

# Trabajo Fin de Grado

## Grado en Ingeniería de las Tecnologías de Telecomunicación

### Identificación de las medidas sin línea de visión directa en un sistema UWB de localización

Autor: Noel Cortés García

Tutora: Eva Arias de Reyna Domínguez

**Dpto. Teoría de la Señal y Comunicaciones**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2019





Trabajo Fin de Grado  
Grado en Ingeniería de las Tecnologías de Telecomunicación

# **Identificación de las medidas sin línea de visión directa en un sistema UWB de localización**

Autor:

Noel Cortés García

Tutora:

Eva Arias de Reyna Domínguez

Profesora Titular de Universidad

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019



Proyecto Fin de Carrera: Identificación de las medidas sin línea de visión directa en un sistema UWB de localización

Autor: Noel Cortés García

Tutora: Eva Arias de Reyna Domínguez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal



*A mi familia*

*A mis maestros*





# Agradecimientos

---

En primer lugar, agradecer especialmente la labor de Dña. Eva Arias de Reyna Domínguez por confiarme este proyecto y prestarme su tiempo para guiarme a llevarlo a buen término, además de proveerme del material de laboratorio y la bibliografía necesarios para realizarlo.

Al departamento de teoría de la señal y comunicaciones, por permitirme realizar el conjunto de las medidas de este trabajo en sus instalaciones.

A todos mis compañeros de universidad, con los que he compartido infinidad de horas, en clase o estudiando, y en especial a Pablo y a Alba, con los que he pasado varios años compartiendo piso en Sevilla.

A mi madre y a Pepe, que gracias a sus esfuerzos he conseguido llegar hasta aquí y poder llegar a redactar este proyecto.

A mi novia Lorena, por su paciencia y sus ánimos para seguir adelante.

*Sevilla, 2019*



# Resumen

---

El objetivo de este proyecto es el de lograr una identificación de señales UWB (Ultra Wide Band) de forma que se pueda hacer una distinción entre señales, con línea de visión directa o sin ella, con una determinada probabilidad de éxito.

Para poder llevar a cabo esto, vamos a servirnos del kit EVK1000 del fabricante DecaWave.

Este trabajo se puede dividir en dos partes: primero una parte teórica y de introducción para después entrar en la parte experimental, siendo ésta última la más importante y la que es el grueso del proyecto.

En el apartado teórico se expone por qué se escoge la tecnología UWB para la localización en interiores. Después, se exponen teóricamente los dos procedimientos utilizados para intentar discernir si la señal tiene línea de visión directa o no.

En el apartado experimental, se realizan mediciones en primer lugar en distintos lugares de la Escuela Técnica Superior de Ingeniería de tal forma que tengamos muestras con línea o sin línea de visión directa. Una vez tomadas esas medidas, se plantean los dos procedimientos de identificación que se van a usar, que son el ratio potencia del primer camino frente a potencia total y el RMS Delay Spread, para definir un umbral de decisión para cada procedimiento.

Una vez obtenidos los umbrales, se realizan mediciones en otros lugares, de forma que este segundo conjunto de medidas sea totalmente independiente del anterior. Con estas medidas, y gracias a los procedimientos nombrados anteriormente, se ponen a prueba los procedimientos obtenidos anteriormente. Además, se realizará el cálculo de la probabilidad de error de ambos procedimientos, ya que se sabe a priori si las medidas son LOS o NLOS, para comprobar el grado de eficacia de dichos procedimientos.

Para tratar los datos de las mediciones se exportan los logs que crea el kit de DecaWave a MatLab/Octave, donde según el procedimiento de identificación se procesarán o los datos de potencia que el kit proporciona o la gráfica de la señal.



# Abstract

---

The objective of this project is to achieve an identification of UWB signals so that a distinction can be made between signals, with or without direct line of sight, with a certain probability of success.

In order to carry out this, we are going to use the EVK1000 kit from the manufacturer DecaWave.

This work can be divided into two parts: first a theoretical and introductory part to later enter the experimental part, the latter being the most important and the one that is the bulk of the project.

In the theoretical section, it is explained why UWB technology is chosen for indoor location. Then, the two procedures used to try to discern whether the signal has direct line of sight or not are theoretically presented.

