación únicamente será necesario calcular el factor de array del conjunto. Para ello se ha hecho uso de la expresión (2):

$$\begin{aligned}
 F(\theta, \varphi, t) &= \sum_{h=-\infty}^{+\infty} F_h(\theta, \varphi, t) = \\
 &= \sum_{h=-\infty}^{+\infty} \sum_{k=1}^{N_a} A_n u_{hk} e^{j\beta k L \cos \psi} e^{j2\pi(f_0 + hf_M)t}
 \end{aligned} \tag{2}$$

La ecuación (2), relaciona el factor de array de nuestro TMA con los parámetros A_n que es una constante que depende de la antena, u_{hk} que son los coeficientes de Fourier calculados según la ecuación (3), β que es la constante de propagación y L que es la separación entre las antenas en metros. La segunda exponencial que aparece en (2) haría referencia a que frecuencia (f_0 , $f_0 \pm f_M$ etc) estamos evaluando el diagrama de radiación que buscamos obtener. Cabe destacar que esta expresión únicamente es válida en el caso de que no se produzca un desfase debido a la diferencia de caminos ópticos de las señales a transmitir por el array. En dicho caso, habría que introducir un término adicional delta, g , sumando a todo el exponente de la segunda exponencial que haga referencia a este hecho e introduzca dicho desfase.

$$u_{hn} = \begin{cases} c_n \operatorname{sinc}(h\pi c_n) e^{-jh\pi(c_n + 2c_{i_n})}, & \text{if } 0 \leq c_{i_n} \leq 1 - c_n \\ \frac{\left[\sin(h\pi(1 - c_{i_n})) e^{-jh\pi(1 + c_{i_n})} + \sin(h\pi(c_{i_n} + c_n - 1)) e^{-jh\pi(c_{i_n} + c_n - 1)} \right]}{h\pi}, & \text{if } 1 - c_n < c_{i_n} < 1 \end{cases} \tag{3}$$

Si observamos la ecuación (3), podemos ver que los coeficientes de Fourier dependen de los parámetros ya mencionados c_n y $c_{i,n}$ (que son los ciclos de trabajo y retardos relativos respectivamente). Realizando un barrido (mediante los bucles *for* del código arriba expuesto) sobre estos parámetros podremos evaluar cual es la solución óptima según sea el problema que se busque solventar en cada momento. En pocas palabras, se evaluará si los valores del diagrama de radiación en las posiciones deseadas son superiores (o inferiores) a unos umbrales establecidos para cada conjunto de parámetros que se esté evaluando en cada barrido.

Por último, cabe destacar que a lo largo de todo el trabajo todas las simulaciones hacen referencia por simplicidad al factor de array y no al diagrama de radiación completo ya que no se considera el término $E_0(r, \theta, \varphi)$ mencionado anteriormente. Esto es debido a que en el rango $[-90^\circ 90^\circ]$ puede asumirse omnidireccional.

3.2. SISTEMA DE RoF SOBRE TMA CON $N_a = 2$ ELEMENTOS EN MATLAB

3.2.1. ESPECIFICACIONES SOBRE EL CÓDIGO IMPLEMENTADO

Cabe destacar ciertos aspectos con respecto al código empleado para el caso en el que $N_a = 2$ elementos:

- 1) El usuario elegirá por teclado el valor del desfase entre los caminos ópticos que quiera compensar.

```
deltha(i) = input('Type the phase shift you want to  
compensate of the antenna in degrees: ');  
deltha(i) = deltha(i)*pi/180;
```

- 2) El 'step' elegido para los parámetros c_n y $c_{i,n}$ es de 0.01 (valor que establece un buen compromiso entre precisión y velocidad computacional).

```
c_sw = linspace(0,1,101); (DC = 0, 0.01, 0.02 ...  
Delay = 0, 0.01, 0.02 ...)
```

- 3) Se podrán establecer dos condiciones:

- Tanto la potencia normalizada de las curvas referidas a los modos $f_0 + f_M$ como $f_0 - f_M$ deben tener un valor menor que la variable 'threshold' en $\Psi = 0^\circ$ simultáneamente (AND o && condition).

```
if(abs(F_tot(length(phi)/2,1)).^2/max(abs(F_tot(:,1)).^2) <=  
threshold &&  
abs(F_tot(length(phi)/2,3)).^2/max(abs(F_tot(:,3)).^2) <=  
threshold)
```

- Tan solo una de las potencias normalizadas de las curvas referidas a los modos $f_0 + f_M$ como $f_0 - f_M$ debe tener un valor menor que la variable 'threshold' en $\Psi = 0^\circ$ simultáneamente (OR o || condition).

```
if(abs(F_tot(length(phi)/2,1)).^2/max(abs(F_tot(:,1)).^2) <=  
threshold ||  
abs(F_tot(length(phi)/2,3)).^2/max(abs(F_tot(:,3)).^2) <=  
threshold)
```

- 4) Para poder generar físicamente los trenes de pulsos se ha hecho uso de un microprocesador STM32 en el que los trenes de pulsos generados deben estar sincronizados. Esto implica que deben tener al menos un filo en común, ya sea de subida o de bajada (es decir, cuando el pulso que alimenta uno de los

elementos del array acaba el otro debe empezar o viceversa). Dicha limitación fue superada en el caso de $N_a = 4$

- 5) Se buscará maximizar la diferencia entre el máximo y el mínimo para el modo fundamental f_0 y también maximizar los modos $f_0 \pm f_M$ en $\Psi = \pm 90^\circ$. Esto se debe a que, para conseguir los objetivos de localización, es necesario que al realizar la diferencia entre los diagramas de radiación de los modos secundarios y el fundamental únicamente se produzca un máximo para así evitar ambigüedad.

3.2.2. RESULTADOS

En primer lugar, se ha buscado recrear el escenario original (el equivalente al obtenido mediante medidas en la Figura 5), con parámetros $g = 10^\circ$, $c_1 = 0.5$, $c_2 = 0.5$, $c_{i,1} = 0$ y $c_{i,2} = 0.5$. En la Figura 8 se muestra el resultado:

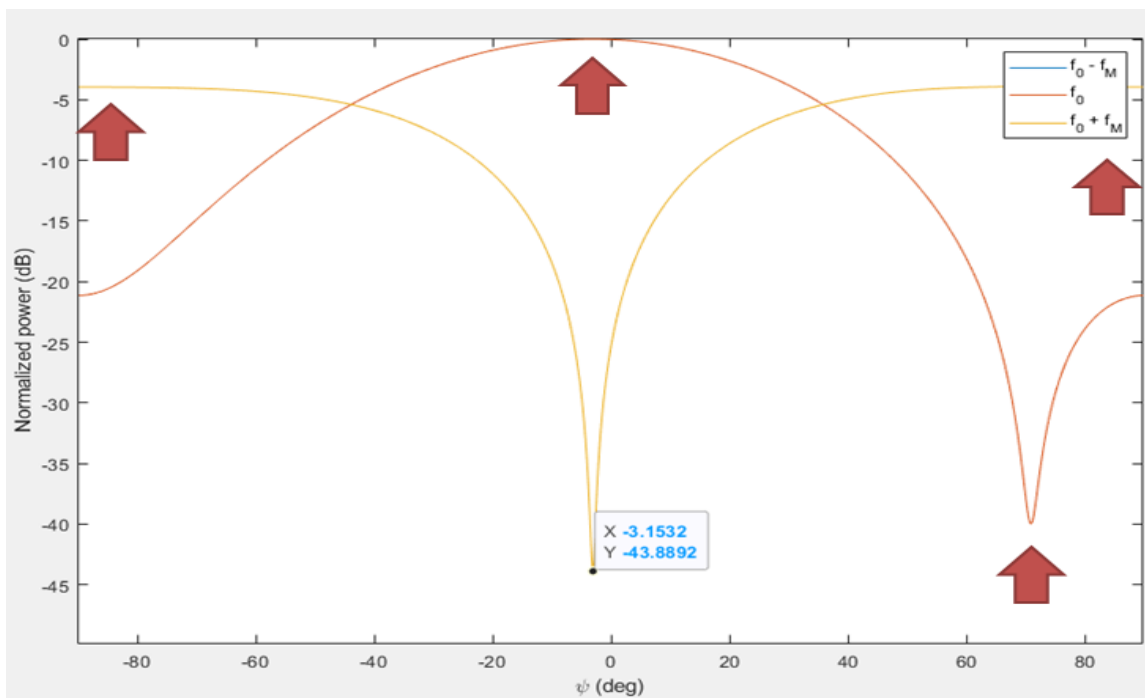


Figura 8. Simulación para la recreación del escenario del problema original.

Como puede verse, el mínimo de los modos secundarios se encuentra desplazado aproximadamente 3° como cabía esperarse. Además, las flechas y círculos rojos indican los puntos que deben optimizarse según las condiciones 4) y 5) expuestas anteriormente.

Si consideramos que ambos modos ($f_0 \pm f_M$) deben tener un valor mínimo simultáneamente en $\Psi = 0^\circ$ (AND condition) y realizamos un barrido de los parámetros c_n y $c_{i,n}$ de 101 puntos entre 0 y 1 obtenemos que no hay solución factible. Para solucionar esto existen dos alternativas:

- 1) Disminuir el umbral: Se obtendrían los resultados de la Figura 9 (Threshold = 10^{-2}). Este umbral resulta demasiado bajo para obtener resultados aceptables por lo que esta opción queda descartada.

