



Universidad de  
Oviedo



# **ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

## **GRADO EN INGENIERÍA EN TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN**

### **ÁREA DE TEORÍA DE LA SEÑAL Y COMUNICACIONES**

#### **TRABAJO FIN DE GRADO**

#### **AGRUPACIÓN DE ANTENAS BASADA EN RADIO DEFINIDA POR SOFTWARE PARA APLICACIONES DE BEAMFORMING**

**D. SOLARES ARDISANA, DANIEL**

**TUTOR: D. GONZÁLEZ AYESTARÁN, RAFAEL**

**COTUTOR: D. PENDÁS RECONDO, ÁLVARO**

**FECHA: JULIO 2024**

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Contextualización . . . . .	3
1.2. Objetivos . . . . .	4
1.3. Tareas . . . . .	5
1.4. Cronograma . . . . .	5
<b>2. Marco teórico</b>	<b>7</b>
2.1. SDR (Software Defined Radio) . . . . .	7
2.2. Agrupaciones de antenas . . . . .	8
2.2.1. Campo cercano y campo lejano . . . . .	9
2.2.2. Síntesis del campo radiado . . . . .	11
2.3. Procesado de señal . . . . .	13
2.3.1. Correlación cruzada . . . . .	13
2.3.2. Retardo entre dos señales . . . . .	14
<b>3. Desarrollo</b>	<b>15</b>
3.1. Diseño del array . . . . .	15
3.2. Antenas y soportes . . . . .	21
3.3. Conexión física . . . . .	26
3.4. GNU Radio . . . . .	29
3.4.1. Diagrama de bloques . . . . .	29
3.4.2. Bloques personalizados . . . . .	32
3.4.2.1. Bloque <i>Cálculo de desfase</i> . . . . .	32
3.4.2.2. Bloque <i>Cálculo de pesos</i> . . . . .	33
3.4.3. Bloques <i>UHD: USRP</i> . . . . .	35
<b>4. Resultados y mediciones</b>	<b>40</b>



4.1.	Puesta en marcha . . . . .	40
4.1.1.	Sincronización de los USRP . . . . .	40
4.1.2.	Conformado de haz . . . . .	43
4.2.	Medidas . . . . .	44
4.2.1.	Sincronización frente a apuntado. . . . .	45
4.2.2.	Enfoque en zona cercana al array . . . . .	47
4.2.3.	Cambios del ángulo de apuntado . . . . .	51
4.2.4.	Array de parches . . . . .	53
<b>5.</b>	<b>Conclusiones y trabajo futuro</b>	<b>59</b>
<b>A.</b>	<b>Código fuente de los bloques personalizados</b>	<b>63</b>
A.1.	Bloque Cálculo de desfase . . . . .	63
A.2.	Bloque Conformador pesos . . . . .	66

# 1.- Introducción

## 1.1.- Contextualización

La radio definida por software (SDR, por sus siglas en inglés) es un concepto utilizado para referirse a un amplio conjunto de sistemas. Una de las posibles definiciones, según el IEEE, es 'Un tipo de radio en la cual algunas o todas las funciones de la capa física están definidas por software.' [1].

Es decir, un sistema SDR es aquel en que las funciones que tradicionalmente se implementan mediante hardware específico, como puede ser la modulación, demodulación, filtrado y/o procesado de señales, se realizan mediante software. Esto dota al sistema de una gran flexibilidad, permitiendo que un mismo hardware pueda adaptarse a diferentes estándares y protocolos de comunicación mediante una simple actualización de software. Esto es especialmente valioso en entornos donde los estándares de comunicación evolucionan rápidamente, y ha hecho que, desde su aparición en la década de los 90 [2], esta tecnología haya sido ampliamente adoptada en todo tipo de aplicaciones militares, comerciales y de investigación.

Esta flexibilidad permite desarrollar todo tipo de sistemas, entre los que se encuentran los de enfoque y conformado de haz, que concentran la energía radiada en puntos o direcciones específicas del espacio [3]. Estos sistemas facilitan la transferencia inalámbrica de energía (WPT, por sus siglas en inglés), posibilitando la alimentación o carga a distancia de dispositivos electrónicos. Tiene aplicaciones en varios campos, desde la carga de teléfonos móviles hasta la alimentación de implantes médicos y sensores remotos.

Estos sistemas de enfoque se basan en un concepto conocido como arrays, un conjunto de múltiples elementos radiantes (antenas individuales), dispuestos en una configuración espacial específica. Estos elementos, además, han de ser alimentados de una forma concreta para llevar a cabo este enfoque. Esto plantea el principal problema al que se enfrenta este trabajo: la sincronización de los USRP de forma que se garantice la alimentación uniforme de cada elemento del array. Es decir que las señales que llegan a cada antena deben estar en

fase. Posteriormente, aplicar un desfase a cada una de esas señales permitirá el enfoque de la energía en una dirección concreta.

## 1.2.- Objetivos

El principal objetivo de este trabajo es la implementación de un sistema que permita el conformado de haz mediante el uso de dispositivos SDR, concretamente con el modelo USRP B210, de la marca Ettus Research.

Si bien los dispositivos utilizados sí son capaces de sincronizarse mediante una referencia externa de forma que las muestras se tomen en el mismo punto de la señal, la fase de estas señales no será la misma. Por ello, uno de los objetivos más importantes es el desarrollo de un programa que permita esta sincronización, de forma que todas las antenas alimentadas compartan una misma referencia de fase.

En una fase distinta a la de sincronización, este mismo sistema se encargará de realizar los cálculos necesarios para el enfoque. Para ello, el usuario indicará una dirección en el espacio y el programa ajustará los pesos de los elementos del array para concentrar la energía en dicha dirección.

Además, para confirmar el correcto funcionamiento del sistema, se llevará a cabo una simulación basada en las expresiones teóricas. Posteriormente, se compararán estos resultados con los obtenidos en la práctica con un demostrador, permitiendo evaluar el rendimiento real del conjunto.

El trabajo también permite el planteamiento de ciertos objetivos a nivel académico, pues requiere la adquisición de conocimientos a nivel personal. La correcta implementación de todo lo mencionado supone formarse en el campo de los dispositivos de radio definida por software, así como con la herramienta GNU Radio, que no están incluidos en las competencias del grado. También supone una profundización en la teoría de arrays con respecto a lo aprendido en los estudios universitarios, concretamente en su comportamiento en la región de campo cercano.

### 1.3.- Tareas

Para llevar a cabo el proyecto, se proponen las siguientes tareas:

1. Estudio de la teoría de agrupaciones de antenas y su aplicación a conformado de haz.
2. Aprendizaje del uso de la herramienta GNU Radio.
3. Implementación de un programa que permita la sincronización en fase de los dispositivos USRP.
4. Implementación de un programa que permita el ajuste de los pesos del array.
5. Simulación de un array de antenas omnidireccionales y comparación de los resultados con los obtenidos en la práctica.
6. Realización de pruebas y análisis de resultados.

### 1.4.- Cronograma





## 2.- Marco teórico

En este capítulo se describen los conceptos teóricos necesarios para el desarrollo del trabajo. Se comienza con una introducción a la radio definida por software, seguido de una descripción de las agrupaciones de antenas y su aplicación al conformado de haz. Posteriormente, se describen las herramientas de procesamiento de señal necesarias para la sincronización de los dispositivos USRP.

### 2.1.- SDR (Software Defined Radio)

La radio definida por software es un sistema en el que funciones como el filtrado o la mezcla de señales, que normalmente se realizan mediante hardware específico, son implementadas vía software. Esto dota de una gran flexibilidad al proceso de implementación de sistemas de comunicaciones, eliminando la necesidad de componentes específicos para cada caso [4].

La arquitectura de un sistema SDR se divide típicamente en dos partes: la sección de RF (Radio Frecuencia) y la sección de procesamiento digital. La primera gestiona la transmisión y recepción de señales, y la segunda se conecta a esta mediante conversores de señal analógica a digital (A/D). En esta etapa, suele haber un FPGA que realiza tareas de procesamiento de señales en tiempo real y un procesador (CPU) que lo controla. En la figura 2.1 se muestra una arquitectura genérica de un sistema SDR.

En la figura 2.2 se muestra una imagen del modelo USRP B210 utilizado en este trabajo. En contraste con la arquitectura ilustrada en la figura 1, este cuenta con 4 puertos distribuidos en dos canales distintos. En cada canal, uno de los puertos es capaz de transmisión y recepción, y el otro solo de recepción. Todos ellos son coherentes entre sí [5], ofreciendo MIMO 2x2. Esta característica simplificará, en parte, la tarea de sincronización, ya que las dos antenas conectadas a cada puerto comparten la misma fase de alimentación.

La configuración de estos dispositivos se puede realizar de distintas formas. En el caso de este trabajo, se utilizará GNU Radio, una herramienta de código abierto que facilita esta

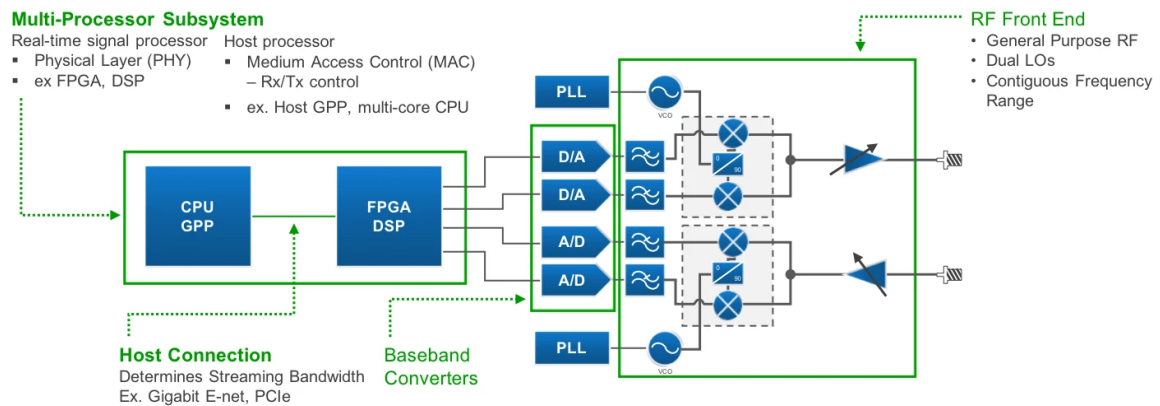


Figura 2.1: Arquitectura genérica de un sistema SDR. Fuente: <https://www.ni.com/en/shop/wireless-design-test/what-is-a-usrp-software-defined-radio/what-is-ni-usrp-hardware-.html>



Figura 2.2: USRP B210. Fuente: <https://www.ettus.com/all-products/usrp-b200-enclosure/>

tarea mediante diagramas de bloques.

## 2.2.- Agrupaciones de antenas

Una agrupación de antenas, conocida como array, consiste en la combinación de múltiples antenas que, alimentadas de una manera específica, pueden realizar diversas funciones. En este trabajo, se utilizarán para concentrar la energía en un punto específico del espacio. No obstante, también tienen aplicaciones en recepción, como puede ser la radiogoniometría, que consiste en determinar la dirección de llegada de ondas electromagnéticas a partir de los desfases medidos en cada uno de los elementos del array.

Si bien este trabajo se centra en el caso de un array lineal, cabe mencionar que, según su geometría, estos pueden ser bidimensionales o incluso tridimensionales. Finalmente, debido

a las dimensiones de las antenas y la frecuencia utilizada, deberá tenerse en cuenta que el array funcionará en campo cercano.

### 2.2.1.- Campo cercano y campo lejano

En el contexto de la radiación electromagnética, se distingue entre el campo cercano y el campo lejano de una antena.

■ **Campo cercano:** se subdivide en:

- **Campo cercano reactivo:** es la región inmediatamente alrededor de la antena, donde predominan los campos reactivos. Es una zona pequeña que depende del tamaño de la antena y de la longitud de onda. En esta región, las componentes del campo eléctrico y magnético no se comportan como ondas planas.
- **Campo cercano radiante (región de Fresnel):** esta región se encuentra entre el campo cercano reactivo y el campo lejano. Aquí predominan los campos radiantes y la distribución del campo depende de la distancia a la antena. La forma del patrón de radiación es una función de la distancia radial, y la componente radial del campo puede no ser despreciable.

- **Campo lejano (región de Fraunhofer):** en esta región, la distribución angular del campo es independiente de la distancia a la antena. Los campos eléctricos y magnéticos se comportan como ondas planas, y las componentes del campo son transversales. Esta región comienza a aproximadamente una distancia  $2D^2/\lambda$  de la antena, con  $D$  la máxima dimensión de la antena y  $\lambda$  la longitud de onda.

Para ilustrar la diferencia entre ambos campos, y como afectan a los cálculos del campo radiado por un conjunto de antenas, se puede considerar el caso de dos elementos como el mostrado en la figura 2.3 [6]. Para el caso de campo cercano, el campo eléctrico radiado por el conjunto se puede expresar como:

$$\vec{E}(\vec{r}) = \frac{E_1(\theta_1)e^{-j2\pi|\vec{r}-\vec{r}'_1|/\lambda}}{|\vec{r}-\vec{r}'_1|} + \frac{E_2(\theta_2)e^{-j2\pi|\vec{r}-\vec{r}'_2|/\lambda}}{|\vec{r}-\vec{r}'_2|} \quad (2.1)$$



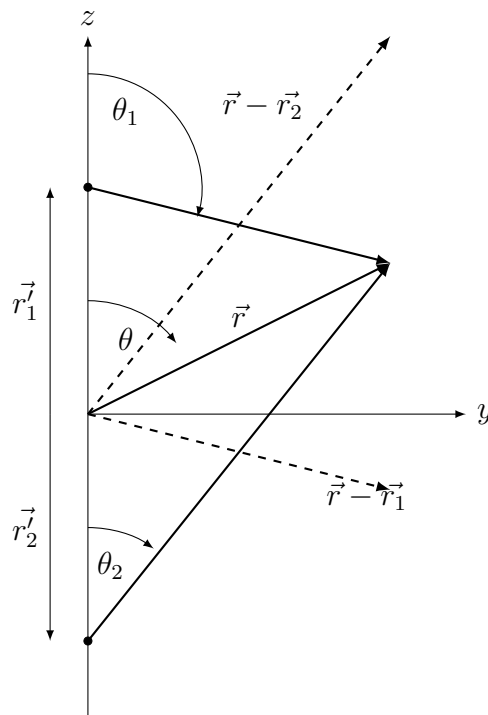


Figura 2.3: Array de dos elementos en campo cercano.

Si se supone el caso de un array de  $N$  elementos, la ecuación, para el caso en el que todas las antenas están alimentadas de forma uniforme, se generaliza a la ec. 2.2.

$$\vec{E}(\vec{r}) = \sum_{n=1}^N \frac{E_n(\theta_n, \phi_n) e^{-j2\pi|\vec{r}-\vec{r}'_n|/\lambda}}{|\vec{r}-\vec{r}'_n|} \quad (2.2)$$

Donde  $\vec{r}$  es el vector posición,  $\theta$  y  $\phi$  son los ángulos de elevación y azimut, respectivamente,  $\vec{r}'_n$  es la posición de la  $n$ -ésima antena y  $\lambda$  es la longitud de onda en espacio libre. El término exponencial representa el desfase en las ondas emitidas causado por la diferencia de posición de cada uno de los elementos.

En el caso de campo lejano, la distancia al punto de interés  $\vec{r}$  será mucho mayor que el vector que representa la posición de cada una de las antenas  $\vec{r}'_n$ . Por tanto, se puede aproximar que la señal radiada por cada antena se propaga de forma paralela, como se muestra en la figura 2.4. Para este caso, se pueden hacer las siguientes aproximaciones para el campo eléctrico:

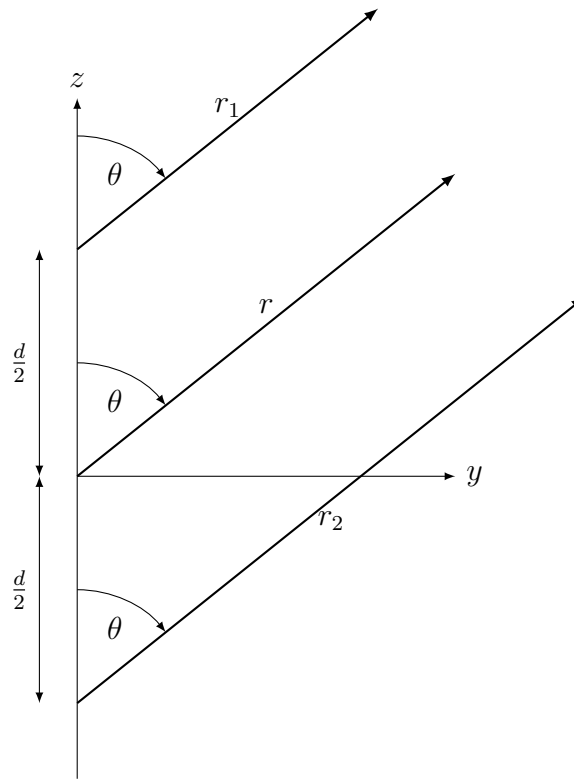


Figura 2.4: Array de dos elementos en campo lejano.

- $r_1 \approx r \approx r_2$  para la amplitud.
- $r_1 \approx r - \frac{d}{2}\cos(\theta)$  y  $r_2 \approx r + \frac{d}{2}\cos(\theta)$  para la fase.

Sin embargo, estas aproximaciones no serán válidas durante el desarrollo de este trabajo, pues las dimensiones del array serán del orden de la distancia a la que se realizarán las medidas.

### 2.2.2.- Síntesis del campo radiado

A la vista de la expresión 2.2, la suma de todas las contribuciones en el punto de interés  $\vec{r}$  puede ser constructiva o destructiva, dependiendo de la posición de los elementos y la longitud de onda.

Si se aplica un peso  $w_n$  a cada antena del array, el campo eléctrico radiado por la agru-

pación en un punto  $\vec{r}$  se puede expresar como:

$$\vec{E}(\vec{r}) = \sum_{n=1}^N \frac{w_n E_n(\theta_n, \phi_n) e^{-j2\pi|\vec{r}-\vec{r}'_n|/\lambda}}{|\vec{r}-\vec{r}'_n|} \quad (2.3)$$

El objetivo de estos valores es ajustar la fase de la señal (y la amplitud, dependiendo del método utilizado) emitida por cada antena, de manera que la suma del campo eléctrico radiado por cada una de las antenas produzca una interferencia constructiva en el punto de interés (punto de enfoque).