In the experimental section, measurements are taken first in different places of the Higher Technical School of Engineering in such a way that we have samples with or without line of direct vision. Once these measures are taken, the two identification procedures to be used are outlined, which are the ratio of the power of the first path versus the total power and the RMS Delay Spread, to define a decision threshold for each procedure.

Once the thresholds are obtained, measurements are made in other places, so that this second set of measurements is completely independent of the previous one. With these measurements, and thanks to the procedures named above, they will be compared with those previously obtained. In addition, the calculation of the probability of error of both procedures will be carried out, since it is known a priori whether the measurements are LOS or NLOS, to verify the degree of effectiveness of said procedures.

To process the measurement data, the logs created by the DecaWave kit are exported to MatLab/Octave, where according to the identification procedure, the power data that the kit provides or the graph of the signal will be processed.

# Índice

---

<b>Agradecimientos</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xiv</b>
<b>Índice de Tablas</b>	<b>xvi</b>
<b>Índice de Figuras</b>	<b>xviii</b>
<b>Notación</b>	<b>xxi</b>
<b>1 Introducción</b>	<b>1</b>
1.1. <i>Objetivos</i>	1
1.2. <i>Estructura</i>	2
<b>2 Conceptos teóricos</b>	<b>5</b>
2.1. <i>UWB</i>	5
2.2. <i>Ratio potencia primer camino frente a potencia total</i>	7
2.3. <i>RMS Delay Spread</i>	8
<b>3 Entorno de trabajo</b>	<b>12</b>
3.1. <i>Introducción</i>	12
3.2. <i>Equipos</i>	12
3.2.1. <i>Transmisores</i>	14
3.3. <i>Funcionamiento de los equipos</i>	18
3.4. <i>Configuración de parámetros</i>	19
<b>3 Medición experimental</b>	<b>21</b>
4.1. <i>Toma de medidas</i>	21
4.1.1. <i>Medidas para la obtención del umbral</i>	21
4.1.2. <i>Medidas para poner a prueba el umbral</i>	22
4.2. <i>Casos particulares</i>	22
4.2.1. <i>Medidas corruptas</i>	24
4.2.2. <i>Caso de la pared</i>	25
4.2.3. <i>Caso de la puerta</i>	26
<b>5 Ratio potencia primer camino frente a potencia total</b>	<b>29</b>
5.1. <i>Obtención de FSL y RSL a partir de los logs</i>	29
5.2. <i>Cálculo del ratio</i>	32
5.3. <i>Comparativa de resultados</i>	35
<b>6 RMS Delay Spread</b>	<b>39</b>
6.1. <i>Cálculo del máximo de la señal</i>	41
6.2. <i>Algoritmo RMS Delay Spread</i>	41
6.2.1. <i>Algoritmo fin de señal</i>	42
6.2.2. <i>Ecuación RMS Delay Spread</i>	42
6.3. <i>Representación de resultados</i>	42

<b>7</b>	<b>Cálculo de la probabilidad de error</b>	<b>46</b>
	7.1. <i>Probabilidad de error para el ratio potencia primer camino frente a potencia total</i>	46
	7.2. <i>Probabilidad de error para el RMS Delay Spread</i>	47
<b>8</b>	<b>Conclusiones</b>	<b>50</b>
<b>9</b>	<b>Apéndice A: Instalación y uso del Software</b>	<b>53</b>
	9.1. <i>Instalación</i>	53
	9.2. <i>Uso</i>	54
	9.2.1. Logs	56
	9.2.2. Exportar datos con MatLab/Octave	59
<b>10</b>	<b>Apéndice B: ficheros ShellScript</b>	<b>62</b>
	10.1. <i>Ratio potencia primer camino frente a potencia total</i>	62
	10.1.1. buscaRSL.sh	62
	10.1.2. buscaFSL.sh	62
	10.2. <i>RMS Delay Spread</i>	63
	10.2.1. buscagrafica.sh	63
<b>11</b>	<b>Apéndice C: ficheros de MatLab/Octave</b>	<b>65</b>
	11.1. <i>Ratio potencia primer camino frente a potencia total</i>	65
	11.1.1. Ratio.m	65
	11.2. <i>RMS Delay Spread</i>	75
	11.2.1. Identifica.m	75
	<b>Referencias</b>	<b>84</b>

# Índice de Tablas

---

Tabla 3-1: Configuración de ajuste del kit de Evaluación.

Tabla 3-2: Modos de operación del kit de Evaluación.