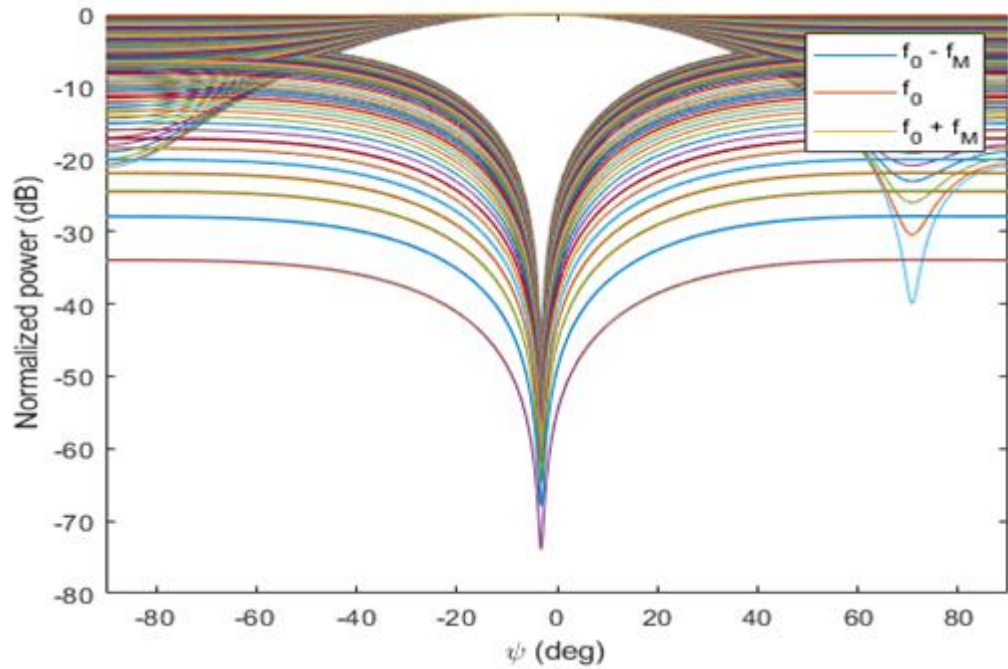


Figura 9. Compensación del desfase disminuyendo el umbral (threshold = 10^{-2}).

- 2) Modificar la condición de AND a OR. En este caso solo una de las dos componentes secundarias debe tener un valor por debajo del umbral para $\Psi = 0^\circ$. En este caso, al ser una condición mucho menos restrictiva, se obtienen 77 posibles soluciones (véase Figura 10).

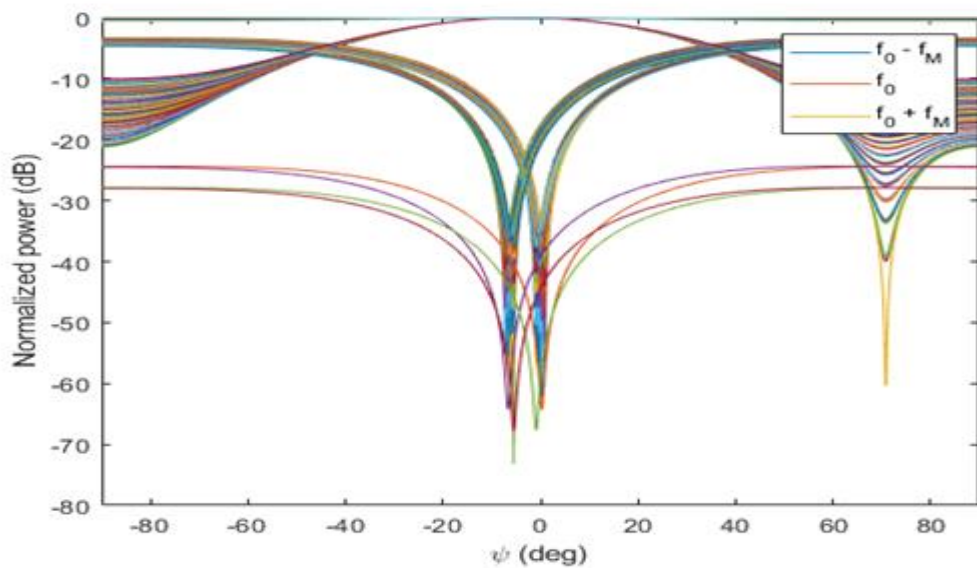


Figura 10. Compensación del desfase modificando la condición (OR condition).

Los distintos valores de los parámetros c_n y $c_{i,n}$ para cada una de las soluciones serían los que aparecen en la Tabla 1:

<u>DC_antenna_1</u>	<u>DC_antenna_2</u>	<u>Delay_antenna_1</u>	<u>Delay_antenna_2</u>
0.4	0.55	0	0.4
0.41	0.54	0	0.41
0.41	0.53	0	0.41
0.42	0.54	0	0.42
0.42	0.53	0	0.42
0.42	0.52	0	0.42
0.43	0.53	0	0.43
0.43	0.52	0	0.43
0.43	0.51	0	0.43
0.44	0.52	0	0.44
0.44	0.51	0	0.44
0.44	0.5	0	0.44
0.44	0.49	0	0.44
0.45	0.51	0	0.45
0.45	0.5	0	0.45
0.45	0.49	0	0.45
0.45	0.48	0	0.45
0.46	0.5	0	0.46
0.46	0.49	0	0.46
0.46	0.48	0	0.46
0.46	0.47	0	0.46
0.47	0.49	0	0.47
0.47	0.48	0	0.47
0.47	0.47	0	0.47
0.47	0.46	0	0.47
0.48	0.48	0	0.48
0.48	0.47	0	0.48
0.48	0.46	0	0.48
0.48	0.45	0	0.48
0.49	0.47	0	0.49
0.49	0.46	0	0.49
0.49	0.45	0	0.49
0.49	0.44	0	0.49
0.5	0.46	0	0.5
0.5	0.45	0	0.5

0.5	0.44	0	0.5
0.5	0.43	0	0.5
0.51	0.45	0	0.51
0.51	0.44	0	0.51
0.51	0.43	0	0.51
0.51	0.42	0	0.51
0.52	0.44	0	0.52
0.52	0.43	0	0.52
0.52	0.42	0	0.52
0.52	0.41	0	0.52
0.53	0.43	0	0.53
0.53	0.42	0	0.53
0.53	0.41	0	0.53
0.53	0.4	0	0.53
0.54	0.42	0	0.54
0.54	0.41	0	0.54
0.54	0.4	0	0.54
0.54	0.39	0	0.54
0.55	0.41	0	0.55
0.55	0.4	0	0.55
0.55	0.39	0	0.55
0.56	0.4	0	0.56
0.56	0.39	0	0.56
0.56	0.38	0	0.56
0.57	0.39	0	0.57
0.57	0.38	0	0.57
0.57	0.37	0	0.57
0.58	0.38	0	0.58
0.58	0.37	0	0.58
0.58	0.36	0	0.58
0.59	0.37	0	0.59
0.59	0.36	0	0.59
0.59	0.35	0	0.59
0.6	0.36	0	0.6
0.6	0.35	0	0.6
0.61	0.35	0	0.61
0.61	0.34	0	0.61
0.62	0.34	0	0.62
0.62	0.33	0	0.62
0.63	0.33	0	0.63
0.97	0.03	0	0.94
0.98	0.02	0	0.96

Tabla 1. Soluciones para la compensación del desfase en el caso OR.

En primer lugar, cabe destacar el valor de $c_{i,1}$ que siempre es igual a 0. Esto se debe al retardo al que hace referencia este parámetro es el relativo entre ambas antenas y no al retardo absoluto. Por ello, uno de los elemento siempre tendrá fijo este parámetro con un valor nulo.

De entre todas estas soluciones, y como se indicó anteriormente, se ha elegido la que maximiza la diferencia entre el máximo y el mínimo para el modo fundamental f_0 y también maximiza los modos $f_0 \pm f_M$ en $\Psi = \pm 90^\circ$. El resultado se muestra en la Figura 11 y se corresponde con los parámetros $c_1 = 0.49$, $c_2 = 0.47$, $c_{i,1} = 0$ y $c_{i,2} = 0.49$.

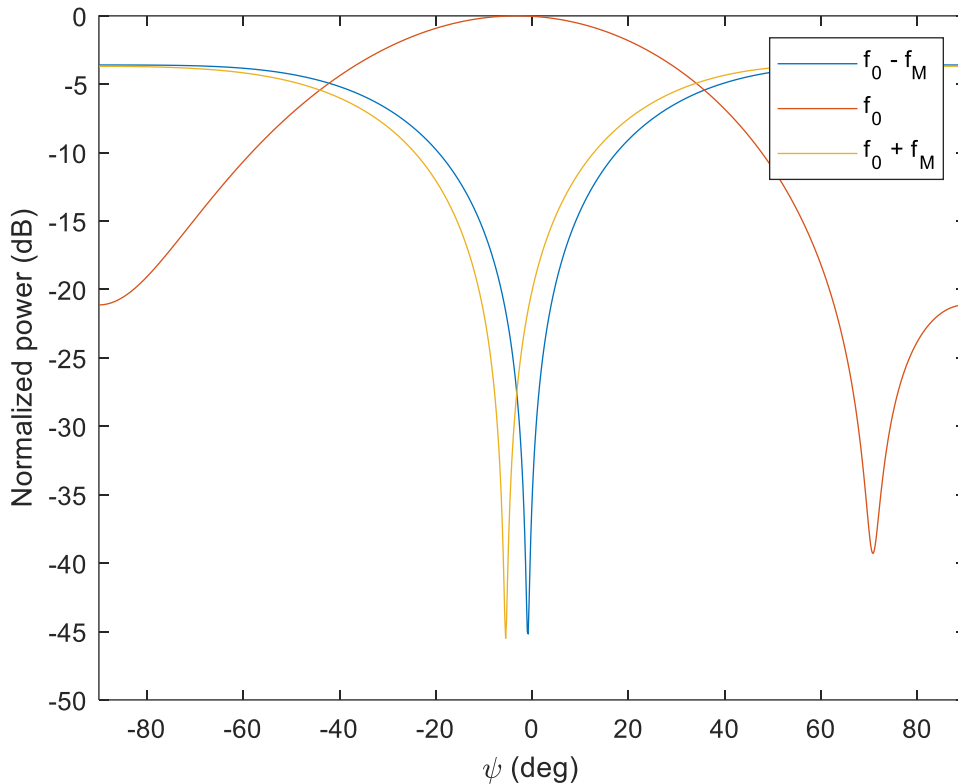


Figura 11. Solución óptima de la compensación del desfase modificando la condición (OR condition).