Cada valor  $w_n$  se puede obtener utilizando distintos métodos. En este caso, se utilizará el método de fase conjugada (Conjugate-Phase method). En este método, los pesos se ajustan únicamente para compensar el desfase por la diferencia de posición de las antenas. Por tanto, el peso a aplicar al  $n$ -ésimo elemento viene dado por:

$$w_n = e^{j2\pi|\vec{r}-\vec{r}'_n|/\lambda} \quad (2.4)$$

Si se sustituye este valor en la ecuación 2.3, se obtiene que el campo eléctrico radiado por la agrupación en un punto  $\vec{r}$  es:

$$\vec{E}(\vec{r}) = \sum_{n=1}^N \frac{E_n(\theta_n, \phi_n) e^{-j2\pi|\vec{r}-\vec{r}'_n|/\lambda} e^{j2\pi|\vec{r}-\vec{r}'_n|/\lambda}}{|\vec{r}-\vec{r}'_n|} = \sum_{n=1}^N \frac{E_n(\theta_n, \phi_n)}{|\vec{r}-\vec{r}'_n|} \quad (2.5)$$

Si además se usan antenas omnidireccionales en el plano  $XY$ , es decir, que el campo eléctrico radiado por cada una de ellas es independiente de  $\phi$ , y se mantiene el ángulo  $\theta$  constante, se tiene que la ecuación 2.5 se puede simplificar, a:

$$\vec{E}(\vec{r}) = \sum_{n=1}^N \frac{E_n}{|\vec{r}-\vec{r}'_n|} \quad (2.6)$$

Por tanto, en el punto de interés  $\vec{r}$ , el campo eléctrico estará compuesto por la suma de los campos eléctricos de cada una de las antenas. La ecuación 2.6 permite también predecir la potencia recibida en el punto de interés. Siendo  $E_n$  el valor cuadrático medio (RMS) de campo radiado por la  $n$ -ésima antena, y suponiendo que  $r'_n$  es despreciable frente a  $\vec{r}$ , se tiene

que:

$$E(\vec{r}) = \sum_{n=1}^N \frac{E_n}{|\vec{r} - \vec{r}'_n|} \approx \sum_{n=1}^N \frac{E_n}{|\vec{r}|} = N \frac{E_n}{|\vec{r}|} \Rightarrow P(\vec{r}) = \frac{(NE_n)^2}{\eta_0 |\vec{r}|^2} = N^2 \frac{E_n^2}{\eta_0 |\vec{r}|^2} \quad (2.7)$$

Donde  $P(\vec{r})$  es la potencia recibida en el punto de interés, y  $\eta_0$  es la impedancia característica del medio. Si se utilizan unidades logarítmicas, la potencia recibida será:

$$P(\vec{r}) = 10 \log_{10} \left( N^2 \frac{E_n^2}{\eta_0 |\vec{r}|^2} \right) = 10 \log_{10} \left( \frac{E_n^2}{\eta_0 |\vec{r}|^2} \right) + 20 \log_{10} (N) \quad (2.8)$$

Es decir, la potencia recibida en el punto  $\vec{r}$  con un array de  $N$  elementos sincronizados y con los pesos ajustados estará  $20 \log_{10} (N)$  decibelios por encima del caso en el que únicamente se utiliza una antena.

## 2.3.- Procesado de señal

Para la sincronización en fase de las señales provenientes de cada uno de los dispositivos USRP, se requiere de ciertas herramientas de procesado de señal, descritas a continuación.

### 2.3.1.- Correlación cruzada

La correlación cruzada es una operación matemática que se utiliza para determinar el grado de similitud entre dos señales. Supóngase el caso de dos señales continuas  $f(t)$  e  $g(t)$ . En este caso, la correlación cruzada se define como:

$$R_{fg}(\tau) = \int_{-\infty}^{\infty} f(t)g(t + \tau)dt \quad (2.9)$$

No obstante, el caso de interés en este trabajo es el de señales discretas. En ese caso, la correlación cruzada se define como:

$$R_{fg}[m] = \sum_{n=-\infty}^{\infty} f[n]g[n + m] \quad (2.10)$$

Si bien esta es la expresión general, en la práctica las señales no tendrán un soporte infinito. Además, las señales que se sincronizarán serán de igual periodo discreto  $N$ . Por tanto, la expresión de la correlación cruzada se simplifica a:

$$R_{fg}[m] = \frac{1}{N} \sum_{n=0}^{N-1} f[n]g[n+m] \quad (2.11)$$

### 2.3.2.- Retardo entre dos señales

A partir de la ecuación 2.11 se puede obtener el retardo entre dos señales. Supóngase el caso donde  $g[n] = f[n+k]$ , con  $k$  un número entero. En este caso, la correlación cruzada se simplifica a:

$$R_{fg}[m] = \frac{1}{N} \sum_{n=0}^{N-1} f[n]f[n+m+k] = R_{ff}[m+k] \quad (2.12)$$

Ha de tenerse en cuenta que el máximo de la función  $R_{ff}[m]$ , por tratarse de la autocorrelación, se halla en  $m = 0$ . Por tanto, si se calcula la correlación cruzada entre dos señales que son idénticas, pero con una de ellas retrasada con respecto a la otra y se obtiene un máximo en  $m = -k$ , se puede afirmar que la señal  $g[n]$  está retrasada  $k$  muestras respecto a la señal  $f[n]$ . Además, se podrá corregir este retardo retrasando la señal  $g[n]$  por un valor  $m$  muestras:

$$g[n] = f[n+k] \Rightarrow g[n+m] = g[n-k] = f[n-k+k] = f[n] \quad (2.13)$$

Este será el principio aplicado en GNU Radio para que las señales transmitidas estén sincronizadas.

## 3.- Desarrollo

En este capítulo se detallan las tareas llevadas a cabo en el desarrollo del proyecto. Se comienza con el diseño del array, seguido de la conexión física de los dispositivos y el funcionamiento del programa en GNU Radio.

### 3.1.- Diseño del array

En primer lugar, se ha realizado un diseño teórico del array de antenas, suponiendo para ello elementos isotrópicos. Esto permite obtener una primera idea de como se comportará el sistema en la práctica, si bien las curvas aquí representadas se verán en la realidad afectadas por los correspondientes diagramas de radiación. Para esta sección y las siguientes, se han supuesto los elementos dispuestos a lo largo del eje X para el caso de un array lineal, y el plano XY para el caso de los arrays bidimensionales. La dirección de propagación será la dirección Z.

El primer diseño propuesto es un array lineal de elementos equiespaciados, con una separación de  $0.7\lambda$ . La frecuencia de trabajo será de 2.4 GHz. Esta frecuencia, además de ser adecuada para las antenas que se utilizarán, corresponde a una separación física entre elementos que hace que el tamaño del array sea razonable:

$$d = 0.7\lambda = 0.7 \frac{c}{f} = 0.7 \frac{3 \cdot 10^8}{2.4 \cdot 10^9} = 0.0875m = 8.75cm \quad (3.1)$$

Además, se trata de una banda de frecuencias ISM (banda para el área industrial, científica y médica), que no requiere licencia para su uso. Con respecto a los SDR utilizados, estos son capaces de trabajar en un rango de frecuencias de 70 MHz a 6 GHz, por lo que la elección de esta frecuencia también es adecuada en este sentido.

Con esta separación entre elementos el diseño se ha llevado a cabo con ayuda de la herramienta *MATLAB*, y haciendo uso de los pesos correspondientes al método de la fase conjugada, tal y como se describe en la sección 2.2. Se ha dispuesto el array a lo largo del

eje X y centrado en el origen y se han representado las curvas correspondientes a caídas de 3 dB para el plano  $y = 0$ , y para el plano  $z = z_{objetivo}$ . La normalización se realiza con respecto al nivel de potencia máximo en el plano correspondiente. Puesto que en la mayoría de casos este máximo estará en un punto muy cercano a las antenas, se obvian los valores para distancias inferiores a  $1.5\lambda$  en la dirección de propagación.

En las figuras 3.1 y 3.2 se muestra el caso en que el array se alimenta de forma uniforme, sin aplicar pesos. En el plano XZ se puede ver que no se produce un enfoque en ningún punto determinado. En los siguientes casos, esta distribución de energía se comparará con la obtenida al aplicar los pesos correspondientes.

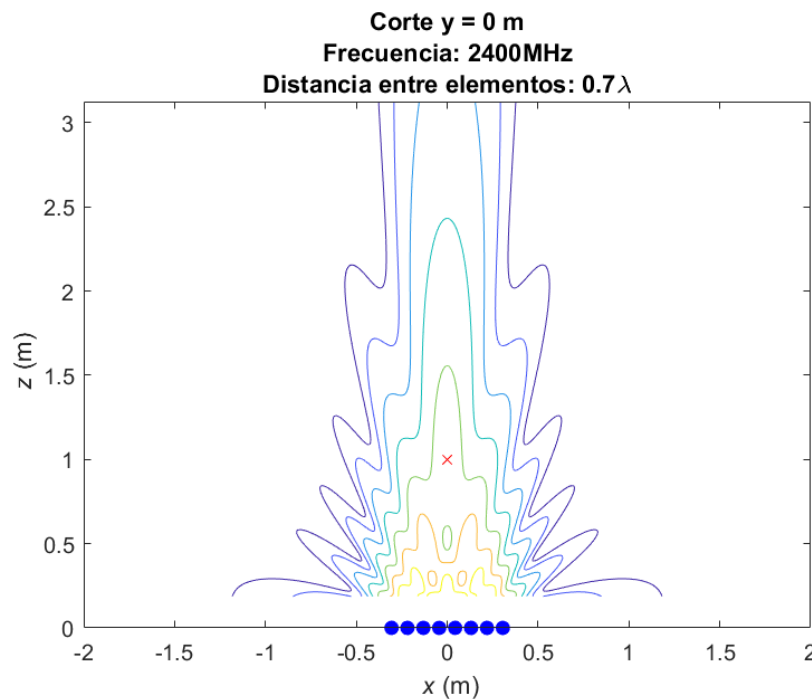


Figura 3.1: Campo eléctrico en plano XZ para alimentación uniforme.

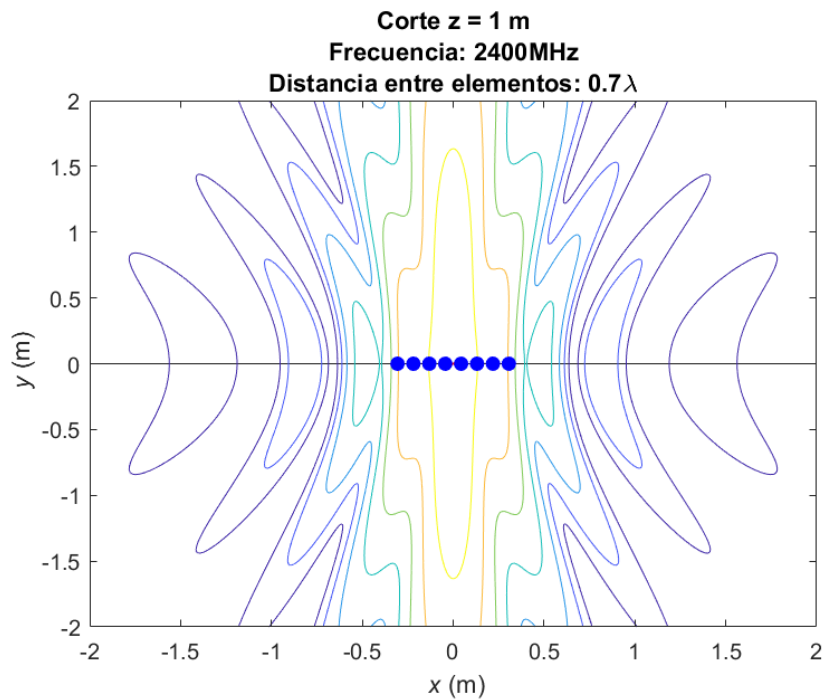


Figura 3.2: Campo eléctrico en plano XY para alimentación uniforme.

A continuación, se ha simulado el caso en que se aplican los pesos correspondientes al punto de enfoque  $(x, y, z) = (0, 0, 1)$ , correspondiente a las figuras 3.3, 3.4. Se aprecia que hay un lóbulo con centro entorno a  $z = 0.6m$  que en el caso anterior no estaba presente. Si bien esta distancia no es la correcta (debería ser  $z = 1m$ ), sí se aprecia un claro cambio en la distribución de la energía con respecto al caso sin enfoque. La diferencia en la posición del lóbulo con respecto al caso ideal es un efecto común en arrays de antenas, conocido como 'focal shift', o desplazamiento del foco [7].



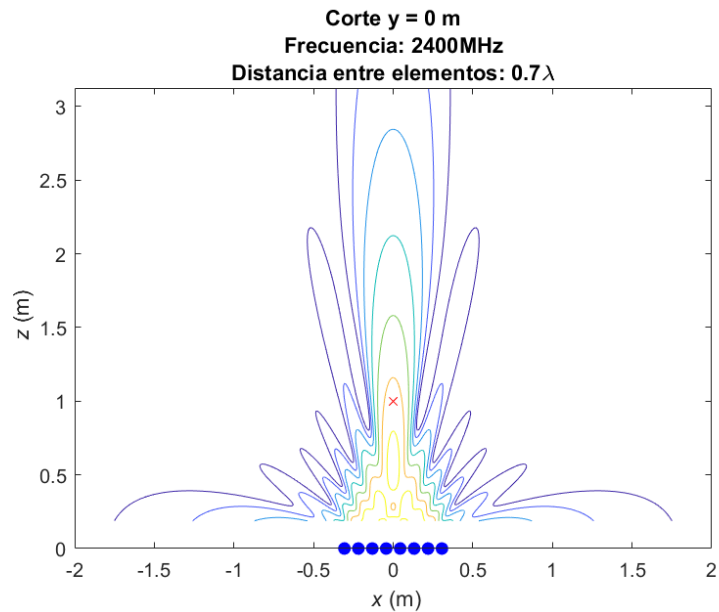


Figura 3.3: Simulación campo eléctrico en plano XZ (Enfoque a  $(0, 0, 1)$  ).

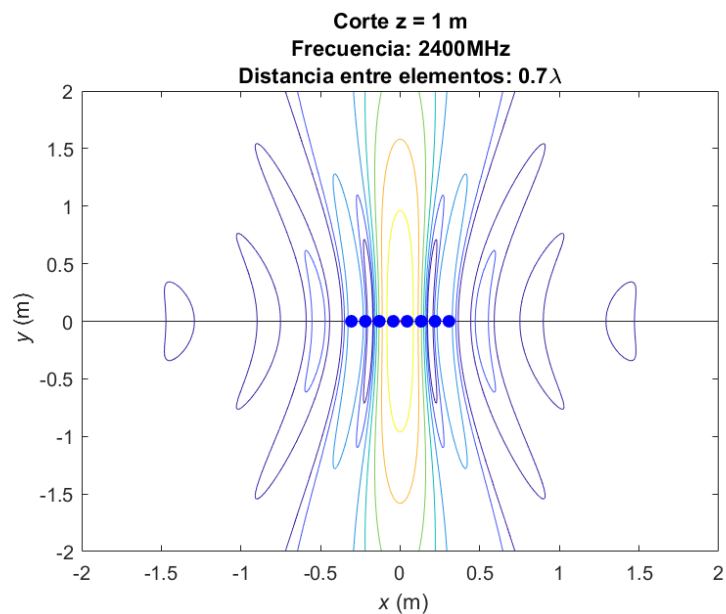


Figura 3.4: Simulación campo eléctrico en plano XY (Enfoque a  $(0, 0, 1)$  ).

El segundo punto de enfoque es  $(\text{sen}(\frac{\pi}{6}), 0, \text{cos}(\frac{\pi}{6}))$ , asociado a las figuras 3.5 y 3.6. Este corresponde a un ángulo de 30 grados con respecto a la normal del array en el plano XZ. Idealmente, el lóbulo mencionado anteriormente debería desplazarse a dicha posición,

manteniendo la distancia al centro del array constante. En las figuras, se aprecia que efectivamente el lóbulo se ha desplazado los 30 grados correspondientes, y se ha hecho ligeramente más pequeño.

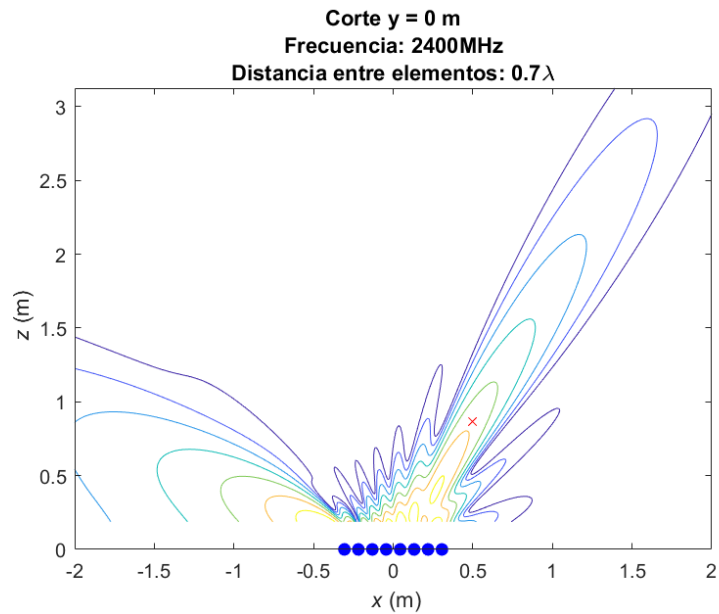


Figura 3.5: Simulación campo eléctrico en plano XZ (Enfoque a  $(\sin(\frac{\pi}{6}), 0, \cos(\frac{\pi}{6}))$ ).

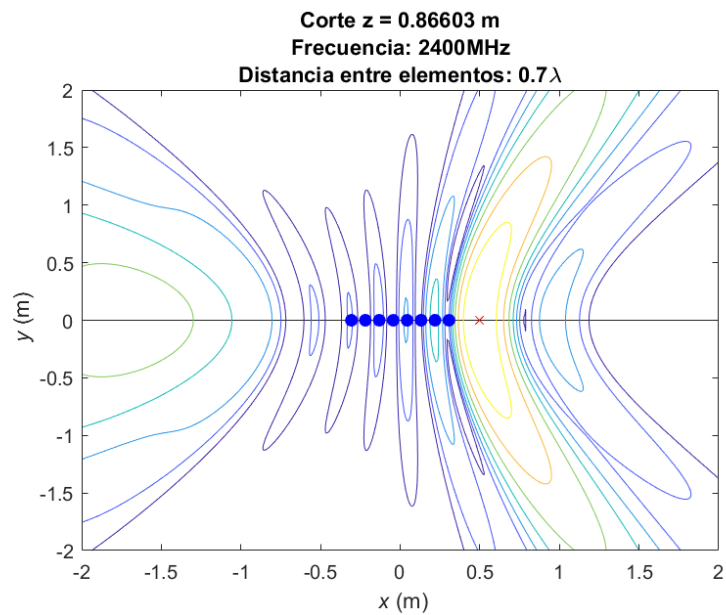


Figura 3.6: Simulación campo eléctrico en plano XY (Enfoque a  $(\sin(\frac{\pi}{6}), 0, \cos(\frac{\pi}{6}))$ ).

Como segundo array, se utilizará también un conjunto de 8 elementos, pero distribuidos en 2 filas de 4 elementos cada una, siendo estas filas paralelas al eje X. La separación entre elementos será la misma que en el caso anterior. En este caso, se ha simulado el enfoque al punto  $(0, 0, 1)$ , como se muestra en las figuras 3.7 y 3.8. Se aprecia que el lóbulo que en el caso lineal se distinguía claramente entorno a  $z = 0.6m$  ha desaparecido. Sin embargo, en el plano  $z = 1m$  se aprecia como el nuevo grado de libertad, resultado de la dimensión Y del array, hace que el lóbulo sea más pequeño.

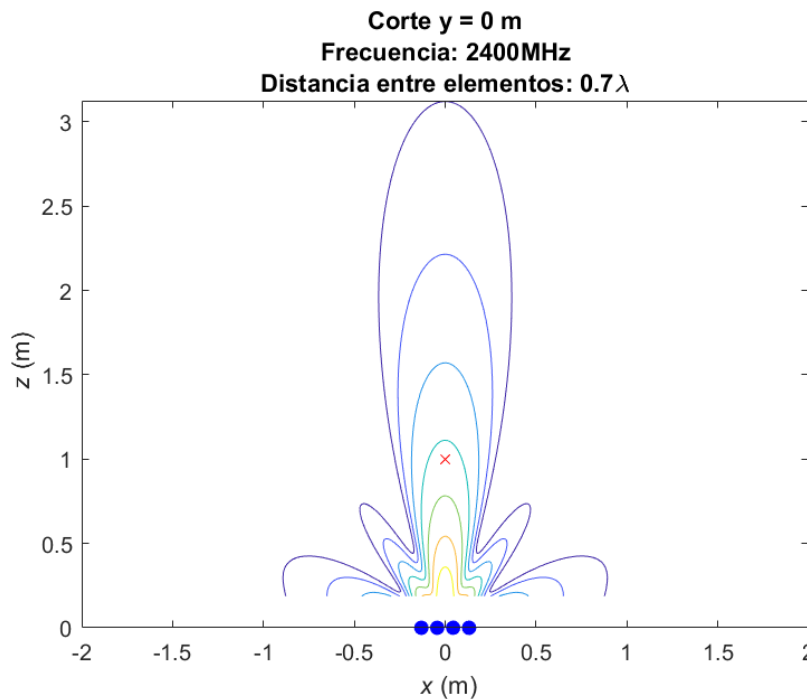


Figura 3.7: Simulación campo eléctrico en plano XZ (Enfoque a  $(0, 0, 1)$  en array 2D).