# Índice de Figuras

---

Figura 2-1: Ejemplo de distribución de tres casos en el ratio.

Figura 2-2: Ejemplo de gráfica de RMS Delay Spread.

Figura 3-1: Caja contenedora del kit de evaluación EVK1000 de DecaWave.

Figura 3-2: Placa kit de Evaluación EVK1000 (ancla).

Figura 3-3: Placa kit de Evaluación EVK1000 (tag).

Figura 4-1: Ejemplo de señal LOS

Figura 4-2: Ejemplo de señal NLOS.

Figura 4-3: Ejemplo de medida corrupta.

Figura 4-4: Ejemplo de medida NLOS con pared como obstáculo.

Figura 4-5: Ejemplo de medida NLOS con puerta como obstáculo.

Figura 5-1: Contenido del fichero de salida de *buscaFSL.sh*.

Figura 5-2: Contenido del fichero de salida de *buscaRSL.sh*.

Figura 5-3: Código para exportar los datos de los ficheros .txt a MatLab/Octave.

Figura 5-4: Código para exportar los datos de los ficheros .txt a MatLab/Octave.

Figura 5-5: Código que calcula el ratio de las medidas tomadas para calcular el umbral.

Figura 5-6: Código que calcula el ratio de las medidas tomadas para poner a prueba el umbral.

Figura 5-7: Histograma de los ratios de las medidas para calcular el umbral.

Figura 5-8: Histograma de los ratios de las medidas para poner a prueba el umbral.

Figura 6-1: Ejemplo de gráfica de una medida.

Figura 6-2: Extracto de código de búsqueda del pico máximo.

Figura 6-3: Histograma del RMS Delay Spread de las medidas para calcular el umbral.

Figura 6-4: Histograma del RMS Delay Spread de las medidas para poner a prueba el umbral.

Figura 7-1: Extracto del código para la probabilidad de error del ratio.

Figura 7-2: Extracto del código para la probabilidad de error del RMS Delay Spread.

Figura 9-1: Página web del software del driver.

Figura 9-2: Configuración del canal de comunicación.

Figura 9-3: pantalla del software DecaWave con la opción de visualizar gráfica.

Figura 9-4: Ejemplo de trama recogido en el *Log Channel Responses*.

Figura 9-5: Ejemplo de respuesta impulsiva en el *Log Channel Responses*.

Figura 9-6: Ejemplo de vuelco de datos en MatLab/Octave.

Figura 9-7: Ejemplo de fichero de salida de script.



# Notación

---

RMS	Root Mean Square
UWB	Ultra Wide Band
GPS	Global Position System
AoA	Ángulo de llegada
ToA	Tiempo de llegada
TDoA	Diferencia de tiempo de llegada
FCC	Federal Communication Commission
B	Ancho de banda
$f_u$	Frecuencia superior
$f_l$	Frecuencia inferior
$f_o$	Frecuencia central
WUSB	Wireless Universal Serial Bus
WPAN	Wireless Personal Area Network
LOS	Line Of Sight
NLOS	Non Light Of Sight
NLOS/DDP	Non Light Of Sight/Detected Direct Path
NLOS/UDP	Non Light Of Sight/Undetected Direct Path
dBm	Decibelios (mW)
:	Tal que
<	Menor o igual
>	Mayor o igual



# 1 INTRODUCCIÓN

---

Las comunicaciones inalámbricas me han llamado la atención desde el Bachillerato. Siempre me ha gustado la física y, cuando todo el mundo empezó a tener WiFi en sus casas, luego a conexión a Internet en los teléfonos móviles, etc. me empezó a llamar la atención el saber cómo funcionaba todo eso de verdad, qué había detrás de todo eso. Y por eso empecé a estudiar Ingeniería de Telecomunicaciones.

Luego, en tercero, la profesora Eva Arias de Reyna nos enseñó en clase un sistema de localización en interiores, pensé que eso era una idea con una utilidad increíble. Y cuando en cuarto vi que ofrecía un Trabajo de Fin de Grado sobre ello no lo dudé.

Es cierto que para localizar cosas, un móvil o un coche por ejemplo, ya existen distintas tecnologías como GPS o GLONASS, pero dichas tecnologías no ofrecen resultados nada aceptables cuando se trata de localizar un objeto en el interior de un edificio. Por eso, se estudian otras opciones.