Sin embargo, actualmente para conseguir las aplicaciones de localización deseadas, conseguir compensar el desfase de solo uno de los modos resulta insuficiente. Esto se debe a que para conseguir esto se busca realizar la diferencia entre los diagramas de radiación de f_0 y $f_0 \pm f_M$ y buscar el máximo resultante. En caso de coincidir ambos mínimos de las componentes secundarias, podría generarse ambigüedad. Es por ello que a la vista de que no resulta posible compensar ambos modos simultáneamente con únicamente dos elementos, se ha procedido al estudio con un sistema de 4 elementos. Aun con ello, estos resultados podrán servir de ayuda para futuras investigaciones.

3.3. SISTEMA DE RoF SOBRE TMA CON $N_a = 4$ ELEMENTOS EN MATLAB

3.3.1. ESPECIFICACIONES SOBRE EL CÓDIGO IMPLEMENTADO

En el caso de $N_a = 4$ antenas, los objetivos buscados son diferentes a los del caso anterior. Esto se debe a que al tener un mayor número de elementos en el array, se dispone de un mayor grado de libertad (hay más parámetros que pueden ajustarse y con ello el diagrama final puede modificarse más libremente). Por ello cabe destacar algunos puntos.

- 1) El '*step*' elegido para los parámetros c_n y $c_{i,n}$ debe ser considerablemente menor que en el caso anterior ya que al tener más antenas se necesitan 7 bucles '*for*' anidados (lo que implica un coste computacional mucho más alto).
- 2) En este caso el parámetro h_{aux} que indica el número de coeficientes de Fourier a computar, y con ello el número de curvas o modos a obtener ($f_0 \pm h \cdot f_M$), está fijado en un valor de 5 ya que buscamos apuntar simultáneamente en 5 direcciones distintas simultáneamente. En el caso de $N_a = 2$, h_{aux} tenía un valor de 3 coeficientes.

```
if (Na==2)
    phi = linspace((-90*2*pi)/360, (90*2*pi)/360, 1000); %Phi es la variable
    c_sw = linspace(0, 1, 101);
    haux = 3; %Numero de coeficientes de Fourier a considerar
else
    phi = linspace((-90*2*pi)/360, (90*2*pi)/360, 100); %Phi es la variable
    c_sw = linspace(0, 0.25, 6);
    haux = 5; %Numero de coeficientes de Fourier a considerar
end
```

- 3) Las limitaciones de sincronización son superadas debido a una óptima programación del microcontrolador como se explicará más adelante,
- 4) Ahora se buscará únicamente que los modos secundarios tengan un valor máximo tan alto como sea posible en los valores de Ψ deseados. Para ello se elegirán unos umbrales y se determinará si el valor de cada modo en ese valor de Ψ es mayor o menor.

```

% delta = [0 -27.4 -58.6 -68.4]
diferencia_1(num) = 20*log10(abs(F_tot(39,2))/max(abs(F_tot(:,3)))); %Normalizado (a -20°)
diferencia_2(num) = 20*log10(abs(F_tot(67,4))/max(abs(F_tot(:,3)))); %Normalizado (a 30°)
diferencia_3(num) = 20*log10(abs(F_tot(39,3))/max(abs(F_tot(:,3)))); %Normalizado (a -20°)
diferencia_4(num) = 20*log10(abs(F_tot(67,3))/max(abs(F_tot(:,3)))); %Normalizado (a 30°)
diferencia_5(num) = 20*log10(abs(F_tot(39,4))/max(abs(F_tot(:,3)))); %Normalizado (a -20°)
diferencia_6(num) = 20*log10(abs(F_tot(67,2))/max(abs(F_tot(:,3)))); %Normalizado (a 30°)
diferencia_7(num) = 20*log10(abs(F_tot(28,1))/max(abs(F_tot(:,3)))); %Normalizado (a -40°)
diferencia_8(num) = 20*log10(abs(F_tot(78,5))/max(abs(F_tot(:,3)))); %Normalizado (a 50°)
diferencia_9(num) = 20*log10(abs(F_tot(28,3))/max(abs(F_tot(:,3)))); %Normalizado (a -40°)
diferencia_10(num) = 20*log10(abs(F_tot(78,3))/max(abs(F_tot(:,3)))); %Normalizado (a 50°)
diferencia_11(num) = 20*log10(abs(F_tot(28,5))/max(abs(F_tot(:,3)))); %Normalizado (a -40°)
diferencia_12(num) = 20*log10(abs(F_tot(78,1))/max(abs(F_tot(:,3)))); %Normalizado (a 50°)

if (num == 1)
    if((diferencia_3(num) <= threshold_3 && diferencia_4(num) <= threshold_3 && ...
        ((diferencia_1(num) >= threshold_2 && diferencia_2(num) >= threshold_2) || ...
        (diferencia_5(num) >= threshold_2 && diferencia_6(num) >= threshold_2))) && ...
        diferencia_9(num) <= threshold_3 && diferencia_10(num) <= threshold_3 && ...
        ((diferencia_7(num) >= threshold_4 && diferencia_8(num) >= threshold_4) || ...
        (diferencia_11(num) >= threshold_4 && diferencia_12(num) >= threshold_4)))

```

3.3.2. RESULTADOS

En primer lugar, para poder realizar las simulaciones, fue necesario realizar la medida del desfase relativo producido entre los distintos caminos ópticos. Para ello se empleó un analizador de redes vectorial y un software denominado ‘S2VNA’. Tras realizar las medidas se obtuvo que para las longitudes de los cables a emplear el desfase relativo entre los elementos del array era de 0° , -27.4° , -58.6° y -68.4° respectivamente (valores determinados por las ligeras diferencias de longitud entre las fibras ópticas empleadas). Una vez conocidos estos parámetros, pudieron ser introducidos en la simulación.

Este desfase produce que los máximos de radiación de los diferentes modos se desplacen y por ello ya no estén simétricamente distribuidos en torno a $\Psi = 0^\circ$ [5]. Por ello haciendo uso de un barrido de los parámetros c_n y c_{in} se ha buscado obtener los máximos de radiación en torno a los siguientes valores (escogidos arbitrariamente ya que, como se verá más adelante, podrán ser modificados debido a las propiedades de ‘beam-steering’ del array):

- 1) $f_0 - 2*f_M$ a -40°
- 2) $f_0 - f_M$ a -20°
- 3) f_0 se mantiene constante en 8°
- 4) $f_0 + f_M$ a 30°
- 5) $f_0 + 2*f_M$ a 50°

El resultado óptimo obtenido para el apuntamiento múltiple se muestra en la Figura 12 donde los parámetros c_n y c_{in} son $c_1 = 0.02$, $c_2 = 0.07$, $c_3 = 0.07$, $c_4 = 0.02$ y $c_{i,1} = 0$, $c_{i,2} = 0.15$, $c_{i,3} = 0.3$, $c_{i,4} = 0.45$ respectivamente.

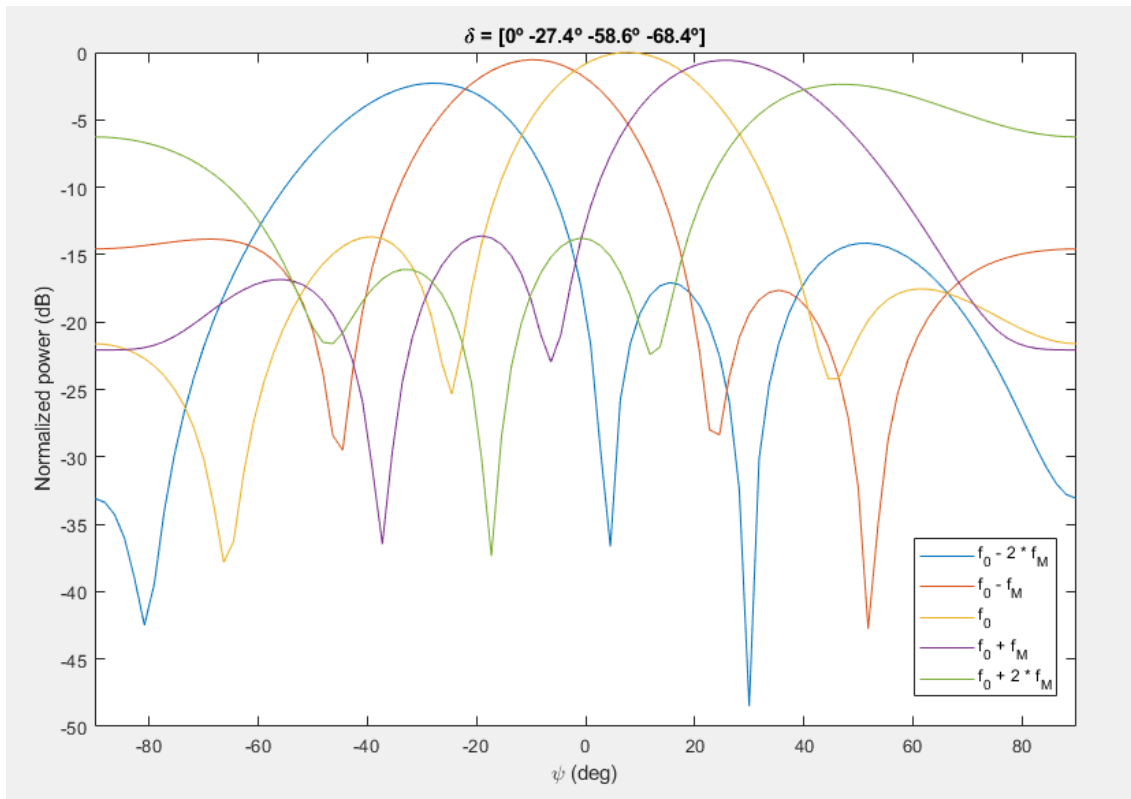


Figura 12. Diagrama de radiación óptimo para el apuntamiento simultáneo en 5 direcciones.