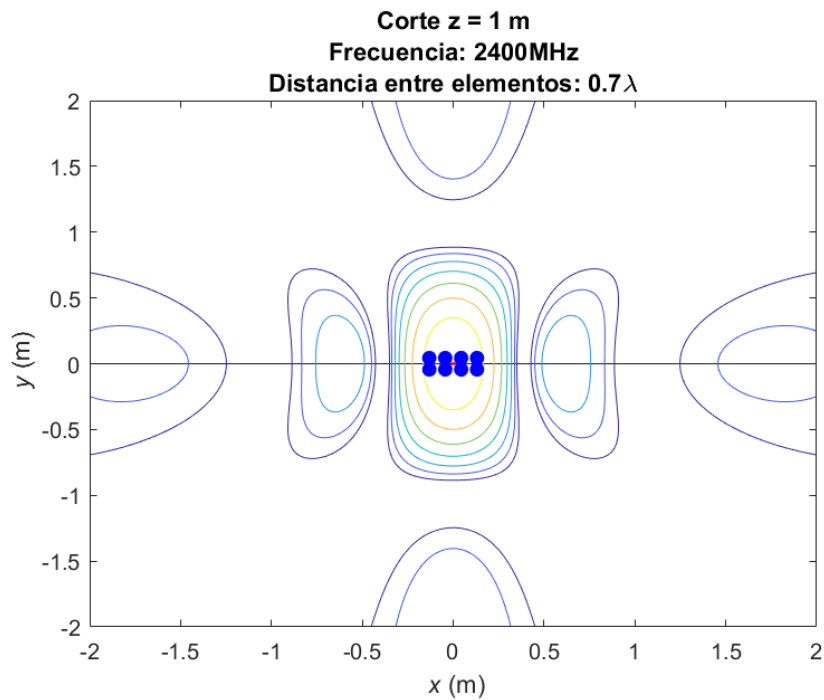


Figura 3.8: Simulación campo eléctrico en plano XY (Enfoque a  $(0, 0, 1)$  en array 2D).

La distribución bidimensional del array hace que este no posea simetría de revolución, lo que permite apuntado en el plano XY paralelo al array. Para ilustrar esto, se elige como punto de enfoque  $(0.4330, 0.2500, 0.8660)$ , que se corresponde con las coordenadas esféricas  $(r, \theta, \phi) = (1, \frac{\pi}{6}, \frac{\pi}{6})$ . Las figuras 3.9 y 3.10 muestran el resultado.

En la figura 3.10, se puede ver como la nueva dimensión del array hace que la curva de caída a  $-3\text{dB}$  no sea simétrica con respecto al eje Y, si no que se concentra entorno a la dirección deseada. Si se tratase de un array lineal, esto no ocurriría por la simetría de revolución. En resumen, se ha comprobado que el diseño teórico del array es correcto, y que los pesos aplicados en el programa deberían permitir el enfoque en los puntos deseados.

### 3.2.- Antenas y soportes

Puesto que se han supuesto elementos isotrópicos para el diseño, la mayoría de las medidas se realizarán con monopolos, pues en el plano perpendicular a estos, el campo radiado es, en teoría, omnidireccional. En la figura 3.11 se muestra una de estas antenas. No obs-

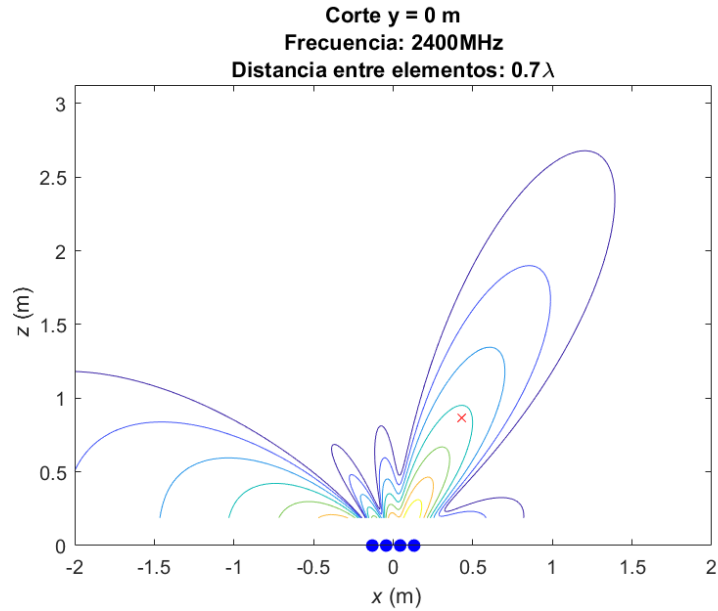


Figura 3.9: Simulación campo eléctrico en plano XZ (Enfoque a  $(0.4330, 0.2500, 0.8660)$  en array 2D).

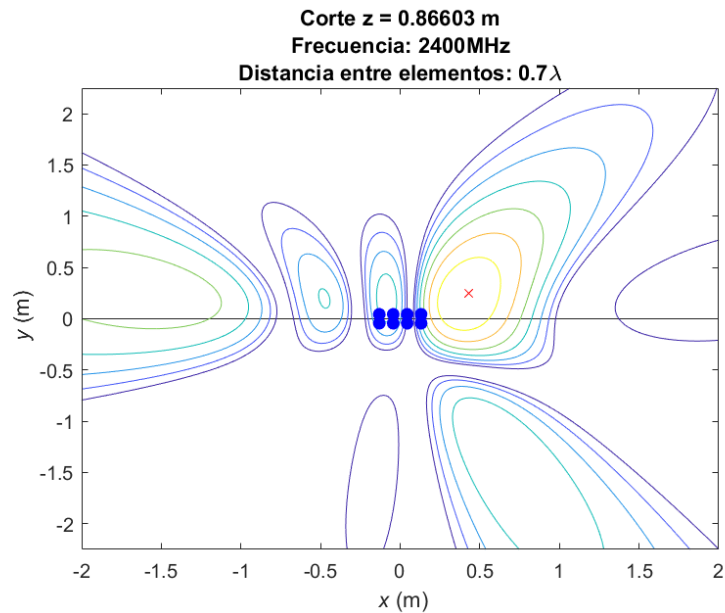


Figura 3.10: Simulación campo eléctrico en plano XY (Enfoque a  $(0.4330, 0.2500, 0.8660)$  en array 2D).

tante, para el caso del array bidimensional se han utilizado parches, como el mostrado en la figura 3.12, pues facilitan la disposición en una configuración bidimensional. Estos parches han sido diseñados y fabricados en el Área de Teoría de la Señal y Comunicaciones de la EPI Gijón, por Luis Fernando Herrán Ontañón.



Figura 3.11: Antena de tipo monopolo utilizada. Modelo: ANT-4GBLADE220-SMA.

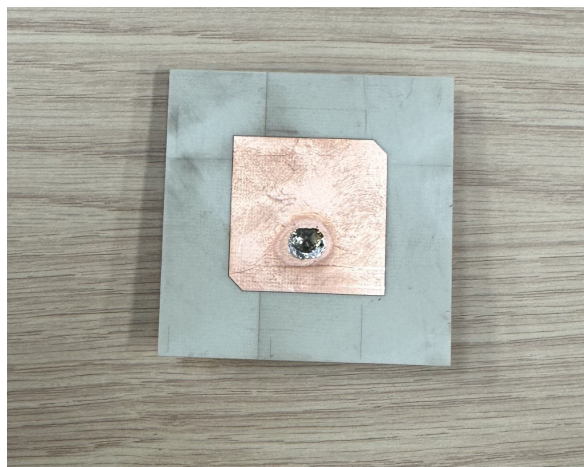


Figura 3.12: Antena de tipo parche utilizada.

Como se menciona en 3.1, uno de los aspectos más importantes de un array es la distancia

entre elementos. Por ello, se han diseñado soportes que permiten la alineación de las antenas y la fijación de las mismas a una determinada distancia. Se ha utilizado el programa de diseño Autodesk Fusion 360. En las figuras 3.13 y 3.14 se muestra el diseño de los soportes.

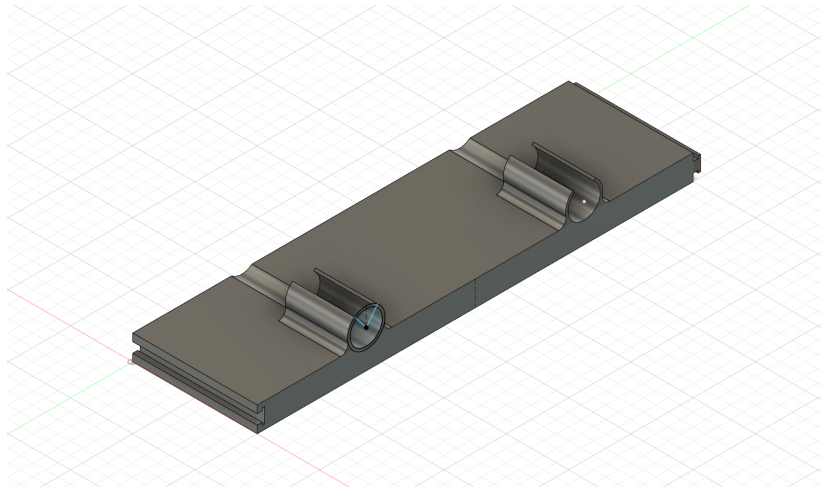


Figura 3.13: Diseño del soporte para las antenas tipo monopolo.

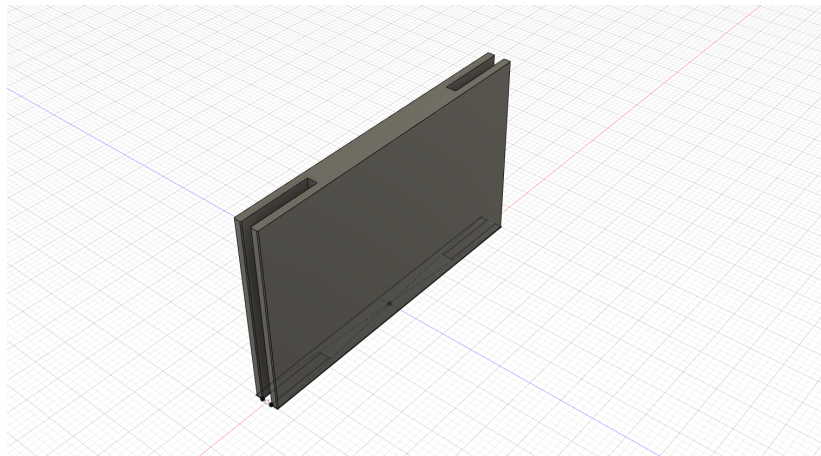


Figura 3.14: Diseño del soporte para las antenas tipo parche.

Para el caso del soporte de monopolos, este cuenta con dos ranuras circulares, en las que se insertarán dos antenas. La distancia entre ambas ranuras es de  $8.75\text{cm}$ , es decir,  $0.7\lambda$ . Además, en cada uno de los extremos se ha añadido un raíl que permite la fijación de unas piezas a otras, facilitando así la colocación de más de dos antenas a la distancia correcta. En la figura 3.15 se muestra el resultado de la impresión de cuatro piezas ya ensambladas.





Figura 3.15: Soporte y ensamblado del array de monopolos.

Para el caso de los parches, el soporte también se ha diseñado de forma que la distancia entre antenas sea de  $0.7\lambda$ . El resultado se muestra en la figura 3.16

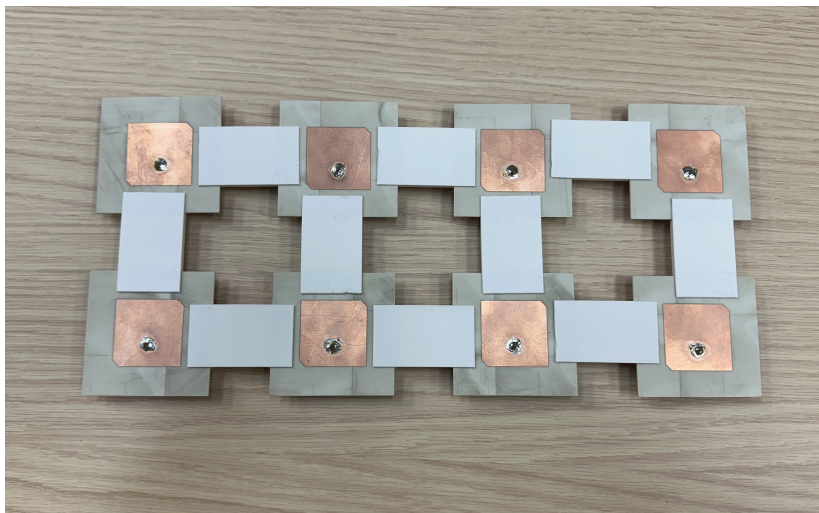


Figura 3.16: Soporte ensamblado del array de parches.

Por el propio método de fabricación, la precisión de los soportes para separar los elementos no es extremadamente alta. Esto significa que la distancia entre antenas no será exactamente  $0.7\lambda$ . Sin embargo, esta solución sigue siendo mucho mejor que si no se tu-



Dispositivo	Dirección serial
1	32711F7
2	3237BFC
3	3237C53
4	3271305

Tabla 1: Direcciones seriales de los dispositivos SDR.

viene nada para mantener las antenas en su posición, y es más que suficiente para ilustrar el funcionamiento del sistema.

### 3.3.- Conexión física

Antes de comenzar con la parte software del sistema, es conveniente tener una idea clara de las conexiones físicas que se van a realizar. En la figura 3.17 se muestra la numeración que se utilizará para referirse a cada uno de los dispositivos y en la tabla 1 se muestra la dirección serial correspondiente. Esto será de importancia durante el desarrollo del programa pues determinará los pesos a aplicar en cada puerto de cada USRP.

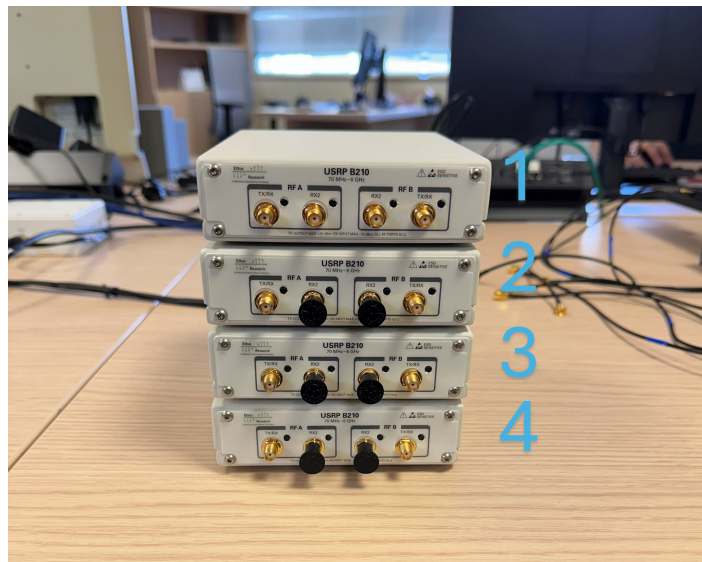


Figura 3.17: Numeración de referencia de los dispositivos SDR.

A diferencia de lo que ocurre con la fase, los USRP B210 sí permiten la sincronización en frecuencia mediante el uso de un reloj de referencia externo. En este caso, se ha utilizado

el OctoClock CDA-2990. Para estas conexiones, así como las conexiones entre los USRP y las antenas, se han utilizado cables coaxiales con conectores SMA, concretamente el modelo de cable RG174/U. Las conexiones entre los USRP, junto con las necesarias para la conexión desde un ordenador a los USRP se muestran en la figura 3.18. Cabe mencionar que el orden de conexión de los dispositivos con el OctoClock no es relevante, ya que este dispositivo se encarga de distribuir la misma señal de reloj a todos los dispositivos conectados. Lo mismo ocurre con la conexión USB de los USRP con el ordenador. Esta conexión se mantendrá durante el resto del proyecto.

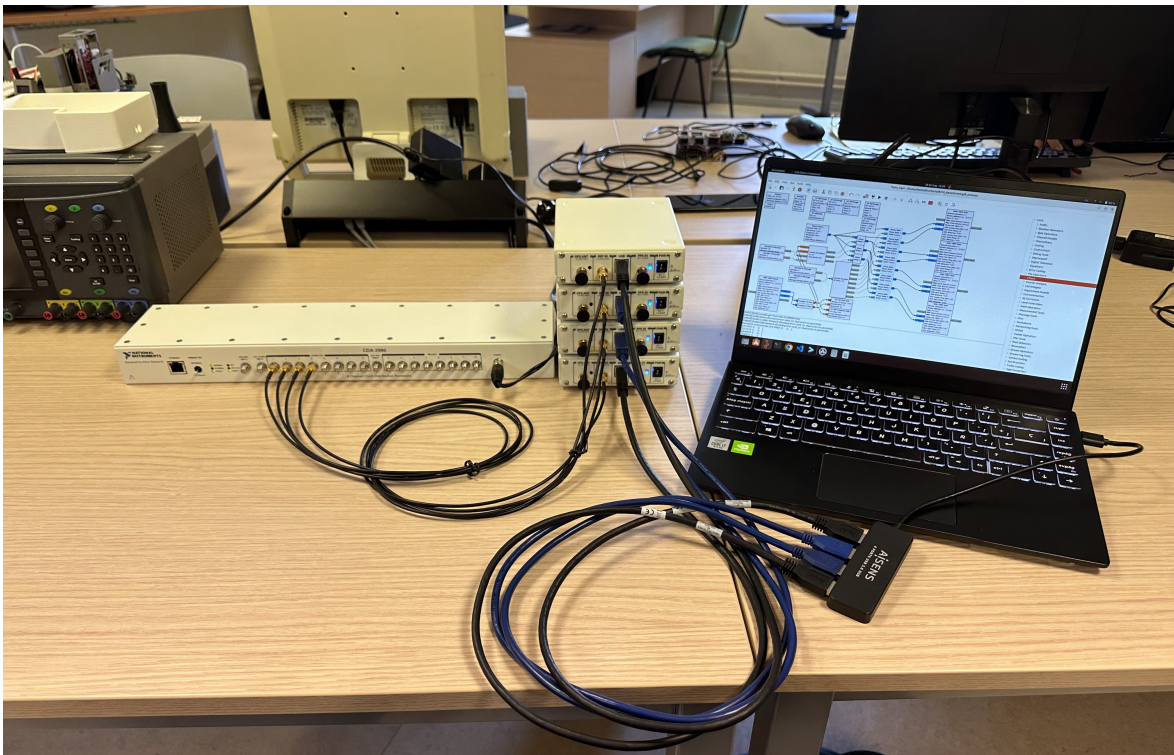


Figura 3.18: Conexión básica de los dispositivos.

Los dispositivos elegidos son capaces de transmitir y recibir señales simultáneamente. Por lo tanto, en la etapa de sincronización uno de ellos se utilizará como osciloscopio y referencia de fase. En esta etapa, se mantendrá un cable coaxial conectado a un transmisor y receptor de este dispositivo. Habrá otro cable que permanecerá conectado en uno de sus extremos al otro puerto receptor de este dispositivo y que irá conectándose a cada uno de los puertos transmisores del resto de dispositivos para su sincronización. Por ello, es importante que

estos cables sean de la misma longitud. En la figura 3.19 se muestra la conexión para la sincronización del dispositivo número 2.