En otras asignaturas estudié acerca de Ultra Wide Band (UWB sus siglas en inglés), conocí ventajas e inconvenientes de esta tecnología y la verdad es que en el tema que nos ocupa en este proyecto, es la tecnología que mejor cumple los requisitos para un sistema de estas características.

Hay varias técnicas para estimar a qué distancia se encuentra un objeto, algunas de ellas son según el ángulo de llegada (AoA), según el tiempo de llegada (ToA) o según el nivel de potencia (RSS) de la señal recibida. Con el kit que utilizaremos en este proyecto, la técnica utilizada será la del tiempo de llegada (ToA).

En este proyecto utilizaremos las técnicas de RMS Delay Spread y el ratio potencia primer camino frente a potencia total para lograr identificar si una señal tiene línea de visión directa (LOS) o no tiene línea de visión directa (NLOS).

## 1.1 Objetivos

- Fundamentos teóricos de la tecnología UWB y de las técnicas RMS Delay Spread y el ratio potencia primer camino frente a potencia total.
- Funcionamiento básico y comunicación de comunicaciones UWB usando el kit de DecaWave EVK1000 y el software Decarang, proporcionado igualmente por DecaWave.
- Toma de medidas en múltiples puntos para formar una base de datos suficientemente grande.

- Para completar el planteamiento de las dos técnicas de identificación mencionadas, aplicarlas a la base de datos de medidas obtenida anteriormente y extraer un umbral de decisión LOS/NLOS con cada una de ellas.
- Cálculo de la probabilidad de error de las dos técnicas de identificación establecidas, mediante un conjunto de medidas diferente del utilizado para decidir los umbrales, teniendo en cuenta que a priori sabemos si las medidas realizadas eran con línea de visión directa o sin ella.
- Una vez tenido todo lo anterior en cuenta, evaluar cuál el grado de idoneidad de cada una de las dos técnicas para el objetivo del proyecto.

## 1.2 Estructura

La organización del presente documento consta de 11 capítulos:

- Primer capítulo: Breve introducción del presente proyecto, junto con la motivación, objetivos y estructura del mismo.
- Segundo capítulo: Capítulo teórico donde se tratan las características tanto de la tecnología UWB como de las técnicas que se utilizan para la identificación LOS/NLOS.
- Tercer capítulo: En este capítulo se explica todo lo relacionado con el entorno de trabajo del proyecto.
- Cuarto capítulo: En el cuarto capítulo se explican la serie de medidas que se han tomado, si estas son LOS o NLOS y se detallan los casos particulares que se han ido encontrando a medida que se realizaba este proyecto.
- Quinto capítulo: Se explica detalladamente en este capítulo el funcionamiento del programa que calcula el ratio potencia primer camino frente a potencia total de las medidas tomadas con anterioridad, proponiendo un valor adecuado de umbral para la identificación. Se ilustran también los resultados obtenidos a raíz de su ejecución.
- Sexto capítulo: En este sexto capítulo se muestra cómo funciona el programa que calcula el RMS Delay Spread de las medidas obtenidas. Como en el apartado anterior, en este también se propone un valor adecuado de umbral de identificación. Se muestran los resultados conseguidos a través de la ejecución del código.
- Séptimo capítulo: En el séptimo capítulo se aplican las dos técnicas empleadas en la identificación LOS/NLOS de un nuevo conjunto de medidas y se realiza el cálculo de la probabilidad de error de cada una de las técnicas.



- Octavo capítulo: Capítulo dedicado a los resultados y conclusiones del proyecto.
- Apéndice A: Se resume el funcionamiento del software Decaranging, de DecaWave, utilizado en este proyecto para obtener los logs.
- Apéndice B: Se encuentran todos los códigos ShellScript utilizados en este proyecto.
- Apéndice C: Se encuentran todos los códigos MatLab/Octave utilizados en este proyecto.



## 2 CONCEPTOS TEÓRICOS

---

La teoría que se desarrolla en este punto se divide en apartados claramente diferenciados:

- Un primer apartado en el que se explican las características de la tecnología UWB y lo que la hace ser la elegida para un sistema de localización.
- En el segundo apartado se expone el contenido teórico que lleva detrás la técnica del ratio potencia del primer camino frente a la potencia total.
- Para finalizar este punto puramente teórico, se desarrolla brevemente cómo funciona la técnica de identificación RMS Delay Spread.