Sin embargo, aunque esta secuencia proporciona los resultados óptimos en relación a los máximos de radiación normalizados, presenta el inconveniente de que la ganancia de dichos modos es muy pequeña ya que los ciclos de trabajo tienen valores muy pequeños. Por ello, como se verá más adelante, se han puesto a prueba otras secuencias con el fin de solventar dicho inconveniente.

Además también se demostrará el hecho de que variaciones en la secuencia proporcionan variaciones en las direcciones de apuntamiento. Esto resultará muy útil ya que implicará que simplemente modificando los parámetros c_n y $c_{i,n}$ será posible re direccionar los máximos en las direcciones deseadas.

4. DESARROLLO EXPERIMENTAL EN EL LABORATORIO

Una vez realizadas las simulaciones, el siguiente paso fue comprobar los resultados mediante equipos reales en el laboratorio.

4.1. SISTEMA DE RoF SOBRE TMA CON $N_a = 2$ ELEMENTOS EN EL LABORATORIO

En primer lugar, se consideró el caso más sencillo, es decir, $N_a = 2$ antenas. El espacio de trabajo se muestra en las Figuras 13, 14 y 15. En primer lugar debe situarse las antenas tipo parche debidamente alineadas al dispositivo de rotación que será el encargado de realizar el barrido en azimut.

Una vez hecho esto, se han de conectar los fotoreceptores tipo PIN que van conectados por medio de la fibra al láser que modulará la señal generada por el transmisor. Es muy importante conseguir que los distintos elementos estén fijos ya que de lo contrario el movimiento de rotación podría provocar movimientos en los elementos y con ello causar alteraciones en las medidas.

Por último, debe situarse la antena receptora de tipo bocina debidamente alineada con el TMA y conectar las fuentes de alimentación.



Figura 13. Lugar de trabajo en el laboratorio (1).



Figura 14. Lugar de trabajo en el laboratorio (2).



Figura 15. Lugar de trabajo en el laboratorio (3).

Para poder generar los trenes de pulsos que actúan sobre el TMA se ha empleado un microcontrolador STM32 que es programado a través de un software llamado ‘*Atolic True Studio v9.0*’. A través de él pueden cargarse los coeficientes c_n y $c_{i,n}$ previamente calculados en MATLAB en el microprocesador. En este caso se empleó un código

previamente diseñado que establece la limitación de que los pulsos deben estar sincronizados.

En primer lugar se ha realizado una calibración del sistema recreando el escenario original con los parámetros $c_1 = 0.5$, $c_2 = 0.5$, $c_{i,1} = 0$ y $c_{i,2} = 0.5$ al igual que en la Figura 5. En este caso se ha obtenido la Figura 16:

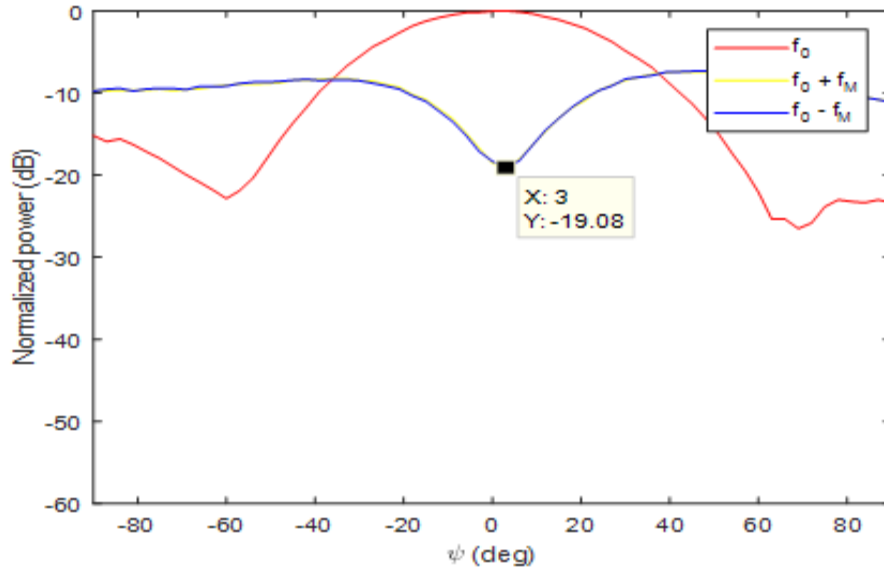


Figura 16. Calibración del sistema en el laboratorio.

En este caso obtenemos el mismo diagrama de radiación pero invertido, ya que el mínimo se encuentra en $\Psi = 3^\circ$ y no en $\Psi = -3^\circ$. Esto se debe a que en nuestro caso el desfase de caminos ópticos o q es de $[10^\circ \ 0^\circ]$ en vez de $[0^\circ \ 10^\circ]$ como en el caso anteriores (donde el primer valor es el referido a la primera antena y el segundo a la segunda). En definitiva, al invertir el desfase se invierte el diagrama de radiación.

Una vez calibrado el sistema, se procedió a recrear el escenario de la Figura 11 con los parámetros $c_1 = 0.49$, $c_2 = 0.47$, $c_{i,1} = 0$ y $c_{i,2} = 0.49$ obteniendo los resultados de la Figura 17.

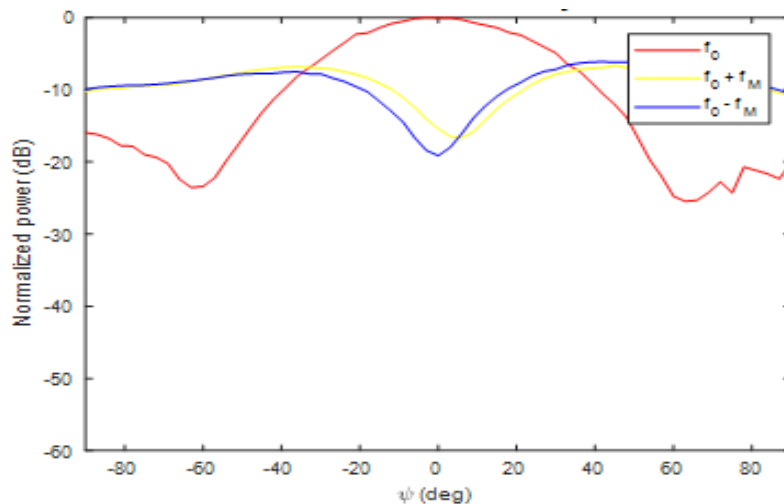


Figura 17. Resultados de la secuencia óptima puesta a prueba en el laboratorio ($N_a = 2$).

Al analizar los resultados, se aprecian dos diferencias entre las figuras. La primera es la ya anteriormente comentada; los diagramas están invertidos debidos a que el desfase está invertido. La segunda sería que también las curvas azul y amarilla están al revés. Esto se produce debido a que los trenes de pulsos están también invertidos. Es decir mientras que en la Figura 11 los coeficientes son $c_1 = 0.49$, $c_2 = 0.47$, $c_{i,1} = 0$ y $c_{i,2} = 0.49$ en la Figura 18 son $c_1 = 0.47$, $c_2 = 0.49$, $c_{i,1} = 0.49$ y $c_{i,2} = 0$. En la simulación de la Figura 18. se comprueba este hecho.

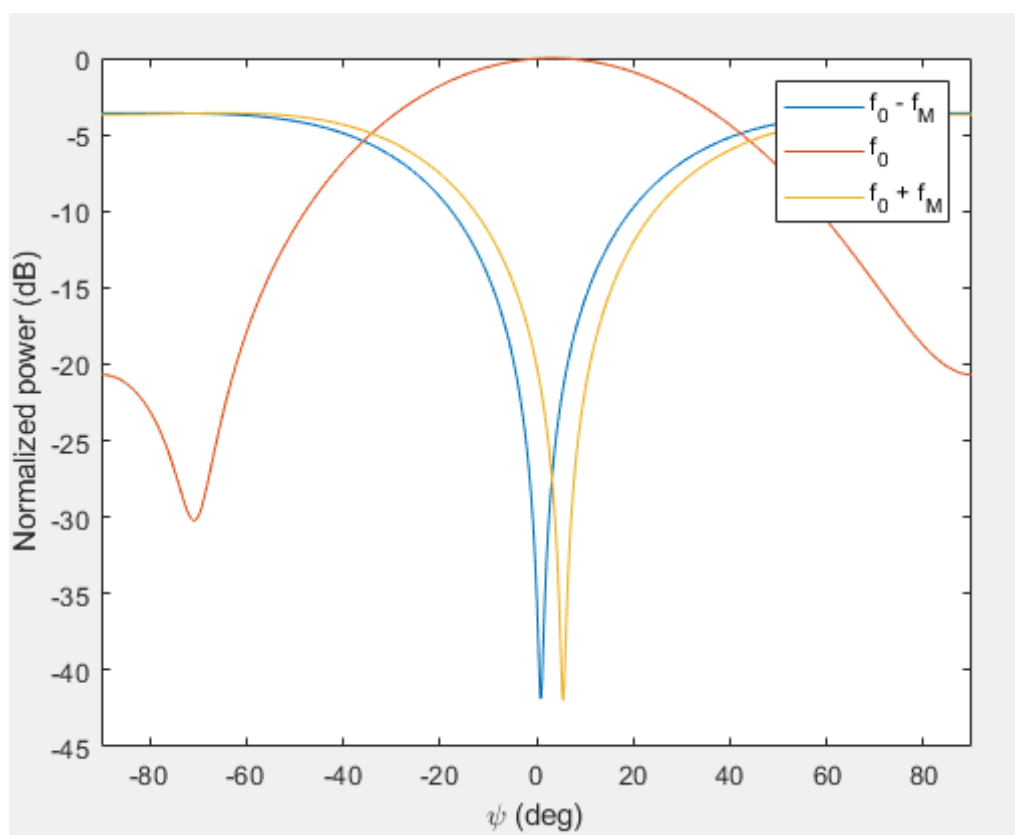


Figura 18. Simulación de la inversión de modos.