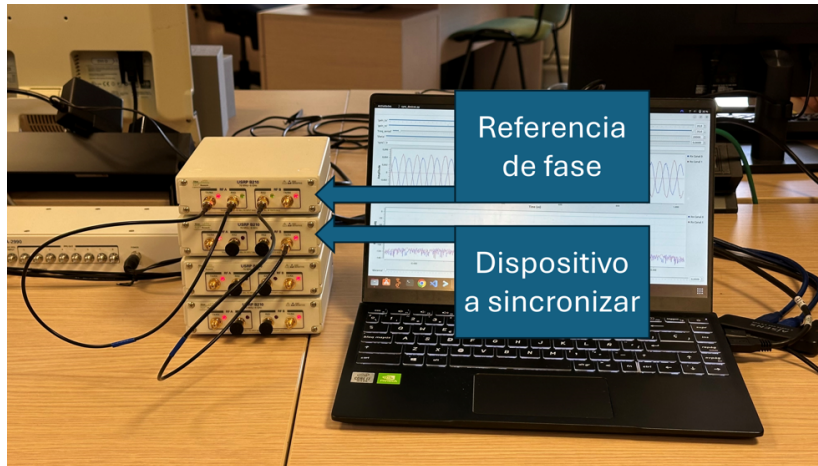


Figura 3.19: Conexión para la sincronización del dispositivo número 2.

Para la segunda fase, cada uno de los puertos transmisores se conectarán a cada una de las antenas a utilizar. En ese momento, se utilizará una interfaz de usuario que permitirá la selección del punto en el que se desea concentrar la energía radiada. En las figuras 3.20 y 3.21 se muestra el orden de conexión utilizado para cada caso.

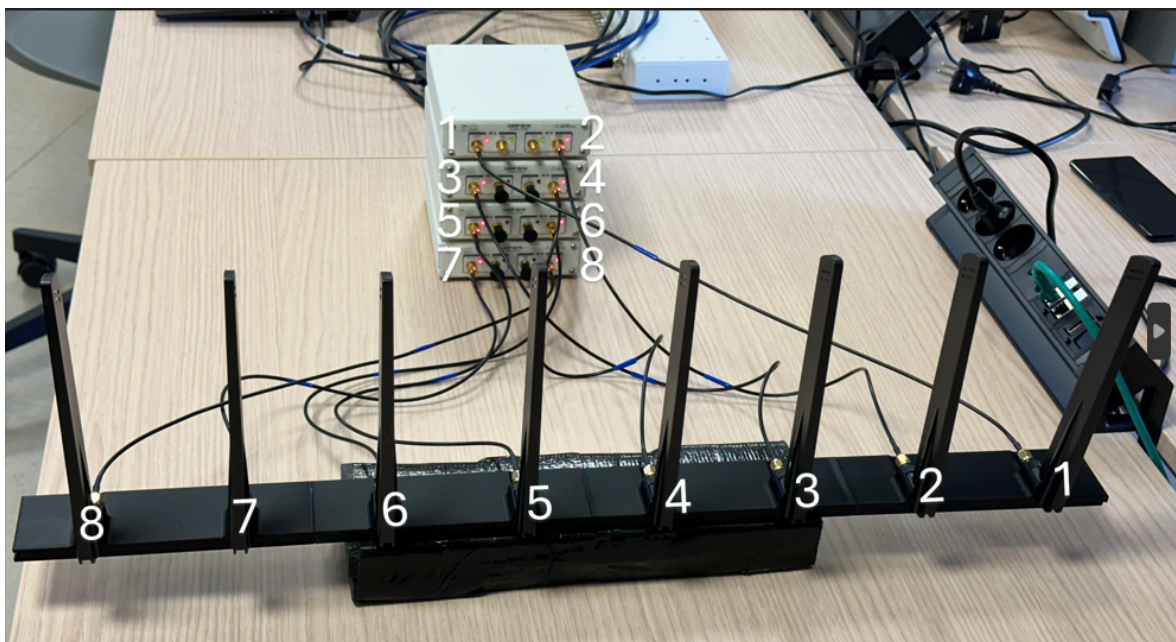


Figura 3.20: Conexión de las antenas para el array lineal.



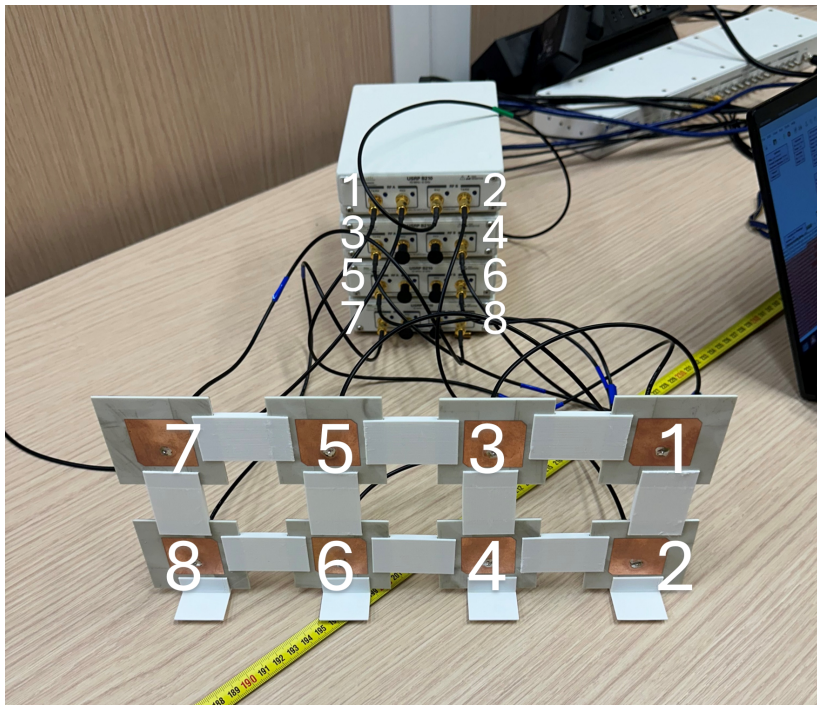


Figura 3.21: Conexión de las antenas para el array bidimensional.

### 3.4.- GNU Radio

GNU Radio es una herramienta gratuita y de código abierto que facilita la implementación de sistemas de comunicaciones basados en SDR. Facilita la creación de flujos de procesamiento de señal, utilizando bloques incluidos en la propia instalación del programa, y también la creación de bloques con código personalizado.

#### 3.4.1.- Diagrama de bloques

El programa desarrollado se muestra en su totalidad en la figura 3.22. El número junto a los bloques correspondientes a USRP Transmisores y USRP Receptores indica a qué dispositivo físico se corresponde cada bloque.

El bloque *Signal Source* genera una señal senoidal a la frecuencia especificada por la variable `freq_senial`. Esta señal se conecta, en el caso del USRP número 1 que actúa como referencia de fase, directamente al bloque *Phase Shift*, que aplicará el peso del elemento del array en forma de desfase. En el resto de casos, antes de este bloque se coloca un *Delay*, que

será el encargado de contrarrestar el desfase entre la señal de referencia y la del USRP en cuestión, de forma que la alimentación del array este sincronizada.

Las señales de salida de cada uno de los bloques *Phase Shift* van a una entrada de un bloque *USRP Sink*. La señal recibida en este puerto será la señal que se transmitirá utilizando la portadora a la frecuencia indicada por la variable `rf_freq`.

Con respecto a la etapa de recepción, se configuran los dos puertos receptores del USRP número 1 utilizando un bloque *USRP Source* y se extraen las partes reales de las señales recibidas utilizando el bloque *Complex To Real*. Estas señales se conectan, para su visualización, a un bloque *QT GUI Time Sink*, para el dominio del tiempo, y a un bloque *QT GUI Frequency Sink*, para el dominio de la frecuencia.

Estas mismas señales también se procesan en el bloque denominado *Cálculo de desfase*, encargado de calcular el retardo correspondiente entre señales. Este cálculo no se hará de forma continua, si no que se realizará cuando se detecte un cambio en la variable `sync`, que será modificada por el usuario de acuerdo a los dispositivos a sincronizar en cada momento. De acuerdo a esa misma variable, el so calculado se emitirá por el puerto *DelayOut* oportuno.

Finalmente, cuando todas los dispositivos estén sincronizados, se podrá modificar la dirección de apuntado deseada en la interfaz de usuario, que se enviará al bloque *Cálculo de pesos*. Este bloque calculará los pesos asociados a cada antena y los enviará a los bloques *Phase Shift* apropiados.

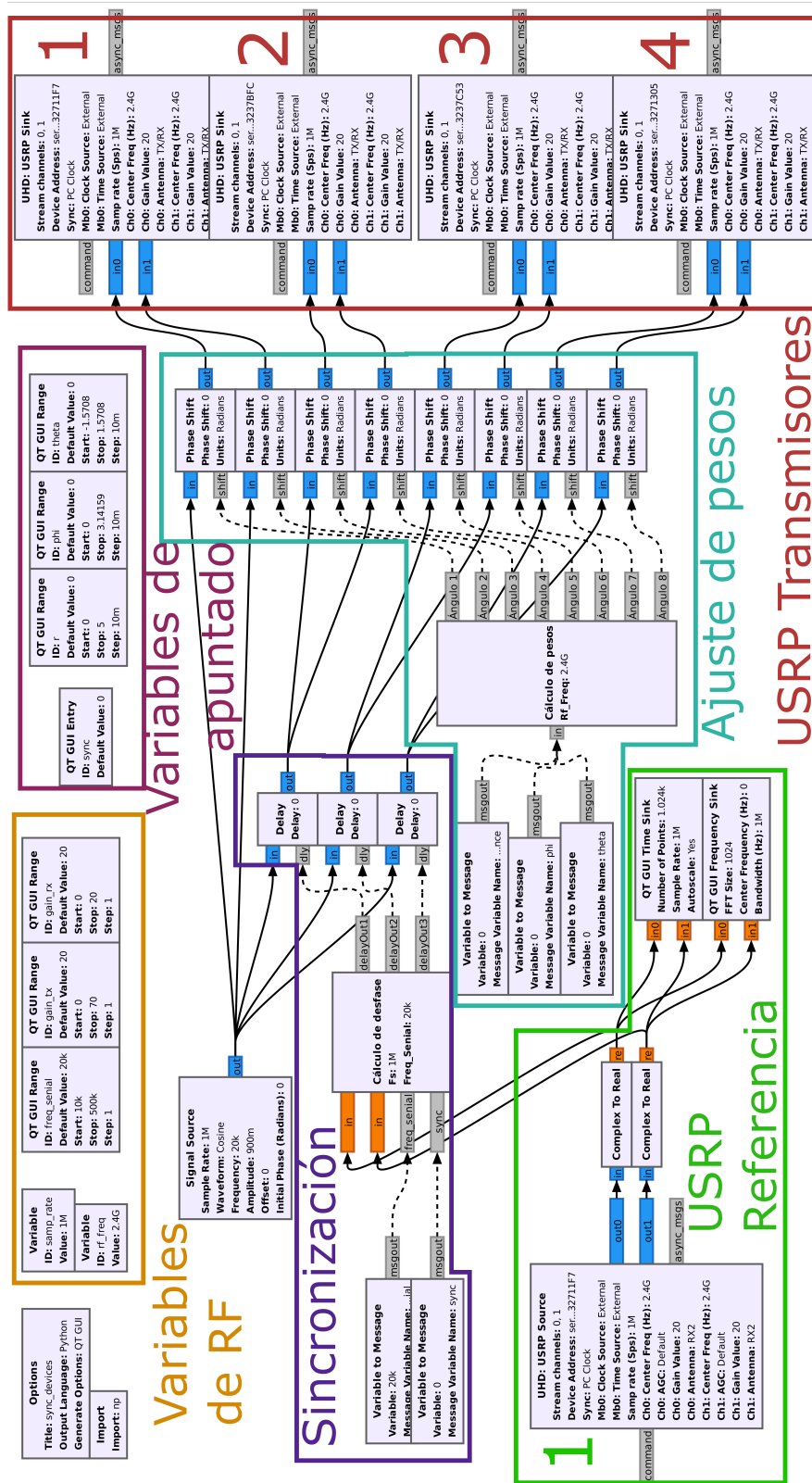


Figura 3.22: Implementación del sistema en GNU Radio.

A continuación, se describe la configuración y funcionamiento de los bloques más relevantes.

### 3.4.2.- Bloques personalizados

Para la creación de bloques en GNU Radio, se pueden utilizar dos lenguajes de programación: Python y C++. En este trabajo, se ha optado por el uso de Python, por su sencillez y puesto que el rendimiento de los bloques no es un factor crítico. Para la mayor parte del programa, se ha hecho uso de la funcionalidad de mensajes, una forma de comunicación asíncrona entre bloques. El código de los bloques mencionados en esta sección se incluyen en el anexo A.

#### 3.4.2.1. Bloque *Cálculo de desfase*

Como se menciona en 2.3.1, una de las herramientas que pueden ser utilizadas para calcular el desfase entre dos señales es la correlación cruzada. Este bloque, mostrado en la figura 3.23, implementa la lógica necesaria para ello. Las dos entradas 'in' corresponden a las señales que se desean comparar, y la entrada 'freq\_senial' permite al bloque conocer la frecuencia de la señal que se está transmitiendo. El bloque tiene tres salidas, una para cada uno de los USRP que se sincronizarán en este caso.

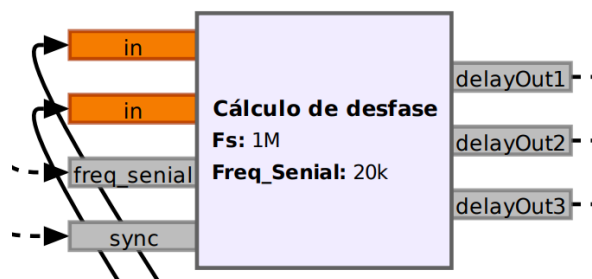


Figura 3.23: Bloque *Cálculo de desfase*.

El número de salidas se podría modificar fácilmente cambiando el código del mismo. Para ello, se ha de hacer doble clic en el bloque, y presionar en 'Open in Editor'. Aparecerá el código del bloque, que se muestra parcialmente en la figura 3.24. En la línea 18 se define el número de salidas con una variable llamada `self.number_of_usrp`, que podría

modificarse si se tuviesen más dispositivos.

```
1 import numpy as np
2 from gnuradio import gr
3 import pmt
4
5 class blk(gr.sync_block):
6     def __init__(self, fs=1, freq_senial=1):
7         gr.sync_block.__init__(
8             self,
9             name='Cálculo de desfase',
10            in_sig=[np.float32, np.float32],
11            out_sig=None
12        )
13        self.fs = fs
14        self.freq_senial = freq_senial
15        self.samples_per_period = self.fs // self.freq_senial
16        self.in0_buffer = np.array([])
17        self.in1_buffer = np.array([])
18        self.number_of_usrp = 3
19
20        self.portNamesOut = [f'delayOut{i}' for i in range(1, self.number_of_usrp + 1)]
```

Figura 3.24: Código parcial del bloque *Cálculo de desfase*.

El bloque utiliza dos *buffers* que almacenan las últimas muestras de señal recibidas. Al recibir una señal de sincronización, se calcula la correlación cruzada entre las dos señales almacenadas en ellos. Esta señal de sincronización se ha implementado utilizando los bloques *QT GUI Entry* y *Variable to Message*. El bloque *QT GUI Entry* hará que se muestre en pantalla una entrada con un nombre determinado, en este caso 'sync'. Cuando el usuario introduzca un valor entre 1, 2 y 3 (correspondientes a los dispositivos 2, 3 y 4), el bloque *Variable to Message* emitirá un mensaje con este valor al bloque *Cálculo de desfase*. En ese momento, el bloque calculará la autocorrelación entre las dos señales almacenadas, y enviará un mensaje al bloque *Delay* adecuado con el desfase calculado en número de muestras. Este mensaje hará que la señal correspondiente se retarde el número de muestras recibido.

### 3.4.2.2. Bloque *Cálculo de pesos*

Este bloque se encarga de calcular los pesos que se deben aplicar a cada una de las antenas del array de acuerdo al punto de enfoque. Es decir, este bloque calcula los valores  $w_n$  según el método de fase conjugada, tal y como se explica en la sección 2.2.



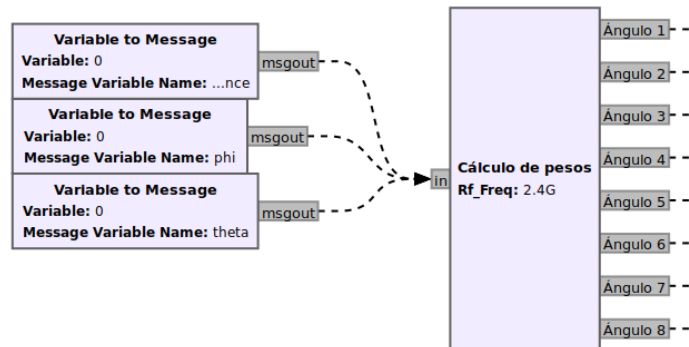


Figura 3.25: Bloque *Cálculo de pesos*.

Antes de ejecutar todo el sistema de GNU Radio, se ha de modificar el código de este bloque con el número de elementos y su disposición, de forma idéntica al caso anterior. Se abrirá una ventana de texto con el código del bloque como la mostrada en la figura 3.26. En las líneas 15 y 16 se han de modificar los valores de `self.N` y `self.M` con el número de elementos a lo largo del eje X e Y, respectivamente. Es decir, se supone el array dispuesto en el plano *XY*, y la dirección de propagación Z. También se podría modificar la distancia entre elementos modificando la variable `self.d`. Por ejemplo, para el caso del array lineal, se ha de modificar los valores `self.N = 8` y `self.M = 1`. Para el caso del array bidimensional, se ha de poner `self.N = 4` y `self.M = 2` (o viceversa, solo se ha de tener en cuenta el correspondiente cambio en las coordenadas del punto de enfoque).

```

1  import numpy as np
2  import pmt
3  from gnuradio import gr
4
5  class blk(gr.sync_block):
6      def __init__(self, rf_freq=1):
7          gr.sync_block.__init__(self,
8                                 name='Cálculo de pesos',
9                                 in_sig=[],
10                                out_sig=[])
11
12         self.port_name = "in"
13         self.message_port_register_in(pmt.intern(self.port_name))
14         self.set_msg_handler(pmt.intern(self.port_name), self.handle_msg)
15
16         self.N = 8# Número de elementos a lo largo del eje x
17         self.M = 1# Número de elementos a lo largo del eje y
18         self.d = 0.7 # Separación entre elementos en lambdas
19         self.portNameOut = [f"Ángulo {i+1}" for i in range(self.N*self.M)]

```

Figura 3.26: Código parcial del bloque *Cálculo de pesos*.

Cuando se ejecute el programa, en la interfaz de usuario se mostrarán tres barras deslizantes, una para cada coordenada esférica  $(r, \theta, \phi)$ . Estas barras se corresponden con los bloques *QT GUI Range* que se muestran en 'Variables de apuntado' en la figura 3.22. Cuando se modifiquen estos valores, los bloques '*Variable to Message*' enviarán un mensaje al bloque *Cálculo de pesos* con los nuevos valores. Este bloque calculará los pesos y los enviará a los bloques *Phase Shift* correspondientes, que aplicarán el desfase necesario a la señal del elemento correspondiente.