4.1. SISTEMA DE RoF SOBRE TMA CON $N_a = 4$ ELEMENTOS EN EL LABORATORIO

En el caso de $N_a = 4$, fue necesario reprogramar el microcontrolador STM32 debido a que la solución obtenida requería necesariamente que los trenes de pulsos no debieran estar sincronizados. Para ello se ha hecho uso del siguiente código programado en C++ en el ya mencionado software 'Atollic Studio'.

```

#include "stm32f10x.h"

void PA0_Configuration(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitTypeDef GPIO_InitStructure1;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
    /* Configure the GPIO PAx */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0|GPIO_Pin_6|GPIO_Pin_9; //
Timer 2 (1) , 3 (1) and 1 (2)
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure); /* GPIOA refers to PA0, PA1,
... */

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    /* Configure the GPIO PBx */
    GPIO_InitStructure1.GPIO_Pin = GPIO_Pin_6; // Timer 4 (1)
    GPIO_InitStructure1.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_InitStructure1.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure1); /* GPIOB refers to PB0, PB1,
... */
}

static void TIM7_Config(void){
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure_TIM7;
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM7,ENABLE);
    TIM_TimeBaseStructure_TIM7.TIM_Period = 65536-1; //Max 2^16 - 1;
16bits
    TIM_TimeBaseStructure_TIM7.TIM_Prescaler = 24-1; //1 us
    TIM_TimeBaseStructure_TIM7.TIM_ClockDivision = 0;
    TIM_TimeBaseStructure_TIM7.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM7, &TIM_TimeBaseStructure_TIM7);
}

void delay_us (uint16_t us){
    TIM_SetCounter(TIM7,0);
    TIM_Cmd(TIM7,ENABLE);
    while(TIM_GetCounter(TIM7)<us);
    TIM_Cmd(TIM7,DISABLE);
}

int main(void)
{
    //BEEMSTEARING

    //uint16_t DC1 = 26;
    //uint16_t DC2 = 24;
    //uint16_t DC3 = 25;
    //uint16_t DC4 = 24;
    //uint16_t DC1 = 25;
    //uint16_t DC2 = 23;
    //uint16_t DC3 = 25;
    //uint16_t DC4 = 25;
    //uint16_t DC1 = 25;
    //uint16_t DC2 = 25;
    //uint16_t DC3 = 26;
    //uint16_t DC4 = 23;
}

```

```

uint16_t DC1 = 23;
uint16_t DC2 = 26;
uint16_t DC3 = 26;
uint16_t DC4 = 24;

//uint16_t DELAY1 = 22;
//uint16_t DELAY2 = 21;
//uint16_t DELAY3 = 2;
//uint16_t DELAY1 = 14;
//uint16_t DELAY2 = 12;
//uint16_t DELAY3 = -6;
//uint16_t DELAY1 = 17;
//uint16_t DELAY2 = 18;
//uint16_t DELAY3 = 0;

uint16_t DELAY1 = 17;
uint16_t DELAY2 = 19;
uint16_t DELAY3 = 1;

uint32_t Frequency = 10000; // Desired frequency [Hz] (remember
24 MHz is the bus frequency)
uint32_t TimerPeriod = 24000000/Frequency; // Timer period
(work as long as the prescaler is set to 0)
uint16_t DutyCycleOffset = 0; // Duty cycle offset (first
antenna: 50%-offset; second antenna: 50%-offset with 180° phase shift) [max
50 %]
uint16_t DCAntenna1 = (TimerPeriod*(DC1-DutyCycleOffset))/100-
1;
uint16_t DCAntenna2 = (TimerPeriod*(DC2+DutyCycleOffset))/100-
1;
uint16_t DCAntenna3 = (TimerPeriod*(DC3+DutyCycleOffset))/100-
1;
uint16_t DCAntenna4 = (TimerPeriod*(DC4+DutyCycleOffset))/100-
1;

TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
TIM_OCInitTypeDef TIM_OCInitStructure;
TIM_OCInitTypeDef TIM_OCInitStructure_TIM_3_4;
uint16_t PrescalerValue = 0;
PA0_Configuration();

//NVIC_Configuration();
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE);

/* Compute the prescaler value */
PrescalerValue = (uint16_t) 0;

/* Time base configuration */
TIM_TimeBaseStructure.TIM_Period = TimerPeriod-1;
TIM_TimeBaseStructure.TIM_Prescaler = PrescalerValue;
TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);
TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);

```

```

TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);
TIM7_Config();

/* Output Compare Timing Mode configuration: Channel1. NB: all
numbers refer to specific hardware (i.e. TIM_OCMode_PWM1) */
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TIM_OCInitStructure_TIM_3_4.TIM_OCMode = TIM_OCMode_PWM1;;

/* OC1 signal is output on the corresponding output pin */
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = 0;
TIM_OCInitStructure_TIM_3_4.TIM_OutputState =
TIM_OutputState_Enable;
TIM_OCInitStructure_TIM_3_4.TIM_Pulse = 0;

/* OC1 active high */
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High; /*
NB: OC1=PA0, OC2=PA1, ... Polarity high=first signal part high and second
signal part low. Polarity low=opposite of high */
TIM_OCInitStructure_TIM_3_4.TIM_OCPolarity =
TIM_OCPolarity_High;
TIM_OC1Init(TIM2, &TIM_OCInitStructure);
TIM_OC1Init(TIM3, &TIM_OCInitStructure_TIM_3_4);
TIM_OC1Init(TIM4, &TIM_OCInitStructure_TIM_3_4);
TIM_OC2Init(TIM1, &TIM_OCInitStructure_TIM_3_4);

/* TIMx_CCR1 can be written at anytime, the new value is taken
in account immediately */
TIM_OC1PreloadConfig(TIM2, TIM_OCPreload_Disable);
TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Disable);
TIM_OC1PreloadConfig(TIM4, TIM_OCPreload_Disable);
TIM_OC2PreloadConfig(TIM1, TIM_OCPreload_Disable);
TIM_ITConfig(TIM2, TIM_IT_CC1, ENABLE);
TIM_ITConfig(TIM3, TIM_IT_CC1, ENABLE);
TIM_ITConfig(TIM4, TIM_IT_CC1, ENABLE);
TIM_ITConfig(TIM1, TIM_IT_CC2, ENABLE);

// Initialize channel 2 and channel 3
//TIM_OC2Init(TIM2, &TIM_OCInitStructure);
//TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High; //
Remember to change polarity or other parameters before initializing OCn
//TIM_OC3Init(TIM2, &TIM_OCInitStructure);

/* TIMx_CCR2 can be written at anytime, the new value is taken
in account immediately */
//TIM_OC2PreloadConfig(TIM2, TIM_OCPreload_Disable);
//TIM_ITConfig(TIM2, TIM_IT_CC2, ENABLE);

/* TIMx_CCR3 can be written at anytime, the new value is taken
in account immediately */
//TIM_OC3PreloadConfig(TIM2, TIM_OCPreload_Disable);
//TIM_ITConfig(TIM2, TIM_IT_CC3, ENABLE);

TIM_BDTRInitTypeDef TIM_BDTRInitStruct; //With this 5 lines we
make timer 1 to work as general purpose timers
TIM_BDTRStructInit(&TIM_BDTRInitStruct);
TIM_BDTRConfig(TIM1, &TIM_BDTRInitStruct);
TIM_CCPreloadControl(TIM1, ENABLE);
TIM_CtrlPWMOutputs(TIM1, ENABLE);

```

```

/* TIM2 enable counter */
TIM_Cmd(TIM2, ENABLE);
delay_us(DELAY1-8);
TIM_Cmd(TIM3, ENABLE);
delay_us(DELAY2-9);
TIM_Cmd(TIM4, ENABLE);
delay_us(DELAY3+11); //Because TIM1 is quicker
TIM_Cmd(TIM1, ENABLE);
TIM_SetCompare1(TIM2, DCAntenna1);
TIM_SetCompare1(TIM3, DCAntenna2);
TIM_SetCompare1(TIM4, DCAntenna3);
TIM_SetCompare2(TIM1, DCAntenna4);

while(1);
return(0);
}

```

A partir de este código se consiguen programar de manera independiente los ‘Timers’ del microcontrolador que son los encargados de generar periódicamente las muestras de los trenes de pulsos. Únicamente han de seleccionarse apropiadamente las variables DELAYx y DCx.

Una vez solventada esta limitación, se procedió a poner la secuencia expuesta en la Figura 12 realizando el barrido en azimuth del TMA. Los resultados se muestran en la Figura 19 ($c_1 = 0.02$, $c_2 = 0.07$, $c_3 = 0.07$, $c_4 = 0.02$ y $c_{i,1} = 0$, $c_{i,2} = 0.15$, $c_{i,3} = 0.3$, $c_{i,4} = 0.45$).

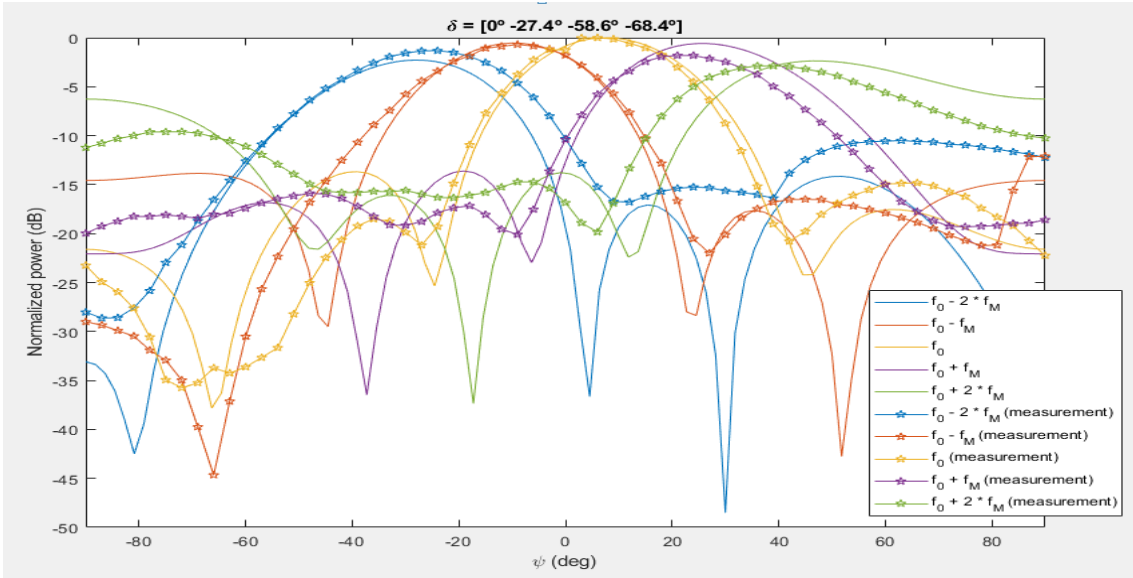


Figura 19. Resultados de la secuencia óptima puesta a prueba en el laboratorio ($N_a = 4$).

Como puede apreciarse claramente, los resultados obtenidos coinciden con los esperados ya que los máximos de radiación se encuentran en las posiciones deseadas. Sin embargo, para poder solventar los problemas de potencia anteriormente

mencionados se pusieron a prueba otras secuencias mientras simultáneamente se trataba de demostrar el hecho de que variaciones en la secuencia producen variaciones en las posiciones de los máximos de radiación. Esto se conoce como ‘*beam-steering*’. A continuación se muestran los resultados en las Figuras 19, 20, 21 y 22 (donde todas las direcciones de Ψ están referidas a la dirección ‘*broadside*’):

Apuntar: [60° 30° 0° -30° -60°]

$c_1 = 0.2588, c_2 = 0.2386, c_3 = 0.2462, c_4 = 0.2394$ y $c_{i,1} = 0.2311, c_{i,2} = 0.4577, c_{i,3} = 0.6715, c_{i,4} = 0.8901$.

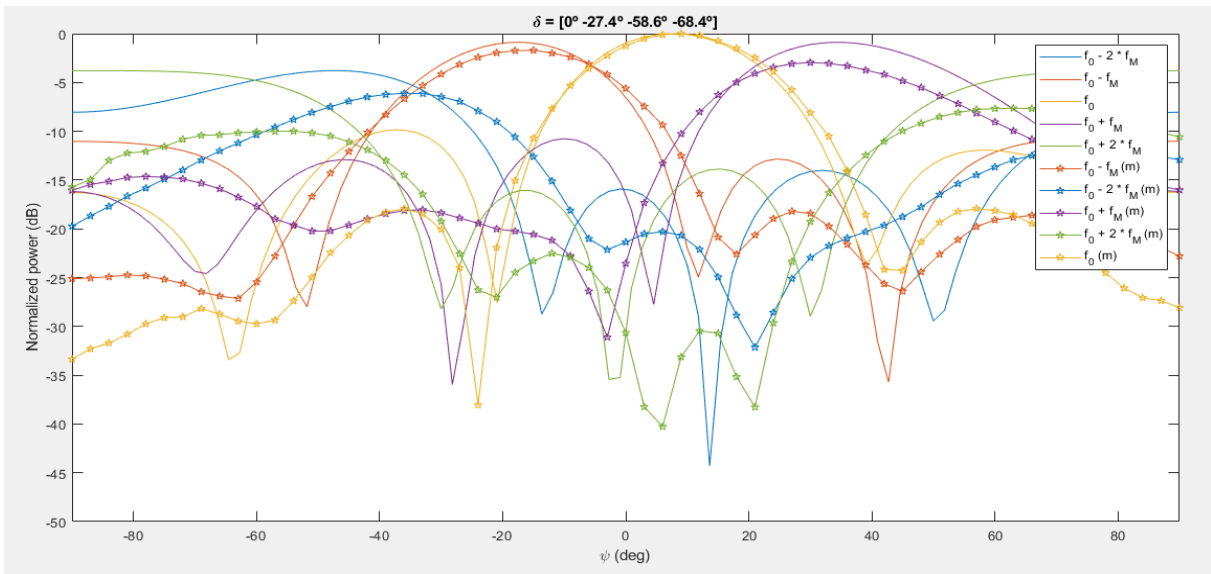


Figura 20. Beam-steering (1).

Apuntar: [30° 20° 0° -20° -30°]

$c_1 = 0.2532, c_2 = 0.2272, c_3 = 0.2526, c_4 = 0.2494$ y $c_{i,1} = 0.1949, c_{i,2} = 0.3351, c_{i,3} = 0.4452, c_{i,4} = 0.5732$.

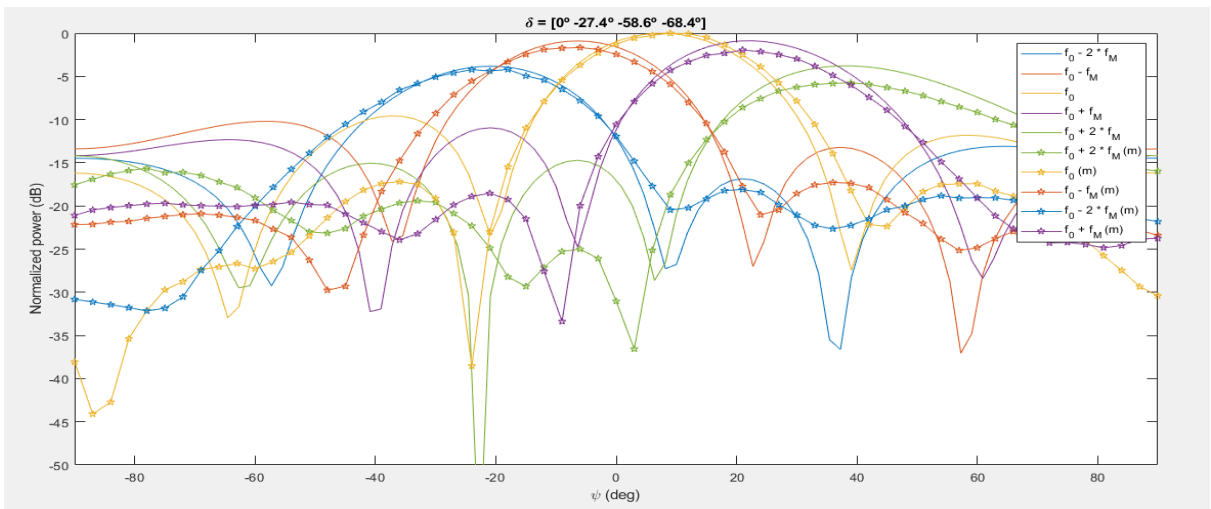


Figura 21. Beam-steering (2).

Apuntar: [45° 20° 0° -20° -45°]

$c_1 = 0.2520, c_2 = 0.2471, c_3 = 0.2557, c_4 = 0.2275$ y $c_{i,1} = 0.4605, c_{i,2} = 0.6420, c_{i,3} = 0.8144, c_{i,4} = 0.0046$.

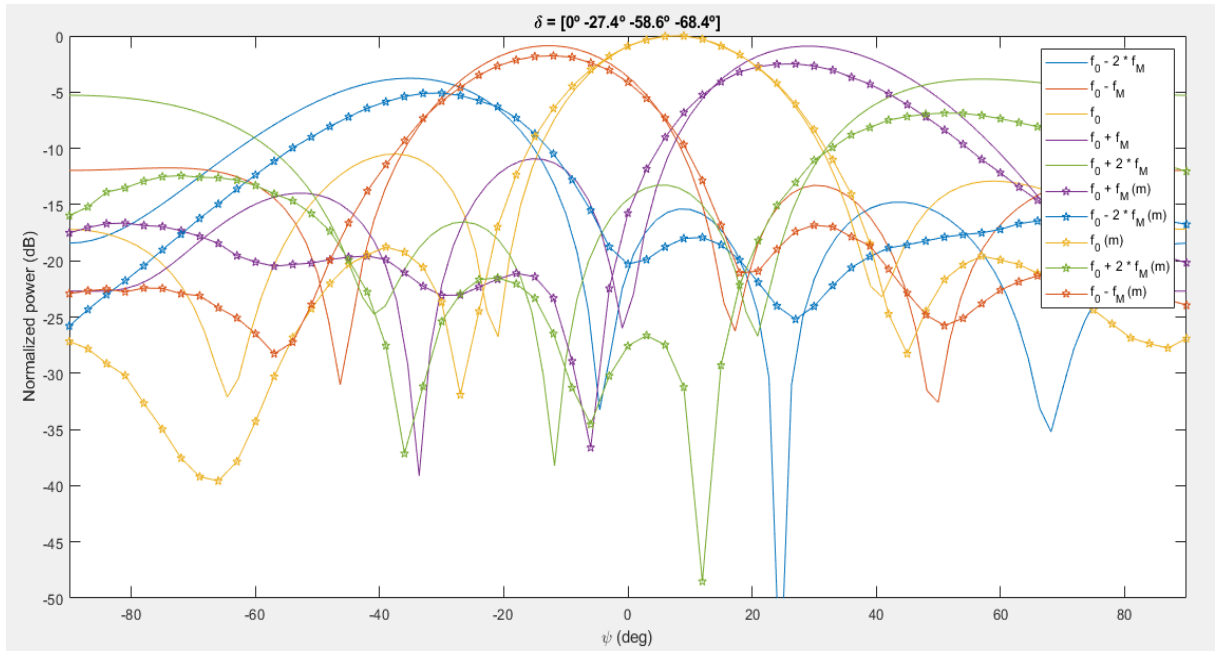


Figura 22. Beam-steering (3).