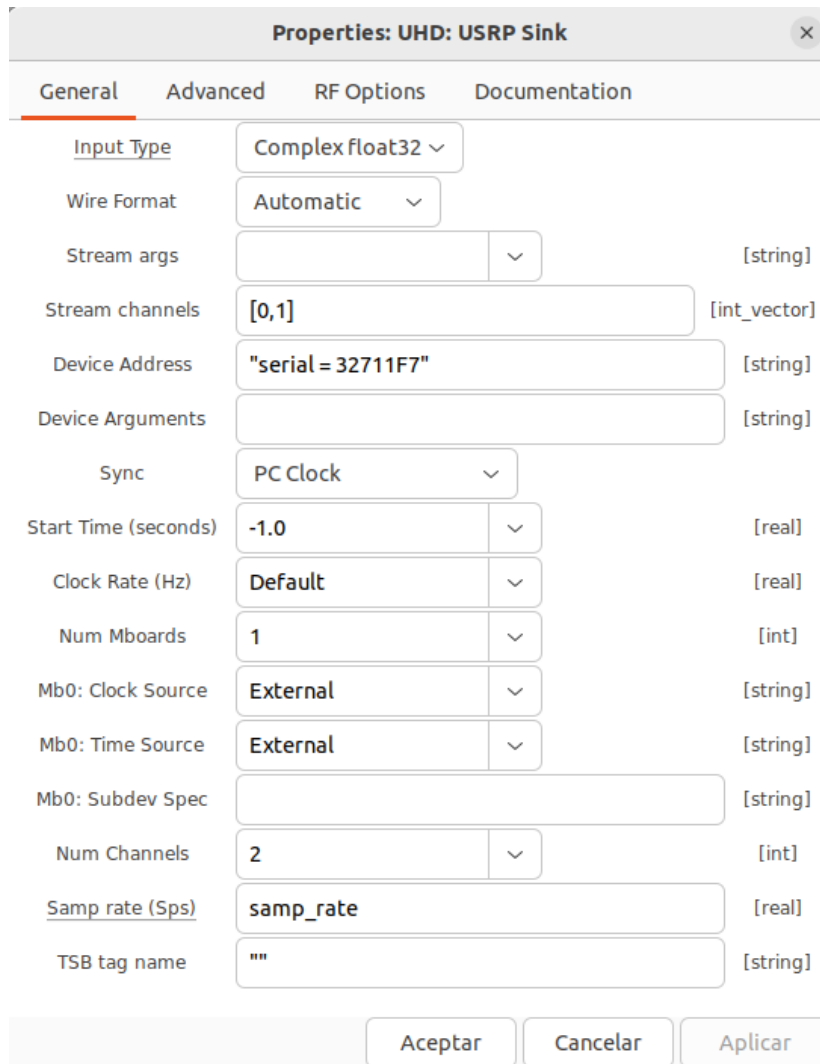
Además, este bloque hará que se impriman las posiciones de los elementos en la consola de GNU Radio. Esto es útil para conectar las antenas correctamente.

### 3.4.3.- Bloques *UHD: USRP*

Los bloques tipo *UHD: USRP* son bloques que permiten la comunicación con los dispositivos USRP. Hay dos tipos: *USRP Sink* y *USRP Source*. El primero indica al dispositivo qué señal transmitir y en qué puerto, mientras que el segundo permite obtener las señales recibidas por el dispositivo ya demoduladas.

Para el caso de la parte de transmisión, la configuración es la mostrada en las figuras 3.27 y 3.28, correspondientes a la configuración general y de radiofrecuencia (RF) respectivamente. Estas son aplicables a todos los dispositivos.

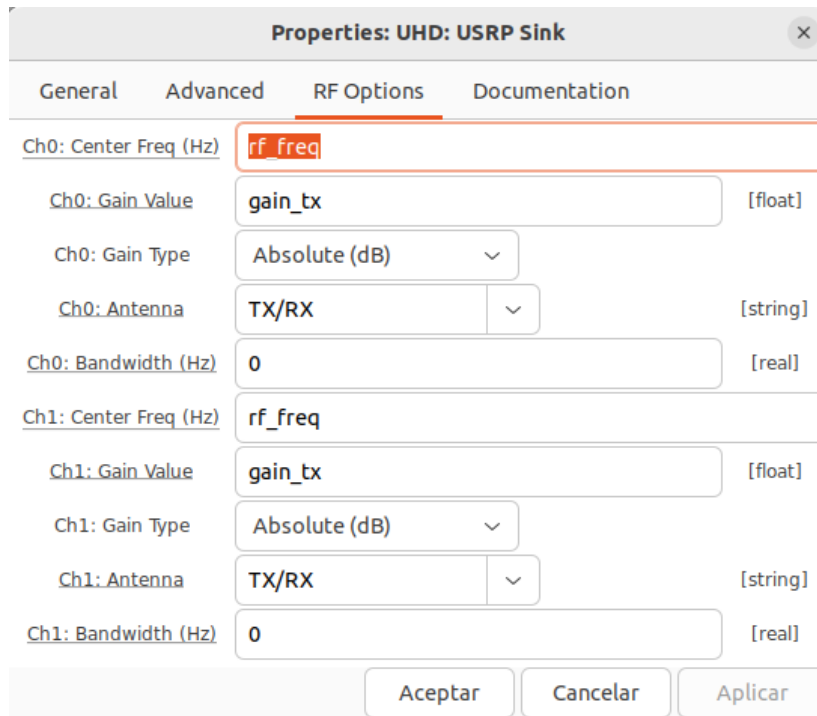
Con respecto a la configuración general (figura 3.27), se ha de indicar la dirección serial del dispositivo correspondiente. Además, en el campo *Stream channels* se introducen los valores 0 y 1 de forma que se utilicen ambos canales. En la sección *Mb0: Clock source* y *Mb0: Time Source* se selecciona 'External' para que el dispositivo se sincronice con el reloj externo (el OctoClock). El número de canales (*Num Channels*) es dos, puesto que a cada dispositivo se conectarán dos antenas.



Property	Value	Type
Input Type	Complex float32	enum
Wire Format	Automatic	enum
Stream args		[string]
Stream channels	[0,1]	[int_vector]
Device Address	"serial = 32711F7"	[string]
Device Arguments		[string]
Sync	PC Clock	enum
Start Time (seconds)	-1.0	[real]
Clock Rate (Hz)	Default	[real]
Num Mboards	1	[int]
Mb0: Clock Source	External	[string]
Mb0: Time Source	External	[string]
Mb0: Subdev Spec		[string]
Num Channels	2	[int]
Samp rate (Sps)	samp_rate	[real]
TSB tag name	""	[string]

Figura 3.27: Configuración general del bloque *USRP Sink*

En las opciones de RF (figura 3.28), se selecciona, para cada canal, la frecuencia de trabajo, que en este caso vendrá indicada por la variable `rf_freq` y valor de la ganancia, que en este caso se podrá cambiar mientras se ejecuta el programa, ya que la variable `gain_tx` corresponde a un bloque *QT GUI Range*. En los campos *Ch0: Antenna* y *Ch1: Antenna* se selecciona 'TX/RX' para que el dispositivo pueda transmitir (si se tratase de un bloque *USRP Source*, esta opción activaría el puerto como receptor, al ser un bloque *USRP Sink*, se activa como transmisor).



The image shows a software dialog box titled "Properties: UHD: USRP Sink" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Advanced", "RF Options" (which is selected and underlined), and "Documentation".

Under the "RF Options" tab, there are two channel configurations, Ch0 and Ch1. Each channel has several parameters:

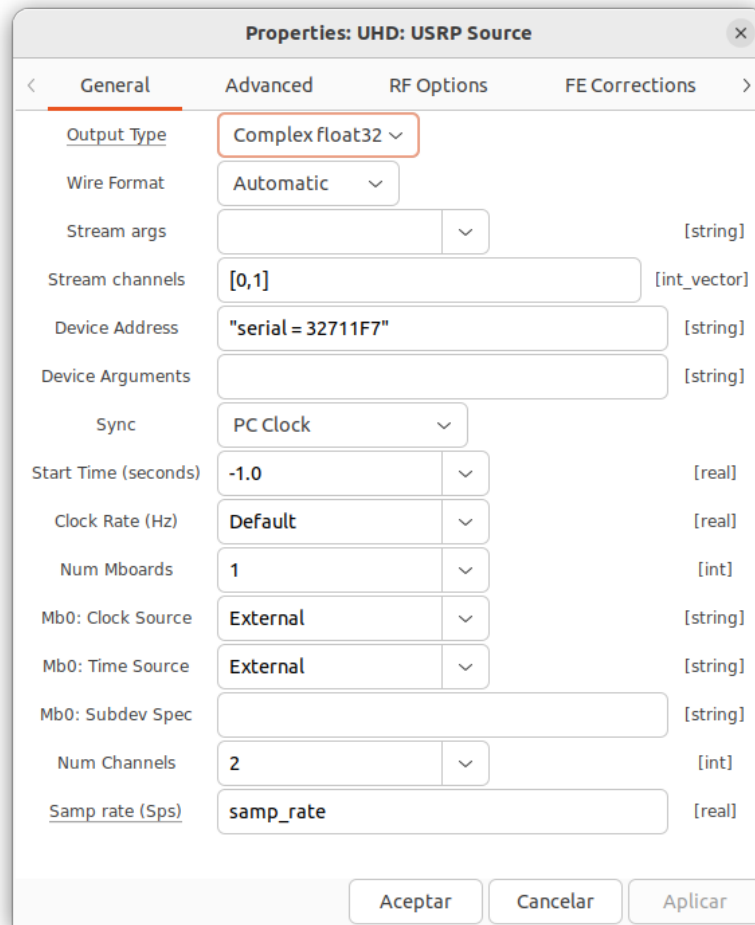
- Ch0: Center Freq (Hz):** A text input field containing "rf\_freq".
- Ch0: Gain Value:** A text input field containing "gain\_tx" with a "[float]" label to its right.
- Ch0: Gain Type:** A dropdown menu showing "Absolute (dB)".
- Ch0: Antenna:** A dropdown menu showing "TX/RX" with a "[string]" label to its right.
- Ch0: Bandwidth (Hz):** A text input field containing "0" with a "[real]" label to its right.
- Ch1: Center Freq (Hz):** A text input field containing "rf\_freq".
- Ch1: Gain Value:** A text input field containing "gain\_tx" with a "[float]" label to its right.
- Ch1: Gain Type:** A dropdown menu showing "Absolute (dB)".
- Ch1: Antenna:** A dropdown menu showing "TX/RX" with a "[string]" label to its right.
- Ch1: Bandwidth (Hz):** A text input field containing "0" with a "[real]" label to its right.

At the bottom of the dialog, there are three buttons: "Aceptar", "Cancelar", and "Aplicar".

Figura 3.28: Configuración RF del bloque *USRP Sink*

Para el caso del dispositivo número 1, que será el dispositivo que además actuará como referencia de fase, se ha de utilizar también un bloque *USRP Source*. La configuración del mismo se muestra en las figuras 3.29 y 3.30.

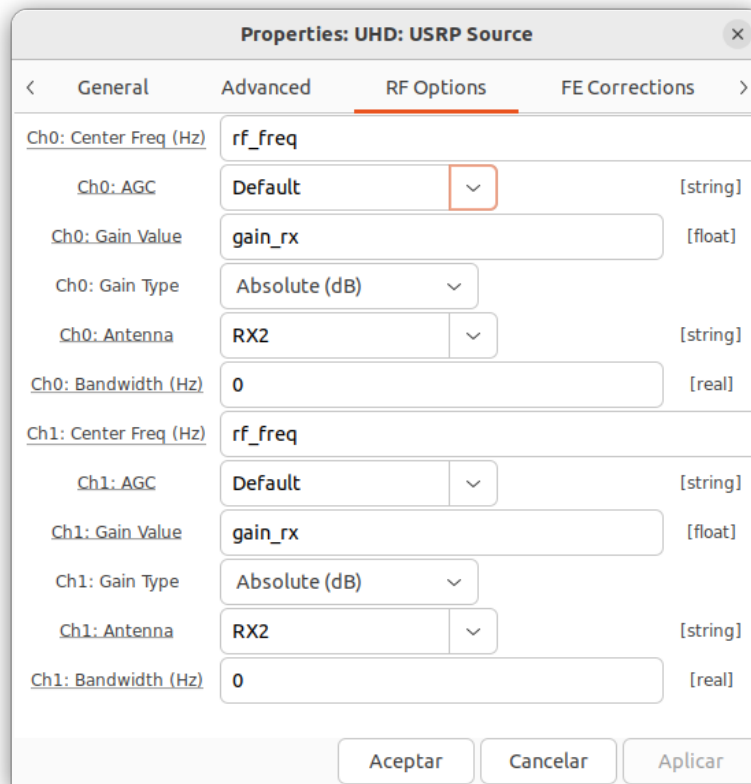
La configuración general, mostrada en la figura 3.29, es idéntica a la del caso de los dispositivos transmisores.



Property	Value	Unit/Type
Output Type	Complex float32	
Wire Format	Automatic	
Stream args		[string]
Stream channels	[0,1]	[int_vector]
Device Address	"serial = 32711F7"	[string]
Device Arguments		[string]
Sync	PC Clock	
Start Time (seconds)	-1.0	[real]
Clock Rate (Hz)	Default	[real]
Num Mboards	1	[int]
Mb0: Clock Source	External	[string]
Mb0: Time Source	External	[string]
Mb0: Subdev Spec		[string]
Num Channels	2	[int]
Samp rate (Sps)	samp_rate	[real]

Figura 3.29: Configuración general del bloque *USRP Source*

La configuración RF, mostrada en la figura 3.30, es ligeramente diferente. La frecuencia central es idéntica, pero la variable que controla la ganancia es diferente, y se corresponde con una variable denominada `gain_rx`. En los campos *Ch0: Antenna* y *Ch1: Antenna* se selecciona 'RX2' para que el dispositivo pueda recibir señales en ambos puertos.



Channel	Property	Value	Type
Ch0	Center Freq (Hz)	rf_freq	[string]
	AGC	Default	[string]
	Gain Value	gain_rx	[float]
	Gain Type	Absolute (dB)	[string]
	Antenna	RX2	[string]
	Bandwidth (Hz)	0	[real]
Ch1	Center Freq (Hz)	rf_freq	[string]
	AGC	Default	[string]
	Gain Value	gain_rx	[float]
	Gain Type	Absolute (dB)	[string]
	Antenna	RX2	[string]
	Bandwidth (Hz)	0	[real]

Figura 3.30: Configuración RF del bloque *USRP Source*

## 4.- Resultados y mediciones

En esta sección se detallan los pasos necesarios para poner en marcha el sistema desarrollado, y se describen las medidas realizadas. Se asume que el lector ya tiene instalado GNU Radio y los drivers de los USRP. En caso contrario, se recomienda seguir las instrucciones de la documentación oficial de GNU Radio [8].

### 4.1.- Puesta en marcha

Para el correcto funcionamiento del enfoque, es necesario que los USRP estén sincronizados. Esto es lo primero que se ha de hacer al iniciar el programa. Una vez sincronizados, se puede proceder al enfoque.

#### 4.1.1.- Sincronización de los USRP

Con todos los bloques correctamente colocados, se puede iniciar el programa pulsando en el botón indicado en la figura 4.1.

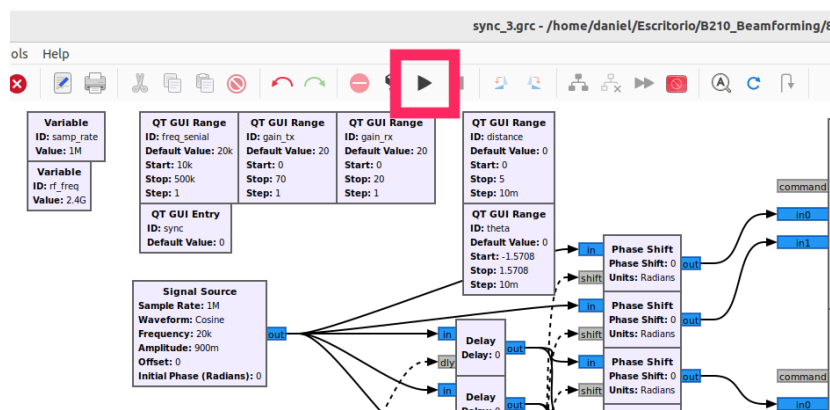


Figura 4.1: Botón de inicio del programa.

Se mostrará la ventana en la figura 4.2.



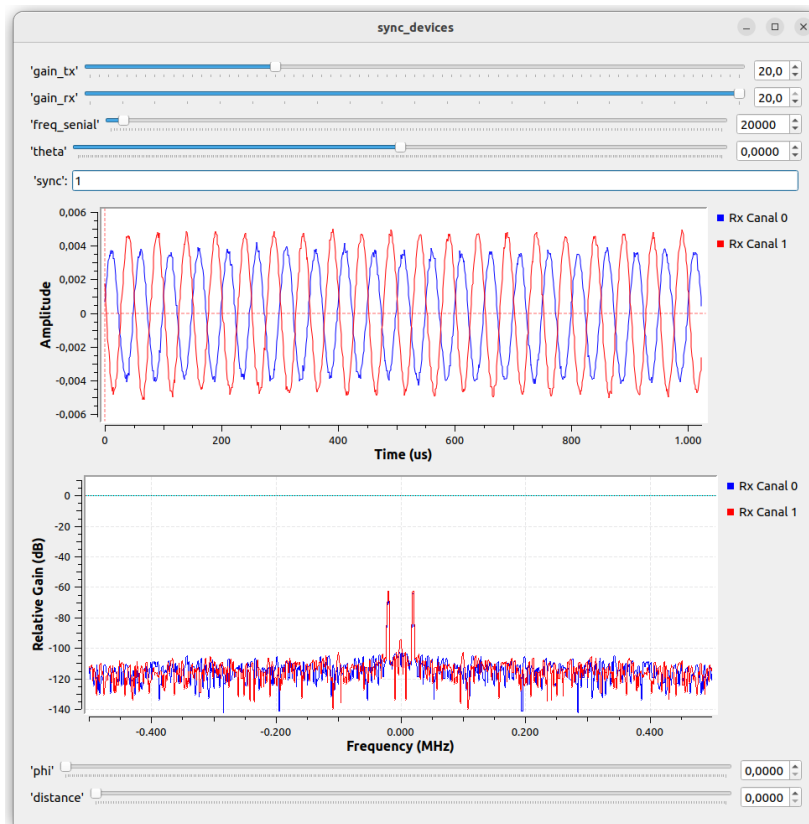


Figura 4.2: Ventana mostrada al inicio del programa.

Si en ese momento hay conectados dos puertos transmisores a los dos puertos receptores del USRP 1, en la ventana aparecerán las dos señales recibidas, tanto en el dominio del tiempo como en el de la frecuencia. Al observar el espectro, se observa que ambas señales están perfectamente solapadas gracias al Octoclock. Sin embargo, en el dominio del tiempo, se observa un desfase entre ambas señales. La señal del canal 0 (azul), representa la señal de referencia (que corresponde al cable conectado entre el puerto transmisor y receptor izquierdos del USRP 1), y la señal del canal 1 (rojo), la del dispositivo a sincronizar.

Para proceder a la sincronización del primer elemento, se utiliza el campo 'sync', que se encuentra en la parte superior de la gráfica de las señales. Como en este caso el dispositivo a sincronizar es el número 2, se ha de introducir el número 1 y presionar la tecla 'Enter'. Se verá las dos señales mostradas pasar a estar en fase, como se muestra en la figura 4.3.