Apuntar: [50° 20° 0° -20° -50°]

$c_1 = 0.2336, c_2 = 0.2567, c_3 = 0.2555, c_4 = 0.2366$ y $c_{i,1} = 0.3903, c_{i,2} = 0.5696, c_{i,3} = 0.7606, c_{i,4} = 0.9631$.

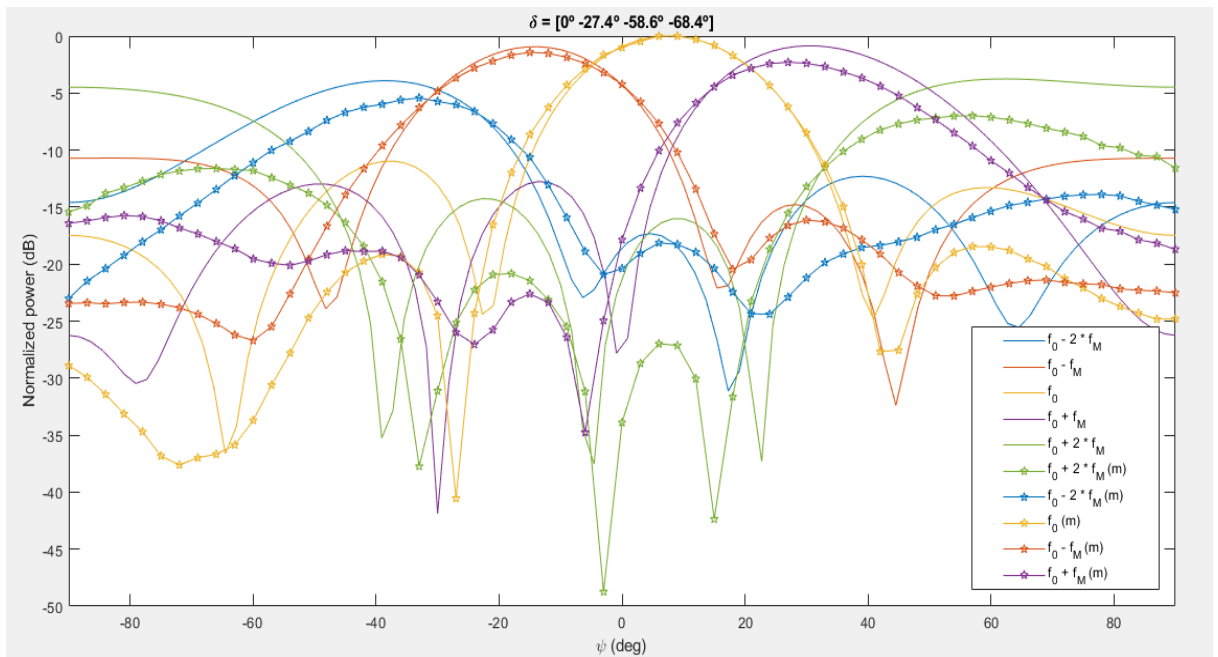


Figura 23. Beam-steering (4).

Por último, también se realizó un estudio similar al que se realizó con el caso de $N_a = 2$ antenas donde trató de demostrarse si variaciones en las secuencias pueden provocar una compensación adecuada en el desfase producido. Para ello se procedió de la siguiente manera:

- 1) Se puso a prueba una secuencia con la que se esperaban buenos resultados con el desfase original ($c_n = [0.34 \ 0.3 \ 0.29 \ 0.2]$, $c_{i,n} = [0.47 \ 0.65 \ 0.82 \ 0.02]$, $g = [0^\circ, -27.4^\circ, -58.6^\circ \text{ y } -68.4^\circ]$).
- 2) Se introdujo un desfase adicional en uno de los elementos del array (añadiendo un tramo de fibra adicional) y se comprobaron los resultados obtenidos con la misma secuencia ($g = [0^\circ, 48.4^\circ, -58.6^\circ \text{ y } -68.4^\circ]$). Al mantener constante la secuencia pero variar el desfase, los resultados esperables son negativos.
- 3) Se procedió a buscar una nueva secuencia que pudiera desplazar los modos deseados ($f_0 + f_M$ en este caso) a sus posiciones originales ($c_n = [0.66 \ 0.69 \ 0.28 \ 0.21]$, $c_{i,n} = [0.31 \ 0.59 \ 0.66 \ 0.83]$).

Los resultados se muestran en las Figuras 24 (paso 1), 25 (paso 2), 26 (paso 3) y 27 (paso 3).

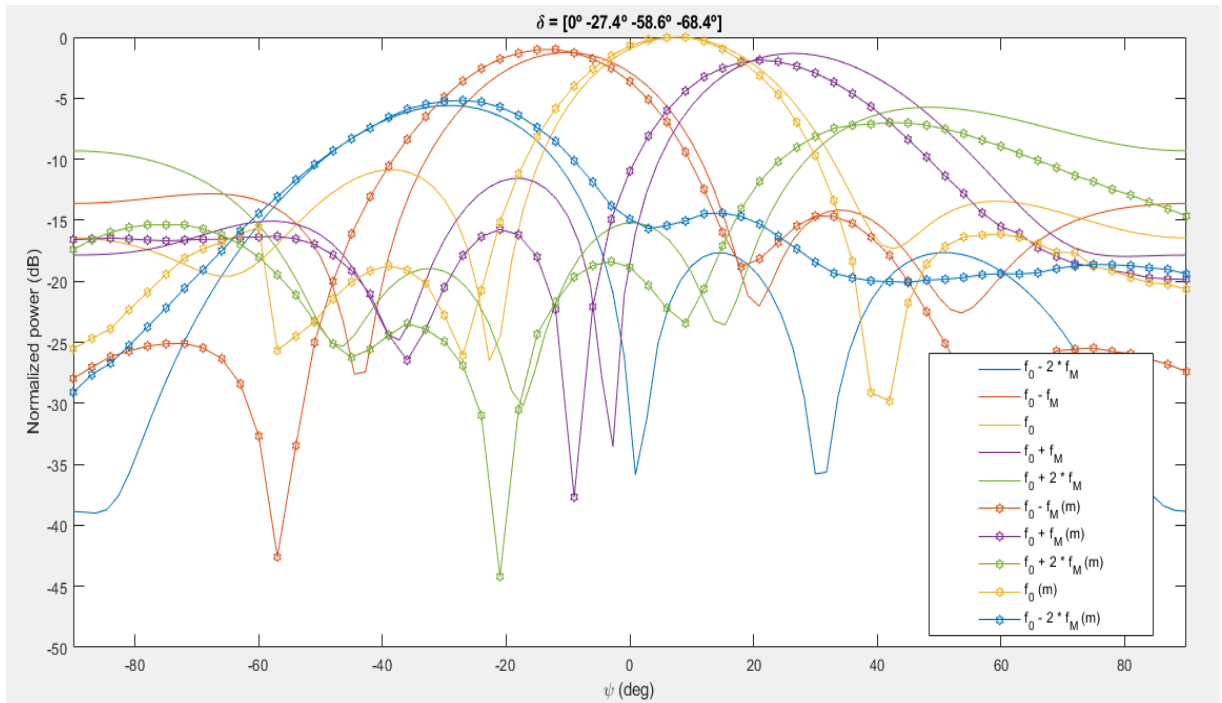


Figura 24. Secuencia óptima sin incluir desfase adicional.

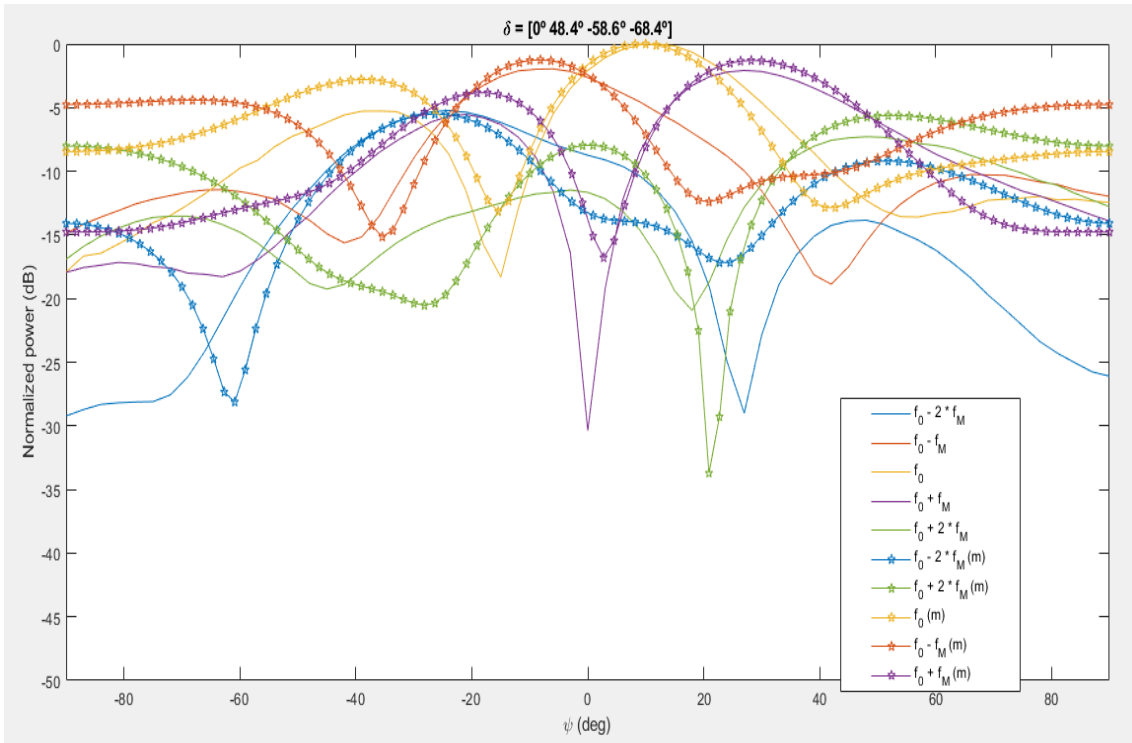


Figura 25. Secuencia óptima original con desfase añadido.