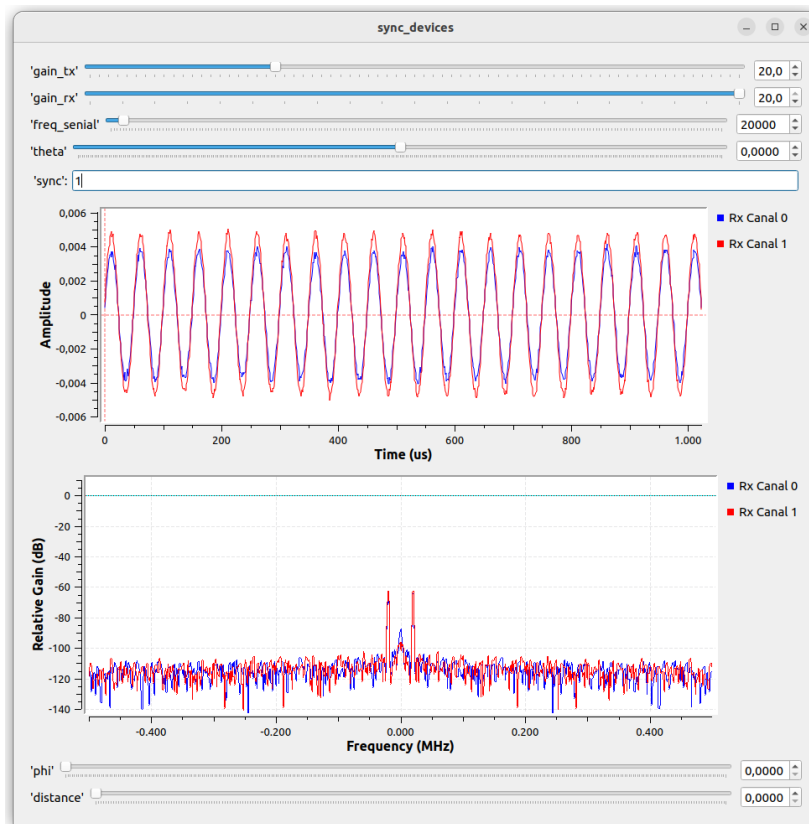


Figura 4.3: Señales mostradas tras la sincronización.

Antes de pulsar en cualquier otro lugar, se ha de devolver el valor del campo 'sync' a 0, de forma que no se intente sincronizar de nuevo (de otra forma, GNU Radio enviará una señal con el valor cuando se pulse en otro lugar de la pantalla, haciendo que los bloques de la cadena se activen). De esta forma, el USRP 2 estará sincronizado con el USRP que actúa como referencia, es decir, el número 1.

Para el resto de USRP, se procederá de forma similar, cambiando el extremo cable conectado anteriormente al puerto transmisor del USRP 2 al puerto correspondiente de otro USRP. El cambio para el dispositivo 3 se muestra en la figura 4.4.



Figura 4.4: Cambio de la conexión para la sincronización del dispositivo 3.

En el programa, se verá como la señal del canal 1 (rojo) cambia a la del dispositivo recién conectado y deja de estar en fase con la señal del canal 0 (azul). De forma similar al caso anterior, si ahora se introduce el número 2 en el campo 'sync' y se pulsa 'Enter' se observará como las señales pasan a estar en fase. Cada vez que se aplica una corrección en el desfase, el programa imprimirá el valor en consola. En la figura 4.5 se muestran los mismos tras haber sincronizado tres dispositivos.

```
Retraso estimado: 21 muestras.
Publicado mensaje '(sync1 . 21)' en el puerto 'delayOut1.'
Retraso estimado: -22 muestras.
Publicado mensaje '(sync2 . 28)' en el puerto 'delayOut2.'
Retraso estimado: 24 muestras.
Publicado mensaje '(sync3 . 24)' en el puerto 'delayOut3.'
```

Figura 4.5: Valores mostrados en la consola tras la sincronización.

#### 4.1.2.- Conformado de haz

El cálculo de los pesos necesarios para el conformado de haz se realiza de forma automática por el bloque 'Cálculo de pesos'. Este bloque se activará por primera vez cuando se mueva uno de los dos deslizadores correspondientes ('distance', 'phi' o 'theta'). Estos va-

lores serán utilizados por el bloque para ajustar cada elemento del array de acuerdo al punto de enfoque deseado. En la consola se mostrarán los pesos calculados para cada elemento, así como el punto de enfoque. Por ejemplo, suponiendo un ángulo de apuntado de 30 grados y una distancia de 1 metro, el programa muestra los valores en la figura 4.6.

```
Apuntando a  $r = 1.0$  y  $\theta = 30.0000701530499$   
Publicado ángulo 9.21028191248049 [rad] en el puerto Ángulo 1.  
Publicado ángulo 6.305008076374428 [rad] en el puerto Ángulo 2.  
Publicado ángulo 3.6025331656085413 [rad] en el puerto Ángulo 3.  
Publicado ángulo 7.418039743100302 [rad] en el puerto Ángulo 4.  
Publicado ángulo 5.220497790967123 [rad] en el puerto Ángulo 5.  
Publicado ángulo 3.3307483831988733 [rad] en el puerto Ángulo 6.  
Publicado ángulo 8.070117537175314 [rad] en el puerto Ángulo 7.  
Publicado ángulo 6.908166770135061 [rad] en el puerto Ángulo 8.
```

Figura 4.6: Valores de los pesos calculados para un ángulo de apuntado de 30 grados y una distancia de 1 metro.

## 4.2.- Medidas

Para evaluar el rendimiento real del sistema desarrollado, se han llevado a cabo una serie de medidas, que se compararán con los valores teóricos, permitiendo así evaluar el modelo utilizado y la capacidad del array para enfocar la energía en un punto dado en la realidad. Estas medidas se llevarán a cabo haciendo uso del array lineal de monopolos, puesto que facilita la realización de medidas en el laboratorio, y la descripción de su comportamiento teórico es más sencilla, puesto que en el plano de las medidas se considerarán como omnidireccionales. No obstante, también se han realizado algunas pruebas con el array de parches, que se describirán en la sección correspondiente. Todas las medidas se han llevado a cabo para un tono de 20 kHz, con una frecuencia portadora de 2.4 GHz.

#### 4.2.1.- Sincronización frente a apuntado.

En primer lugar, se han tomado medidas para el caso en que el array se encuentra solamente sincronizado, es decir, con las antenas alimentadas de forma uniforme, y también con un punto de enfoque a 1 metro del array en la dirección Z. Las curvas teóricas para ambos casos se representan en la figura 4.7. En ellas se observa un cambio notable entre ambas situaciones. En el punto de enfoque, la potencia recibida es aproximadamente 3.5 dB superior a la recibida sin enfoque. Con respecto a la curva teórica para el caso de enfoque, el máximo de la misma (ignorando la zona más cercana al array) ocurre entre 50 y 60 cm de distancia del centro del conjunto, y no en el punto de enfoque. Como se describe en la sección 3.1, a este fenómeno se le conoce como 'focal shift', o desplazamiento del foco.

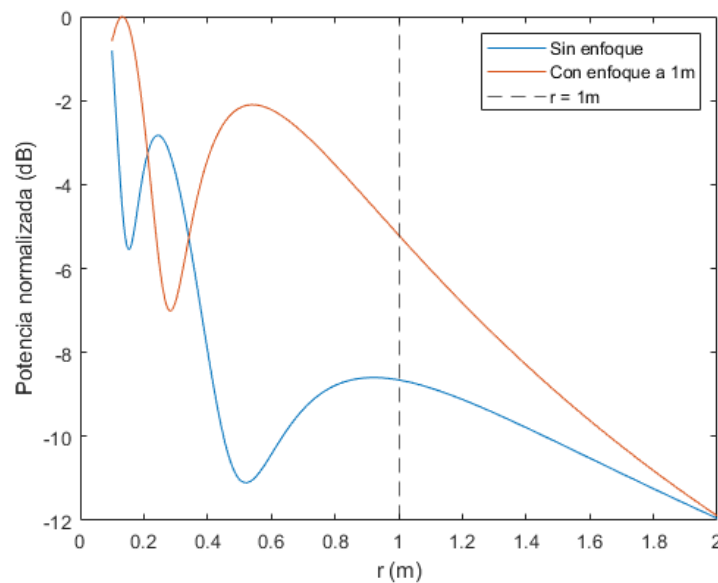


Figura 4.7: Potencia recibida sin enfoque y con enfoque a 1 metro.

El banco de medidas utilizado se muestra en la figura 4.8. Las antenas se han elevado ligeramente de forma que la antena receptora esté a la misma altura que las transmisoras.

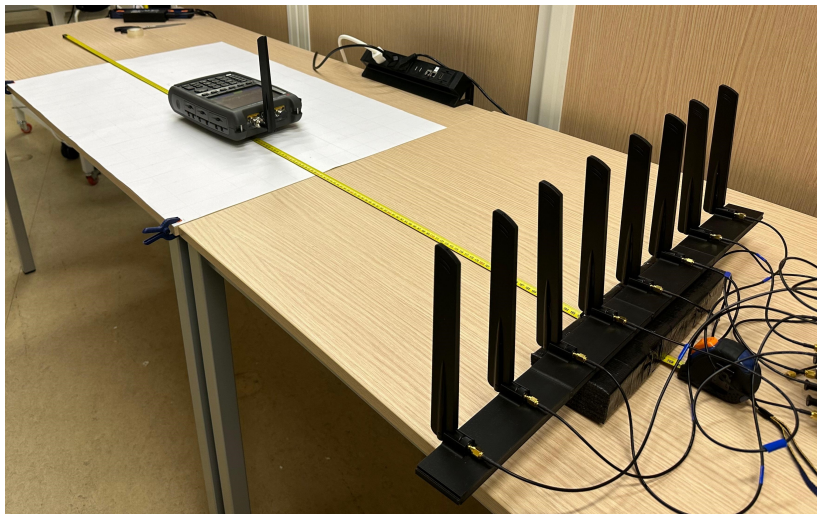


Figura 4.8: Banco de medidas para apuntado a 0 grados.

Los resultados de las medidas se muestran en la figura 4.9. Los valores teóricos se han normalizado con respecto al máximo teórico, y las medidas con respecto al máximo valor obtenido. En la gráfica obtenida, se observa que las curvas medidas una tendencia muy similar a las curvas teóricas, cumpliéndose además las diferencias de nivel entre ambas. Concretamente, en el punto de enfoque se ha medido una diferencia de potencia de 3.8 dB, muy similar a la diferencia teórica de 3.5 dB.

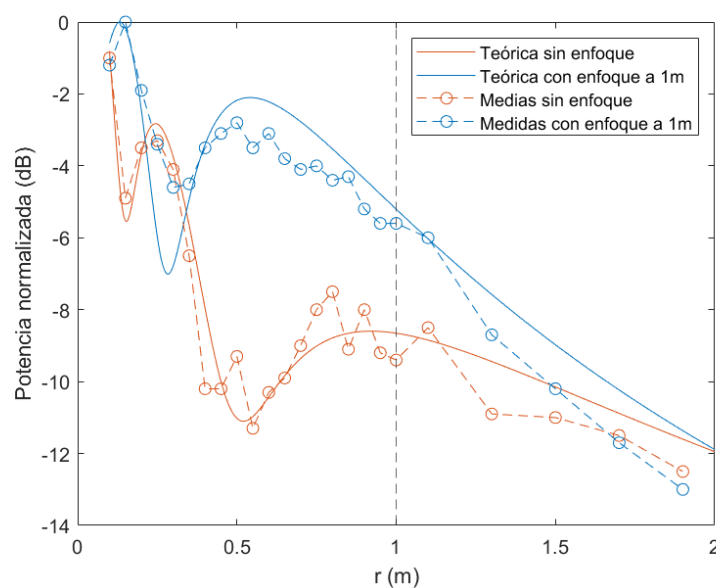


Figura 4.9: Resultados de las medidas sin enfoque y con enfoque a 1 metro.

Por tanto, en este punto, el sistema se comporta de forma muy similar a lo esperado, lo que indica que el sistema de sincronización y el cálculo de pesos funcionan correctamente.

#### 4.2.2.- Enfoque en zona cercana al array

Como se ha observado en la sección anterior, para un valor de apuntado a una distancia de 1 metro del array existe una gran diferencia entre la posición del máximo de energía y la posición del lóbulo producto del enfoque. Por tanto, para comparar la diferencia entre enfoques a distinta distancia, se han tomado medidas para una zona de enfoque más cercana al array, de entre 10 y 40 cm. La elección de este rango de distancias se ha hecho en base a los resultados de representar las curvas teóricas de potencia recibida para varios puntos de enfoque, manteniendo el ángulo de apuntado constante, en este caso 0 grados. En la figura 4.10 se muestran las curvas teóricas para varios valores de distancia de enfoque. En ellas, se aprecia como a medida que la distancia del array al punto de enfoque aumenta, el máximo de potencia recibida se aleja cada vez más de su valor ideal.

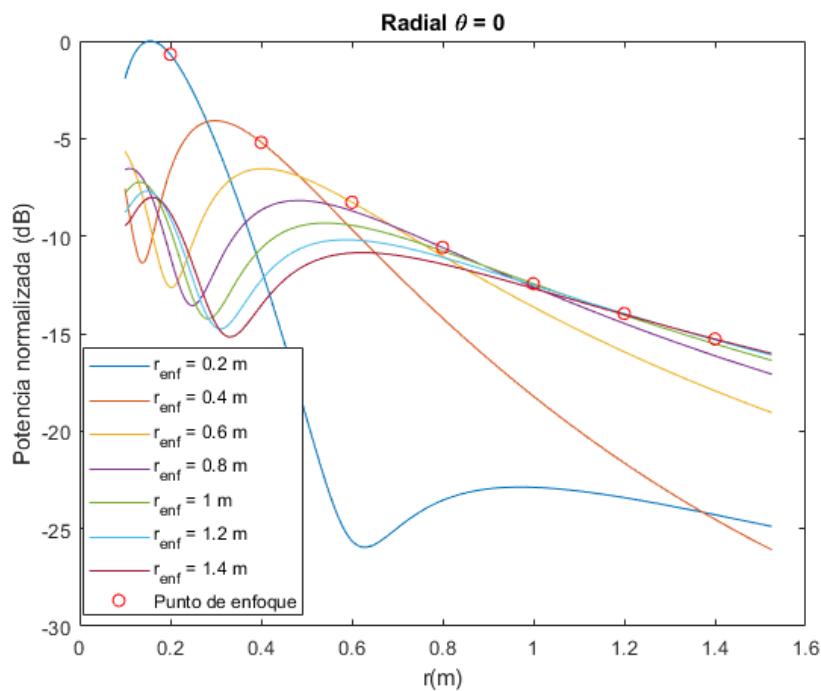


Figura 4.10: Curvas teóricas para diferentes distancias de enfoque con 8 elementos.



Esta limitación del array para concentrar la energía en el punto deseado está directamente relacionada con su tamaño. Si se aumentara el número de elementos a, por ejemplo, 64, los resultados mejorarían significativamente. En la figura 4.11 se muestra la simulación para los mismos puntos de enfoque que en la figura 4.10, pero para el caso de un array de 64 elementos.

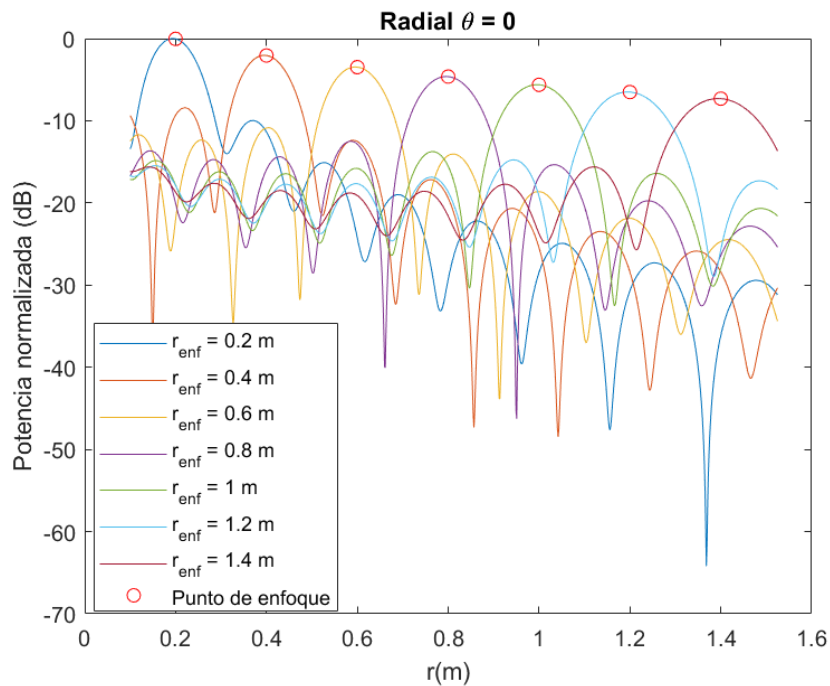


Figura 4.11: Curvas teóricas para diferentes distancias de enfoque con 64 elementos.

En este caso, los máximos sí se localizan de forma prácticamente perfecta en el punto de enfoque. Teniendo en cuenta esto, se han tomado medidas a lo largo de la radial que une el centro del array con el punto de enfoque para dos distancias de enfoque, 20 cm y 30 cm, y para dos ángulos de apuntado, 0 y 30 grados. En cada punto se ha medido la potencia recibida utilizando el mismo banco de medidas que en la sección anterior. Para el caso de apuntado a 30 grados, se ha girado el array de forma correspondiente, como se muestra en la figura 4.12.

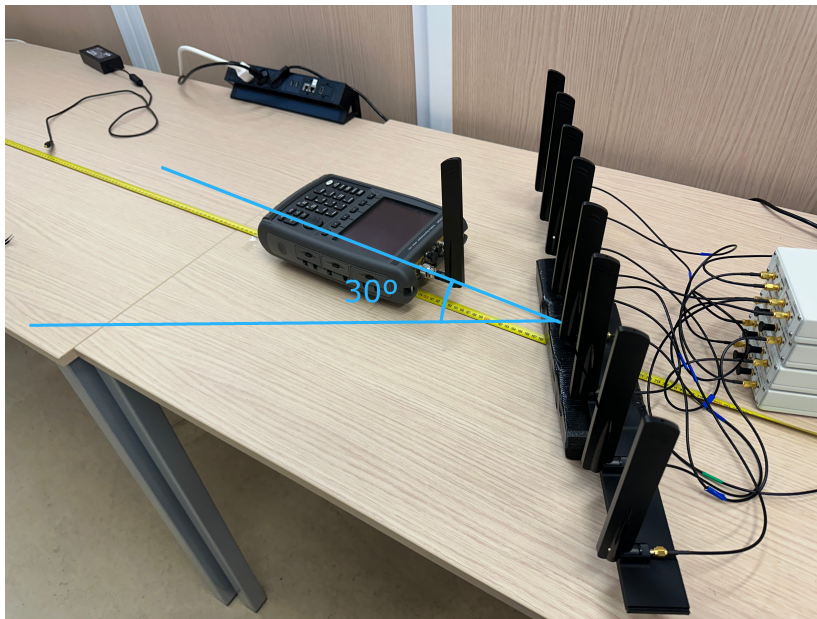


Figura 4.12: Banco de medidas para apuntado a 30 grados.

Los resultados de estas medidas se han representado en la figura 4.13 para el caso de enfoque a 0 grados y la figura 4.15 para el caso de 30 grados. El máximo de cada medida se representa con un punto rojo, y las líneas verticales discontinuas indican la posición del punto de enfoque.

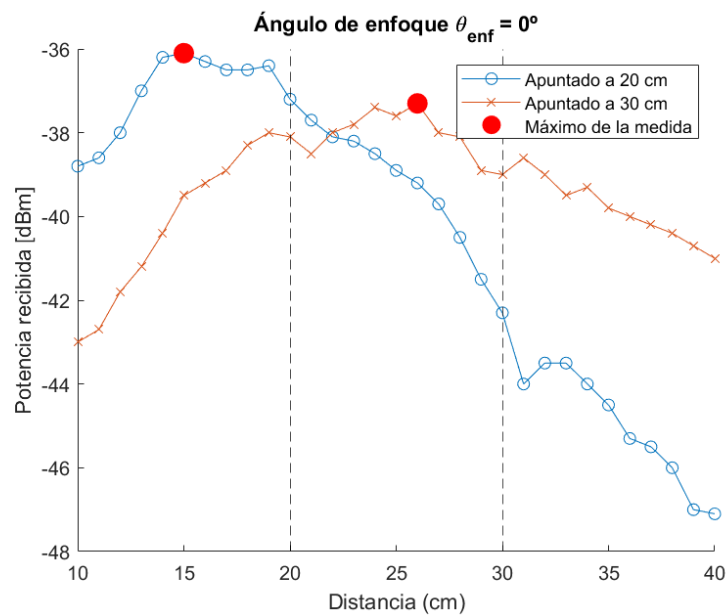


Figura 4.13: Resultados de las medidas radiales con apuntado a 0 grados.