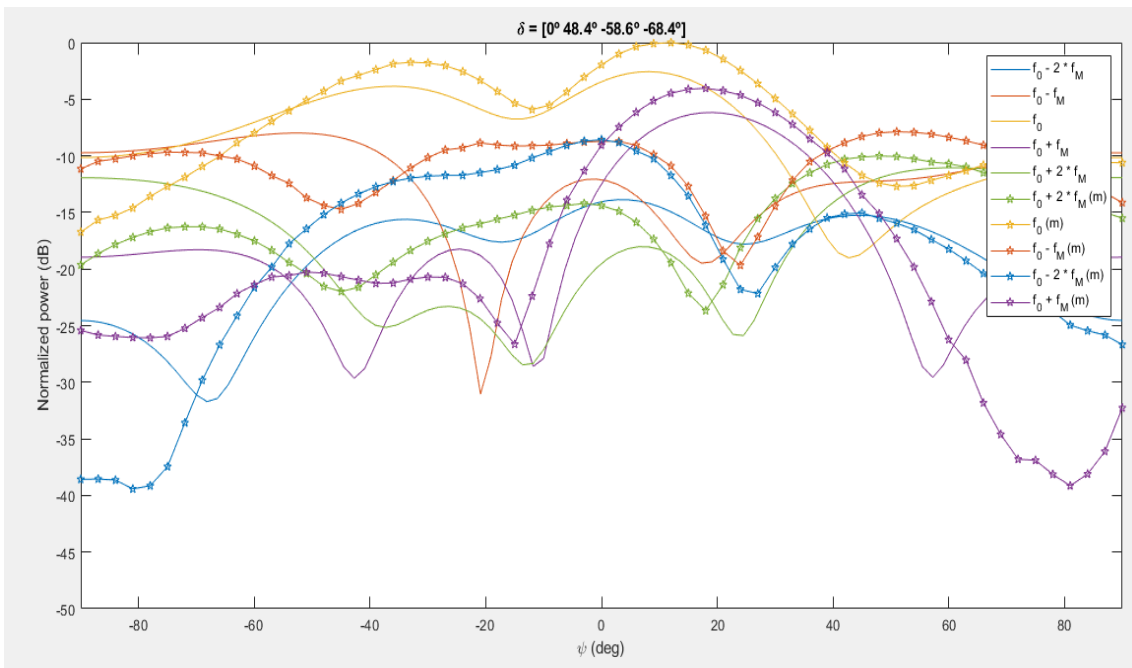


Figura 26. Nueva secuencia óptima para la compensación del desfase.

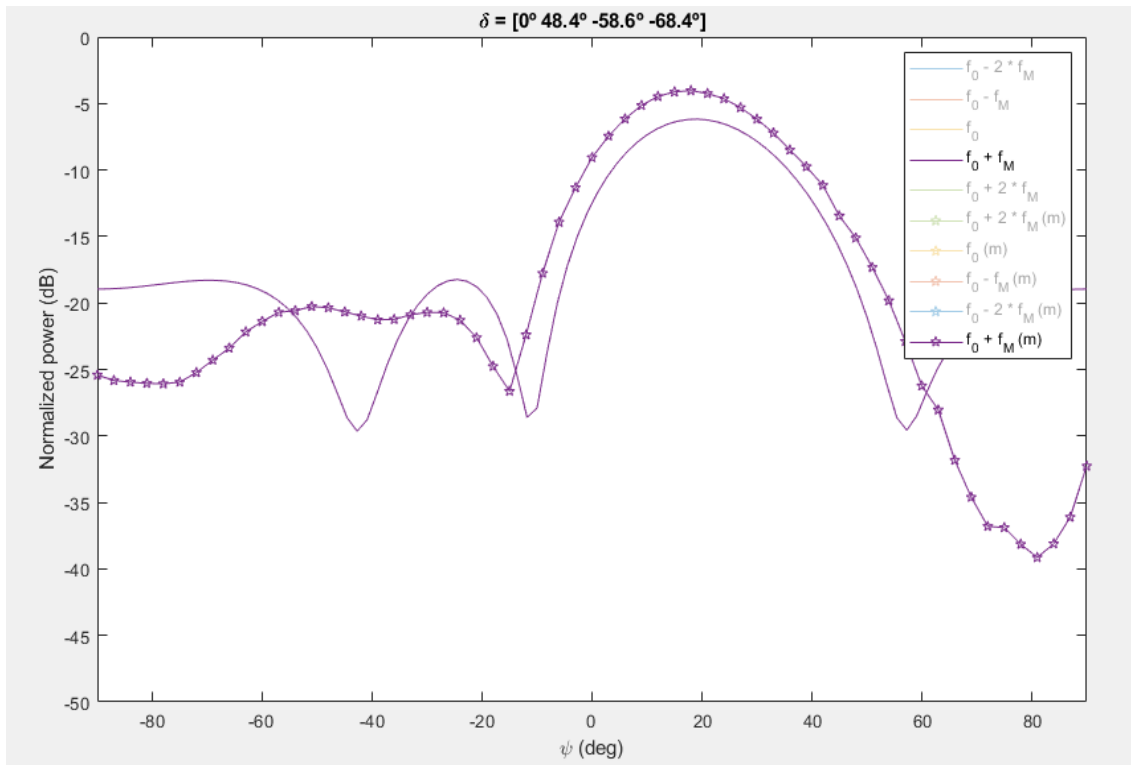


Figura 27. Reconstrucción del modo $f_0 + f_M$.

Cabe destacar que en el ejemplo expuesto únicamente trató de reconstruirse por completo el modo $f_0 + f_M$. Esto es debido a que a priori no parece posible la reconstrucción simultánea de todos los modos. Sin embargo debido a falta de disponibilidad en el laboratorio esto es un tema que deberá conllevar más estudio en el futuro.

5. CONCLUSIÓN Y LÍNEAS FUTURAS

A lo largo del presente trabajo se ha realizado el estudio de diferentes sistemas de RoF basados en TMA. Un primer escenario propuesto fue el caso en el que el array únicamente constaba de $N_a = 2$ antenas. En dicho escenario se buscó compensar el desfase producido por la diferencia de caminos ópticos mediante variaciones en los retardos y ciclos de trabajo de los trenes de pulsos que alimentaban el array. Los resultados concluyeron en que dicho objetivo no era posible ya que no podían compensarse simultáneamente ambos modos secundarios.

El segundo escenario que se analizó constaba de una array de $N_a = 4$ antenas. En este caso el objetivo buscado consistió en el apuntamiento simultáneo del diagrama de radiación del sistema en 5 direcciones diferentes dependiendo de la frecuencia. Además, se buscó explotar el hecho de que variaciones en las secuencias de alimentación del TMA producían variaciones en el diagrama de radiación del sistema. Esto, conocido como '*beam-steering*' fue uno de los logros más importantes obtenidos.

Por último también se busco demostrar el hecho de que es posible compensar de forma exacta los desfases producidos en el caso de $N_a = 4$ antenas. Sin embargo, debido a problemas de disponibilidad de los materiales, no pudo llevarse un estudio exhaustivo de este tema y por ello queda pendiente para futuras investigaciones.

Otras futuras líneas, además de la ya mencionada, serían la optimización de los programas empleados para medir los diagramas de radiación el laboratorio, la implementación en una FPGA de los mecanismos necesarios para generar los trenes de pulsos mejorando así la precisión con respecto del microcontrolador SMT32 y el análisis de sistemas de RoF sobre TMA con un mayor número de elementos en el array.

6. BIBLIOGRAFÍA

- [1] “TIME MODULATED ARRAYS: FROM THEIR ORIGIN TO THEIR UTILIZATION IN WIRELESS COMMUNICATION SYSTEMS”, SENSORS, VOL. 17, NO. 3, PP. 590, ROBERTO MANEIRO-CATOIRA, JULIO BRÉGAINS, JOSÉ A. GARCÍA-NAYA,* AND LUIS CASTEDO, 2017.
- [2] “Beam-Steering Features of Radio-over-Fiber Systems via Antenna Array Time Modulation”, Andrea Giovannini, Jacopo Nanni, Giacomo Paolini, Proceedings of the 2020 International Topical Meeting on Microwave Photonics (MWP), 2020, pp. 140 – 143, Enero 2019.
- [3] “A two-element time-modulated array with direction-finding properties,” A. Tennant and B. Chambers, IEEE Antennas Wireless Propag. Lett., vol. 6, pp. 64–65, 2007.
- [4] “Harmonic Beamforming in Time Modulated Linear Arrays”, Lorenzo Poli, Paolo Rocca, Giacomo Oliveri, Andrea Massa, IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, VOL. 59, NO. 7, JULY 2011.
- [5] “Simultaneous control of sidelobe level and harmonic beam steering in time-modulated linear arrays”, Y. Tong and A. Tennant, IET Electronics Letters, vol. 46, no. 3, pp. 200-202, 2010.