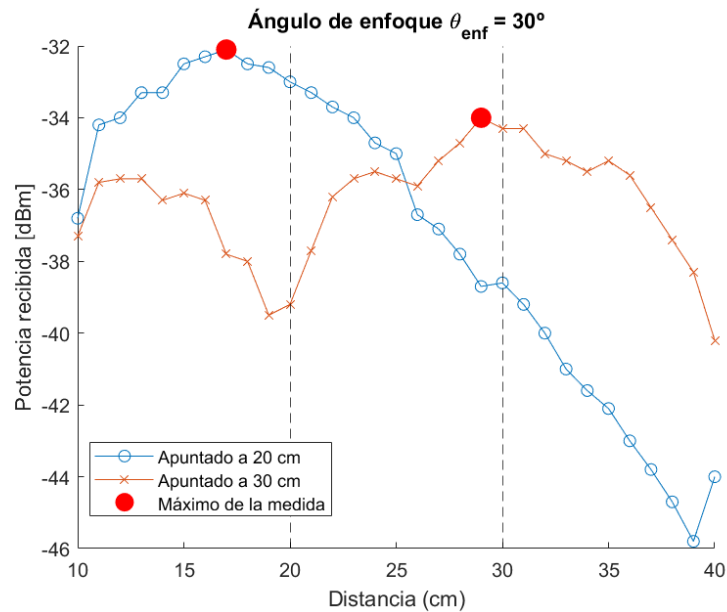


Figura 4.14: Resultados de las medidas radiales con apuntado a 30 grados.

Para ambos casos, se percibe el cambio esperado en la localización del máximo de la energía. En las figuras 4.15 y 4.16 se compara para cada caso los resultados obtenidos con los valores teóricos. Las medidas se han normalizado con respecto al máximo de potencia recibida, y las curvas teóricas se han normalizado con respecto al máximo de la curva correspondiente.

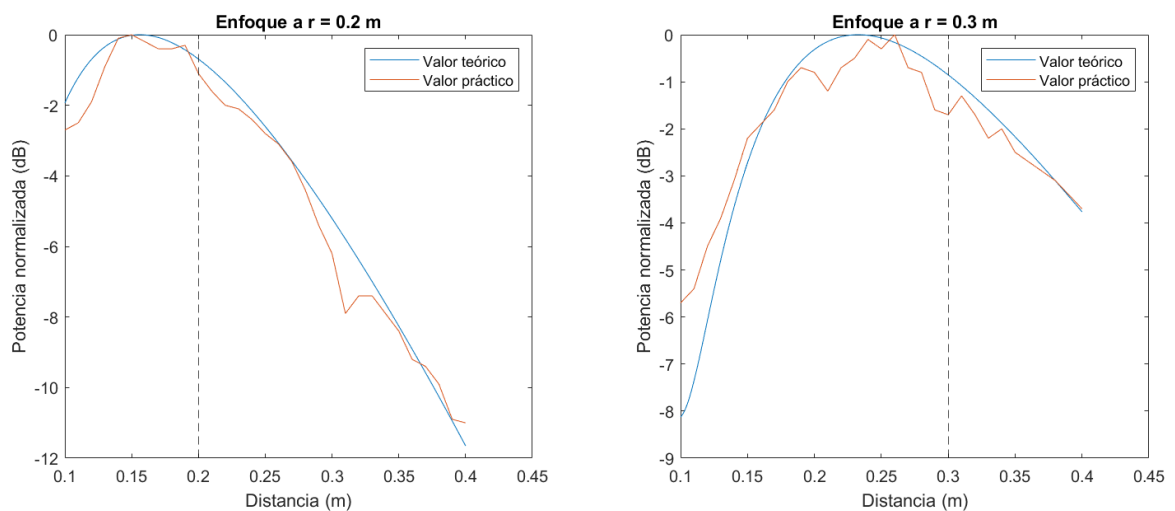


Figura 4.15: Valores de potencia normalizados con apuntado a 0 grados.

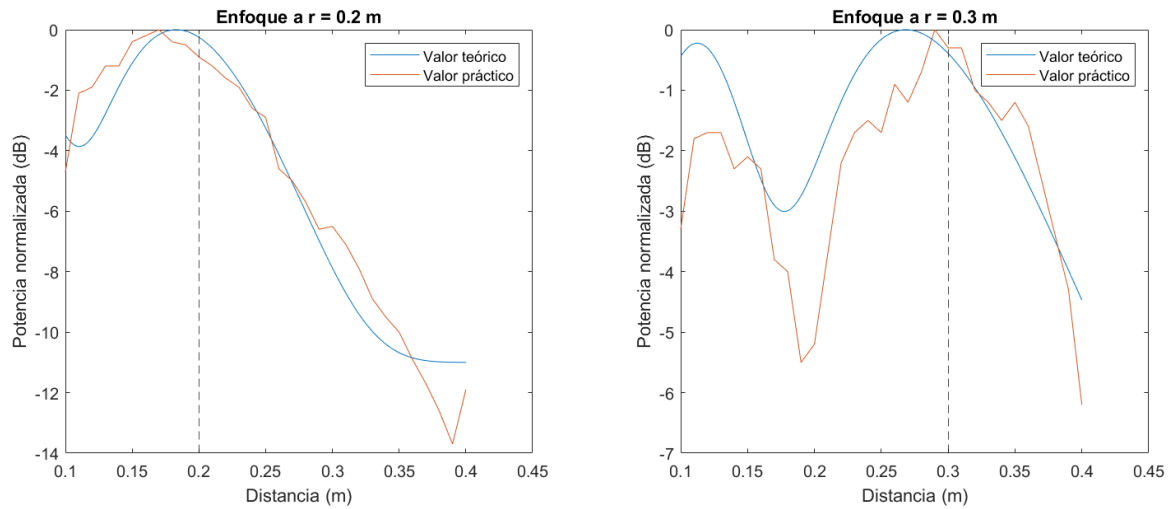


Figura 4.16: Valores de potencia normalizados con apuntado a 30 grados.

Para todos los casos, las medidas siguen una tendencia muy similar a la teórica, con valores muy similares una vez se han normalizado.

#### 4.2.3.- Cambios del ángulo de apuntado

Como se ha visto en la sección anterior, para distancias superiores a aproximadamente 60 cm, el número de elementos utilizado limita la capacidad de enfoque en distancia. No obstante, deberían apreciarse cambios significativos en la energía recibida si se cambia el ángulo de enfoque. Para evaluar esto, se han tomado medidas para varios ángulos, suponiendo una distancia de enfoque de 1.5 m, si bien el impacto de este parámetro no tendrá gran importancia (ver figura 4.10). Para distancias de 1m, 1.5m y 2m, se han tomado medidas a 0 grados, 30 grados, variando el ángulo al que se apunta entre 0, 30 y 60 grados. La configuración para las medidas es idéntica a la de las anteriores secciones. En la tabla 2 se muestran los resultados, normalizados con respecto al máximo valor obtenido. Se puede observar como la potencia es siempre mayor en la dirección de apuntado, como es de esperar.

	A 0°			A 30°		
	1m	1.5 m	2m	1m	1.5 m	2m
Apuntando a 0°	0	-3.1	-6	-17	-24.8	-26.8
Apuntando a 30°	-24.8	-21.9	-18.8	-8.5	-8.8	-15.8
Apuntando a 60°	-14.1	-16	-24.8	-25.8	-11.8	-21.8

Tabla 2: Todos los valores medidos, normalizados con respecto al máximo valor [dB].

En la tabla 3 se muestran los valores teóricos para los puntos estudiados, normalizados con respecto al máximo, y en la tabla 4 se muestra la diferencia entre los valores medidos y los teóricos.

	A 0°			A 30°		
	1m	1.5 m	2m	1m	1.5 m	2m
Apuntando a 0°	-0.2	-3.3	-5.9	-17.6	-22.9	-26.1
Apuntando a 30°	-19.0	-23.0	-25.6	0.0	-3.2	-5.8
Apuntando a 60°	-20.5	-25.5	-28.5	-20.5	-29.8	-36.3

Tabla 3: Valores teóricos en los puntos estudiados, normalizados con respecto al máximo [dB].

	A 0°			A 30°		
	1m	1.5 m	2m	1m	1.5 m	2m
Apuntando a 0°	0.2	0.2	-0.1	0.6	-1.9	-0.7
Apuntando a 30°	-5.8	1.1	6.8	-8.5	-5.6	-10.0
Apuntando a 60°	6.4	9.5	3.7	-5.3	18.0	14.5

Tabla 4: Diferencia entre los valores medidos y los teóricos [dB].

Para el caso en que se apunta a 0 grados, se observa que los valores medidos son muy similares a los teóricos. Sin embargo, para los otros dos casos de enfoque, las diferencias son notables. Esto puede deberse a varios factores que influyen en las medidas, como errores de posicionamiento del receptor, errores en la colocación de los elementos y de posicionamiento del array. El diagrama de radiación de las antenas también podría influir. No obstante, la explicación de las diferencias entre la teoría y la práctica exceden los objetivos de este trabajo. Basta con el hecho de que el array sí es capaz de enfocar la energía en la dirección deseada, aunque no de forma tan eficiente como la teoría predice.

#### 4.2.4.- Array de parches

Como se ilustra en la sección 3.1, la capacidad de enfoque de este array en términos de distancia es considerablemente inferior en comparación con una configuración lineal de 8 elementos. No obstante, el hecho de que el array de parches sea bidimensional introduce un nuevo grado de libertad para el apuntado, a diferencia del caso lineal. Esto se debe a que, en un conjunto unidimensional, la distancia de cada elemento a un punto de enfoque no varía cuando se gira en torno al eje que contiene a los elementos, algo que no sucede en un array de dos dimensiones. Para evaluar las capacidades de enfoque, se ha dispuesto la agrupación tanto en horizontal como en vertical, como se muestra en las figuras 4.17 y 4.18 respectivamente.

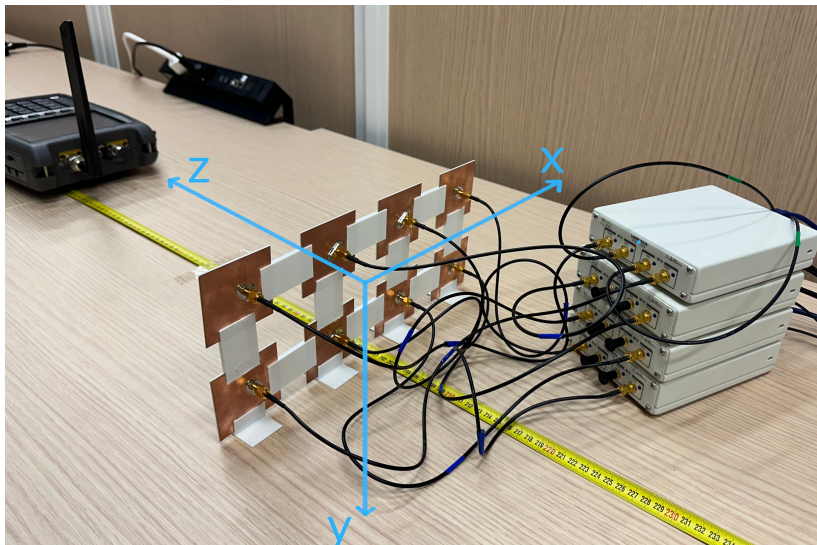


Figura 4.17: Array de parches dispuesto en horizontal.



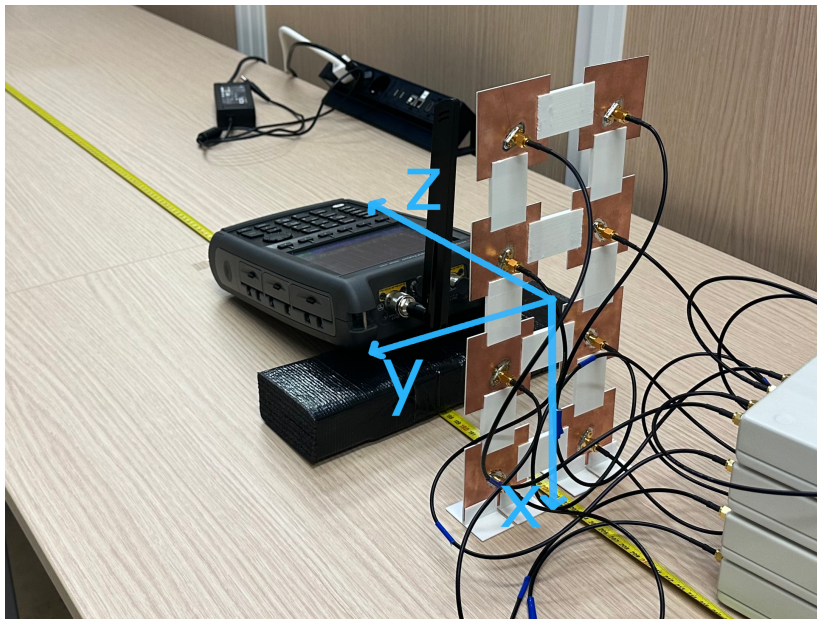


Figura 4.18: Array de parches dispuesto en vertical.

Es importante tener en cuenta la polarización de los parches como elementos individuales para disponerlos de forma que sea la misma que la polarización del monopolo receptor. En la figura 4.19 se muestra la disposición de los parches vistos desde el frontal.

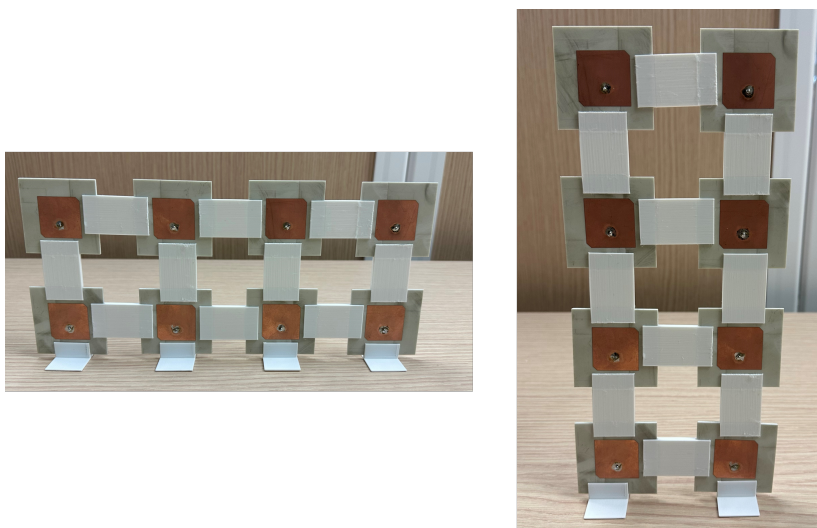


Figura 4.19: Disposición de los parches desde el frente.

En primer lugar, de manera similar a lo hecho con los monopolos, se han tomado medidas para dos puntos de enfoque a diferentes distancias, manteniendo constantes los ángulos

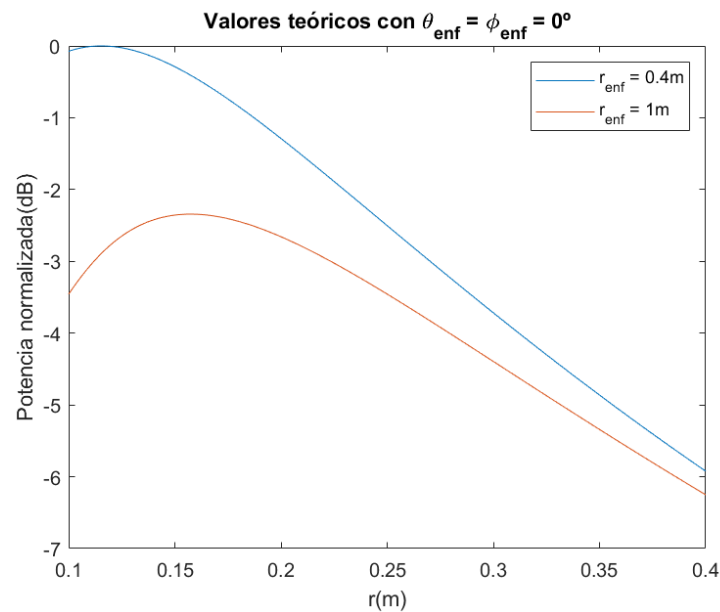


Figura 4.20: Valores teóricos normalizados para los puntos de enfoque ( $r = 0.4m, \theta = 0^\circ, \phi = 0^\circ$ ) y ( $r = 1m, \theta = 0^\circ, \phi = 0^\circ$ ).

de apuntado. Estos puntos son ( $r = 0.4m, \theta = 0^\circ, \phi = 0^\circ$ ) y ( $r = 1m, \theta = 0^\circ, \phi = 0^\circ$ ). Estas medidas se realizaron con el array de parches en posición horizontal. En la figura 4.20 se representan los valores teóricos para estos puntos, normalizados con respecto al máximo total. La elección de estos puntos de enfoque tan distantes respecto a la sección de los monopolos se basa en el gran desplazamiento del foco observado en la sección 3.1, que es mucho mayor que en el caso de una dimensión. En la figura 4.21 se muestran los valores obtenidos, normalizados con respecto al máximo medido.

Se observan claras diferencias, siguiendo las medidas, en una tendencia totalmente distinta a la de los valores teóricos. Estas diferencias pueden deberse a varios factores. En primer lugar, el comportamiento de los parches en campo cercano probablemente sea muy distinto al de los monopolos. Además, la configuración bidimensional hace que las caídas sean mucho menos marcadas para variaciones en la distancia que en el caso unidimensional; en los monopolos, el rango de variación entre 10 y 40 cm era de entre 10 y 12 dB, mientras que aquí es de apenas 5 dB. Por último, aunque la polarización de los parches debiera ser lineal y en el mismo sentido que el monopolo receptor, es posible que el centro de la antena no se haya colocado en su posición ideal, y que el diagrama de radiación de las antenas no sea el



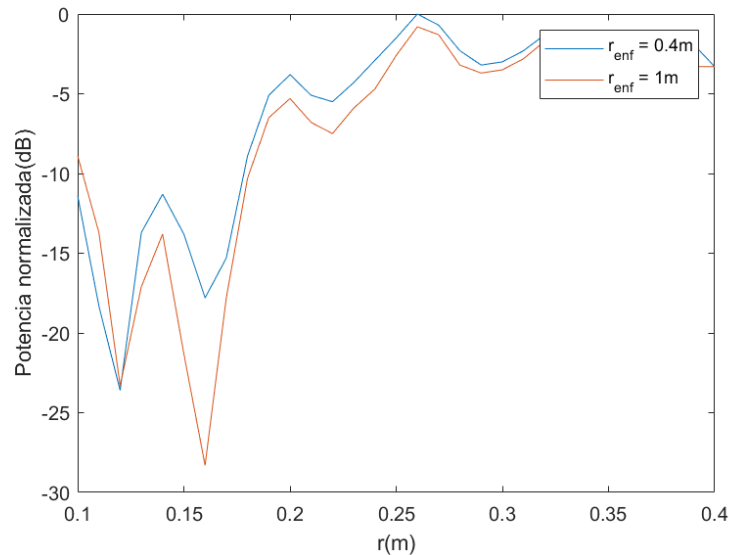


Figura 4.21: Potencia recibida normalizada para los puntos de enfoque ( $r = 0.4m, \theta = 0^\circ, \phi = 0^\circ$ ) y ( $r = 1m, \theta = 0^\circ, \phi = 0^\circ$ ).

ideal.

El único aspecto rescatable es que en las medidas, la potencia para el enfoque a 1 m es inferior a la del enfoque a 0.4 m, y a medida que aumenta la distancia estos valores tienden a igualarse, como predice la teoría.

A la vista de estos resultados, no se han tomado medidas a lo largo de la distancia para el caso en vertical, pues el diagrama de radiación de las antenas está ocultando los efectos del enfoque. En su lugar, para ambos casos se han tomado medidas a 5 distancias diferentes, manteniendo la distancia de enfoque constante y variando el ángulo de apuntado, para permitir evaluar si se producen cambios en la potencia recibida de forma acorde.

Con el array en horizontal, se toman medidas para  $\phi = 0^\circ$  variando el ángulo  $\theta$  entre 0 y 30 grados. De forma idéntica al caso del array de monopolos, se gira el array el ángulo correspondiente, como se muestra en la figura 4.22.

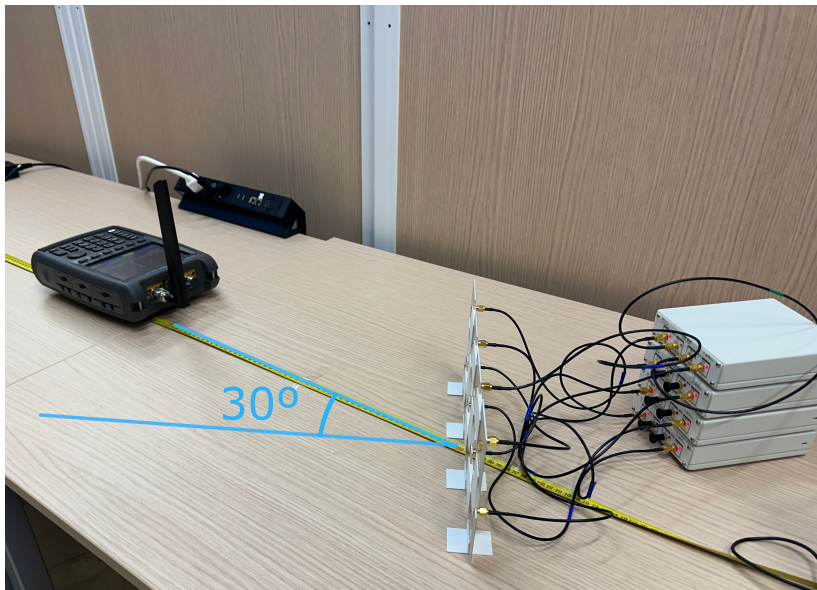


Figura 4.22: Banco de medidas para el array de parches en horizontal con  $\theta_{enf} = 30^\circ$ .

En las tablas 5 y 6 se muestran los valores medidos para ambos casos, normalizados con respecto al máximo del conjunto. Se puede notar como la potencia recibida es, a excepción del caso de 0.125 m, siempre mayor en la dirección de apuntado. Los motivos de esta irregularidad probablemente sean los mismos que en el caso de las medidas a lo largo de la radial; el campo radiado por las antenas parece ser más reactivo que en el caso de los monopolos.

r	0,125m	0,25m	0,5m	1m	1.5m	2m
A 0°	-16.7	0	-1.2	-5	-8	-9.5
A 30 °	-5.8	-10.9	-8.2	-18.5	-22.5	-23.5

Tabla 5: Valores medidos [dB] para el array de parches en horizontal con  $\theta_{enf} = 0^\circ$ ,  $\phi_{enf} = 0$ .

r	0,125m	0,25m	0,5m	1m	1.5m	2m
A 0°	-8	-3.8	-11.6	-20.5	-24.5	-17.5
A 30 °	-3.5	-1.7	-2	-8.1	-9.5	-13.5

Tabla 6: Valores medidos [dB] para el array de parches en horizontal con  $\theta_{enf} = 30^\circ$ ,  $\phi_{enf} = 0$ .

En el segundo caso, se toman medidas para  $\phi = 90$  grados, variando  $\theta_{enf}$  entre 0 y -30 grados. El giro del array se muestra en la figura 4.23.

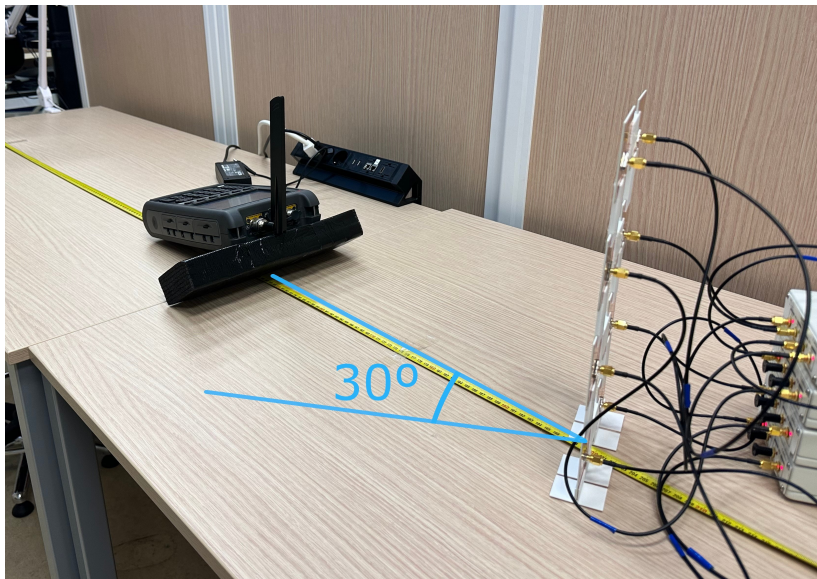


Figura 4.23: Banco de medidas para el array de parches en vertical con  $\theta_{enf} = 30^\circ$ .

Las medidas, normalizadas nuevamente con respecto al máximo del conjunto, se presentan en las tablas 7 y 8. En este caso, la potencia recibida es siempre mayor en la dirección de apuntado. Sin embargo, las diferencias entre la potencia recibida en la dirección de apuntado y en la otra dirección son menores que en el caso anterior. Esto se debe a que el número de elementos a lo largo del eje Y es menor que el número de elementos a lo largo del eje X, lo que limita la capacidad de enfoque en comparación con la disposición horizontal.

r	0,125m	0,25m	0,5m	1m	1.5m	2m
A 0°	0	-3.2	-7.9	-11.9	-13.7	-16.7
A 30 °	-4.5	-8.2	-11.1	-18.7	-19.7	-21.7

Tabla 7: Valores medidos [dB] para el array de parches en vertical con  $\theta_{enf} = 0^\circ$ ,  $\phi_{enf} = 90^\circ$ .

r	0,125m	0,25m	0,5m	1m	1.5m	2m
A 0°	-6.2	-9.9	-14.4	-15.3	-18.2	-24.7
A 30 °	-1.1	-5.3	-8.7	-13.2	-15.2	-22.7

Tabla 8: Valores medidos [dB] para el array de parches en vertical con  $\theta_{enf} = -30^\circ$ ,  $\phi_{enf} = 90^\circ$ .

## 5.- Conclusiones y trabajo futuro

Se presentan a continuación las principales conclusiones extraídas del trabajo presentado en esta memoria, junto con algunas indicaciones sobre posibles líneas de trabajo futuras.

La sincronización constituye el aspecto más destacable del sistema, a pesar de ser algo laborioso de llevar a cabo. Este proceso permite una alimentación uniforme de cada elemento del array de antenas, siempre que se emplee una referencia externa para la sincronización en frecuencia de los dispositivos, en este caso el OctoClock. Esta coordinación es fundamental para aplicaciones de enfoque y conformado de haz, ya que asegura la correcta coordinación de todas las señales emitidas por los elementos del array. Notablemente, esto se logra utilizando dispositivos que, en su diseño original, carecen de capacidades de sincronización entre sí.

En cuanto al método de enfoque, el sistema implementa el método de cálculo de pesos conocido como 'Conjugate Phase' (Fase Conjugada). Las medidas realizadas han demostrado la eficacia del sistema para concentrar la energía en la dirección objetivo, validando así la implementación del método.

No obstante, se han detectado ciertas discrepancias entre las simulaciones y las medidas realizadas en las antenas de parche. Estas diferencias pueden atribuirse a diversos factores, incluyendo imperfecciones en el proceso de fabricación y acoplamientos mutuos no contemplados en las simulaciones, a pesar de que teóricamente los parches no radian en su plano. En contraste, las medidas con antenas omnidireccionales han mostrado una mayor concordancia con los resultados simulados, lo que sugiere que la discrepancia está más relacionada con las características específicas de las antenas de parche que con el sistema de enfoque en sí mismo.

Con respecto a líneas futuras, el método de sincronización desarrollado abre la puerta a la implementación de arrays de cualquier tamaño. En el trabajo se han utilizado cuatro dispositivos idénticos, pero el sistema no está limitado en este sentido. Siempre y cuando se disponga de un dispositivo capaz de usar como señal de referencia la señal generada por el OctoClock, y con los ajustes correspondientes a nivel de software, el sistema podría utilizar

cualquier número de dispositivos.

Otra área de posible ampliación es la implementación de métodos alternativos para el cálculo de los pesos utilizados en el enfoque. Este trabajo se ha centrado en el método de 'Conjugate Phase', pero existen otros métodos que, a diferencia de este, tienen en cuenta la diferencia de posiciones entre antenas.

En su configuración actual, el sistema se limita a sincronizar los dispositivos utilizando un único tono a una frecuencia específica. Esto se debe a que el proceso de sincronización se basa en el retardo de la señal por un número determinado de muestras, valor que depende directamente de la frecuencia del tono y de la tasa de muestreo. La expansión de esta capacidad para sincronizar fases utilizando señales compuestas por múltiples componentes senoidales o señales de banda ancha ampliaría considerablemente el espectro de aplicaciones más allá de la transferencia de energía.

En resumen, el sistema desarrollado ha demostrado ser capaz de sincronizar y enfocar un array de antenas. Aunque se han observado discrepancias en las medidas con antenas de parche, el sistema ha demostrado ser efectivo en la concentración de energía en la dirección objetivo. Además, el sistema es escalable y puede ser ampliado para manejar arrays de cualquier tamaño, así como para implementar métodos de cálculo de pesos alternativos. Por último, la extensión de las capacidades de sincronización para manejar múltiples tonos simultáneamente o señales de banda ancha abriría nuevas posibilidades de aplicación.

# Bibliografía

- [1] «IEEE Standard for Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management,» *IEEE Std 1900.1-2019 (Revision of IEEE Std 1900.1-2008)*, págs. 1-78, abr. de 2019. DOI: [10.1109/IEEESTD.2019.8694195](https://doi.org/10.1109/IEEESTD.2019.8694195). dirección: <https://ieeexplore.ieee.org/document/8694195> (visitado 28-05-2024).
- [2] T. F. Collins, R. Getz, D. Pu y A. M. Wyglinski, *Software-Defined Radio for Engineers* (Artech House Mobile Communications Series). Norwood, MA: Artech House, 2018, 352 págs., ISBN: 978-1-63081-457-1.
- [3] Z. Zhang, H. Pang, A. Georgiadis y C. Cecati, «Wireless Power Transfer—An Overview,» *IEEE Transactions on Industrial Electronics*, vol. 66, n.º 2, págs. 1044-1058, feb. de 2019, ISSN: 1557-9948. DOI: [10.1109/TIE.2018.2835378](https://doi.org/10.1109/TIE.2018.2835378). dirección: <https://ieeexplore.ieee.org/document/8357386> (visitado 29-05-2024).
- [4] J. Machado Fernández, «Software Defined Radio: Basic Principles and Applications,» *Revista Facultad de Ingeniería*, vol. 24, págs. 79-96, 1 de ene. de 2015. DOI: [10.19053/01211129.3160](https://doi.org/10.19053/01211129.3160).
- [5] «Synchronization and MIMO Capability with USRP Devices - Ettus Knowledge Base.» (), dirección: [https://kb.ettus.com/Synchronization\\_and\\_MIMO\\_Capability\\_with\\_USRP\\_Devices](https://kb.ettus.com/Synchronization_and_MIMO_Capability_with_USRP_Devices) (visitado 03-06-2024).
- [6] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3. ed. Hoboken, N.J: Wiley-Interscience, 2005, 1117 págs., ISBN: 978-0-471-66782-7.
- [7] «Near-Field-Focused Microwave Antennas: Near-field shaping and implementation.» (), dirección: <https://ieeexplore.ieee.org/document/7912361/1000> (visitado 18-05-2024).

- [8] «InstallingGR - GNU Radio.» (), dirección: <https://wiki.gnuradio.org/index.php/InstallingGR> (visitado 27-05-2024).



# A.- Código fuente de los bloques personalizados

## A.1.- Bloque Cálculo de desfase

```
import numpy as np
from gnuradio import gr
import pmt

class blk(gr.sync_block):
    def __init__(self, fs=1, freq_serial=1):
        gr.sync_block.__init__(
            self,
            name='Cálculo de desfase',
            in_sig=[np.float32, np.float32],
            out_sig=None
        )
        self.fs = fs
        self.freq_serial = freq_serial
        self.samples_per_period = self.fs // self.freq_serial
        self.in0_buffer = np.array([])
        self.in1_buffer = np.array([])
        self.USRP_number = 3

        self.portNamesOut = [f'delayOut{i}' for i in range(1, self.
USRP_number + 1)]
        self.portNameSync = 'sync'
        self.portNameFreqSerial = 'freq_serial'
        for name in self.portNamesOut:
            self.message_port_register_out(pmt.intern(name))

        self.message_port_register_in(pmt.intern(self.portNameSync))
```

```
self.message_port_register_in(pmt.intern(self.portNameFreqSerial)
)

self.set_msg_handler(pmt.intern(self.portNameSync), self.
handle_sync_msg)

self.set_msg_handler(pmt.intern(self.portNameFreqSerial), self.
handle_freq_serial_msg)

print(f"Frecuencia de muestreo: {self.fs} Hz.\nFrecuencia de la
señal {self.freq_serial} Hz.\n")

self.key_map = {f'sync{i}': pmt.intern(self.portNamesOut[i]) for
i in range(0, self.USRP_number )}

def handle_freq_serial_msg(self, msg):
    if pmt.is_pair(msg):
        key = pmt.to_python(pmt.car(msg))
        value = pmt.to_python(pmt.cdr(msg))
        if key == "freq_serial":
            self.freq_serial = value
            self.samples_per_period = self.fs // self.freq_serial
            print(f"Cambio a frecuencia: {self.freq_serial} Hz\
nMuestras por periodo: {self.samples_per_period}.\n")

        for key in self.key_map:
            self.message_port_pub(self.key_map[key], pmt.to_pmt
(0))

            print(f"Delay configurado a '0' en el puerto '{self.
key_map[key]}' .")

def handle_sync_msg(self, msg):
    if pmt.is_pair(msg):
        key = pmt.to_python(pmt.car(msg))
        value = pmt.to_python(pmt.cdr(msg))
```

```
        if key == "sync":
            self.process_sync(value)

    def process_sync(self, sync_number):
        sync_type = f"sync{sync_number}"
        if len(self.in0_buffer) > 0 and len(self.in1_buffer) > 0:
            delay_estimate = self.estimate_delay(self.in0_buffer, self.
in1_buffer)
            delay_estimate = int(delay_estimate % self.samples_per_period
)
            PMT_msg = pmt.cons(pmt.to_pmt(sync_type), pmt.to_pmt(
delay_estimate))
            port_to_publish = self.key_map[sync_type]
            self.message_port_pub(port_to_publish, PMT_msg)
            print(f"Publicado mensaje '{PMT_msg}' en el puerto '{
port_to_publish}.\n")

    def estimate_delay(self, signal1, signal2):
        correlation = np.correlate(signal1, signal2, 'full')
        delay = np.argmax(correlation) - len(signal2) + 1
        print(f"Retraso estimado: {delay} muestras.\n")
        return delay

    def work(self, input_items, output_items):
        in0, in1 = input_items
        self.in0_buffer = np.concatenate((self.in0_buffer, in0))
        self.in1_buffer = np.concatenate((self.in1_buffer, in1))
        if len(self.in0_buffer) > 1024 * 8:
            self.in0_buffer = np.array([])
            self.in1_buffer = np.array([])

        return len(in0)
```

## A.2.- Bloque Conformador pesos

```
import numpy as np
import pmt
from gnuradio import gr

class blk(gr.sync_block):
    def __init__(self, rf_freq=1):
        gr.sync_block.__init__(self,
                                name='Cálculo de pesos',
                                in_sig=[],
                                out_sig=[])

        self.port_name = "in"
        self.message_port_register_in(pmt.intern(self.port_name))
        self.set_msg_handler(pmt.intern(self.port_name), self.handle_msg)

        self.N = 4 # Número de elementos a lo largo del eje x
        self.M = 2 # Número de elementos a lo largo del eje y
        self.d = 0.7 # Separación entre elementos en lambdas
        self.portNameOut = [f"Ángulo {i+1}" for i in range(self.N*self.M)
                             ]

        for name in reversed(self.portNameOut):
            self.message_port_register_out(pmt.intern(name))

        self.RF_FREQ = rf_freq
        self.lamb = 3e8/self.RF_FREQ
        self.elem_pos = self.lamb * self.genera_pos_el(self.N, self.M, self
        .d)

        print(f"Elementos en posiciones: {self.elem_pos}")

        self.w = np.zeros(len(self.elem_pos), dtype=np.complex64)
        self.last_phi = 0
        self.last_theta = 0
        self.last_r = 0.5
```

```
def genera_pos_el(self, N, M, d):
    xStart = -d * (N - 1) / 2
    xEnd = d * (N - 1) / 2
    yStart = -d * (M - 1) / 2
    yEnd = d * (M - 1) / 2

    xValues = np.linspace(xStart, xEnd, N)
    yValues = np.linspace(yStart, yEnd, M)
    zValue = 0

    xGrid, yGrid = np.meshgrid(xValues, yValues, indexing='ij')

    zGrid = np.full(xGrid.shape, zValue)

    posiciones = np.column_stack((xGrid.ravel(), yGrid.ravel(), zGrid
    .ravel()))
    return posiciones

def handle_msg(self, msg):
    if pmt.is_pair(msg):
        key = pmt.to_python(pmt.car(msg))
        if key == "phi":
            if pmt.is_real(pmt.cdr(msg)):
                phi = pmt.to_python(pmt.cdr(msg))
                self.last_phi = phi
                theta = self.last_theta
                r=self.last_r

        elif key == "theta":
            if pmt.is_real(pmt.cdr(msg)):
                theta = pmt.to_python(pmt.cdr(msg))
                self.last_theta = theta
                phi= self.last_phi
                r=self.last_r
```

```
elif key == "distance":
    if pmt.is_real(pmt.cdr(msg)):
        r = pmt.to_python(pmt.cdr(msg))
        phi = self.last_phi
        theta = self.last_theta
        self.last_r = r

    print(f"Apuntando a r = {r}, phi = {phi*180/np.pi}, theta = {
theta*180/np.pi}")
    x_target = r*np.sin(theta)*np.cos(phi)
    y_target = r*np.sin(theta)*np.sin(phi)
    z_target = r*np.cos(theta)
    for i,el_pos in enumerate(self.elem_pos):
        dist = np.sqrt((el_pos[0]-x_target)**2 +(el_pos[1]-
y_target)**2 +(el_pos[2]-z_target)**2)
        phase = np.pi*2*dist/self.lamb
        self.w[i] = np.complex64(np.cos(phase) + np.sin(phase)*1j
)

    for i,port_name in enumerate(self.portNameOut):
        angle = np.angle(self.w[i])+2*np.pi
        self.message_port_pub(pmt.intern(port_name) , pmt.to_pmt(
angle))

        print(f"Publicado ángulo {angle} [rad] en el puerto {
port_name}.\n")

def work(self, input_items, output_items):
    output_items[0][:] = input_items[0]
    return len(output_items[0])
```