### 2.1 UWB

Para explicar teóricamente la tecnología UWB se han hecho uso de las siguientes fuentes bibliográficas: [1] [10][11] [12].

Una señal UWB se define mediante dos criterios. Si una señal cumple al menos uno de ellos, la señal se considera una señal UWB. Una señal UWB se define como **cualquier transmisión de radio con un espectro de frecuencia mayor al 20% de la frecuencia central o un ancho de banda mayor que 500MHz**, según la FCC [11] (Federal Communications Commission).

Explicado de forma matemática se tiene que [12]:

- El ancho de banda ( $B/f_o$ ) entre dos puntos a 10 dB tiene que ser  $\geq 20\%$ . Esto quiere decir que la densidad espectral de potencia de la señal está a 10dB respecto del máximo.

$$\frac{B}{f_o} = 2 * \frac{f_u - f_l}{f_u + f_l} * 100 \geq 20\%$$

Ecuación 1. Ancho de banda

- El ancho de banda absoluto  $B \geq 500\text{MHz}$ .

$$B = f_u - f_l > 500 \text{ MHz}$$

Ecuación 2. Ancho de banda absoluto

Para el objetivo del proyecto, que es la identificación de medidas en un sistema de localización, esta tecnología presenta algunas ventajas como:

- Bajo margen de error de posición: la precisión con esta tecnología es de centímetros, al menos en condiciones de visión directa, por eso es tan importante la identificación de las medidas NLOS.
- Bajo consumo.
- Alta inmunidad al multirayecto, uno de los grandes atractivos de esta tecnología en los sistemas de localización en interiores [10].
- Buena capacidad de penetración, debida a que al tener un gran ancho de banda, alguna de las frecuencias que componen la señal pueden superar o esquivar obstáculos.
- Reutilización del espectro radioeléctrico: las señales UWB, al tener un espectro tan amplio ocupan frecuencias utilizadas para otros fines. Sin embargo, al ser una señal de muy baja potencia la interferencia que causa es despreciable.

Aunque UWB también tiene desventajas como:

- Poco alcance: al ser una señal de muy baja potencia, no se puede implementar un sistema de largo alcance.
- Interoperabilidad.
- Una solución general tiene actualmente un coste alto.

A modo de curiosidad, otros ejemplos de aplicaciones de la tecnología UWB pueden ser WUSB (Wireless Universal Serial Bus), aplicaciones de streaming de vídeo o entornos WPAN [1].

En los apartados siguientes se tratarán dos técnicas diferentes para identificar si la señales recibidas son LOS o NLOS. La identificación de medidas es un tema complejo ya que hay muchas variables que afectan a las señales electromagnéticas durante su propagación, teniendo esto como resultado un error en la medida de distancia que se intenta estimar. Esto es algo no asumible si queremos un sistema que funcione a una precisión de cm.

El objetivo de este proyecto es hacer una diferenciación LOS/NLOS aunque, a modo de ilustración teórica, se describen los dos casos en los que las señales NLOS se pueden diferenciar: NLOS con camino directo detectado (NLOS/DDP) y NLOS con camino directo no detectado (NLOS/UDP).

El caso NLOS/DDP es un caso NLOS que podría solventarse, una vez identificado correctamente, mediante alguna estimación que corrija el error que existe al haber atravesado la señal un obstáculo.

En cambio, en el caso NLOS/UDP el rayo directo no llega al receptor para poder establecer una medida de distancia a partir de él. Por lo tanto, es una medida de distancia no válida.

Como la identificación de señales en LOS/NLOS es compleja de por sí, se ha estimado oportuno el realizar dicha identificación a lo largo del proyecto.

## 2.2 Ratio potencia primer camino frente a potencia total

Para este apartado se aportan las bibliografías siguientes: [6] [7] [9]:

Esta técnica estudia la relación existente entre la potencia del primer camino, es decir, del rayo que el algoritmo interpreta como rayo directo entre transmisor y receptor, y la potencia total recibida. En señales LOS, la potencia del primer camino será cercana a la potencia total recibida (valor de ratio cercano a 1), mientras que en señales NLOS, el primer camino diferirá más de la potencia total.

En la figura 2-1, se expone un ejemplo de gráfico de un experimento con medidas con el cálculo del ratio [9]:

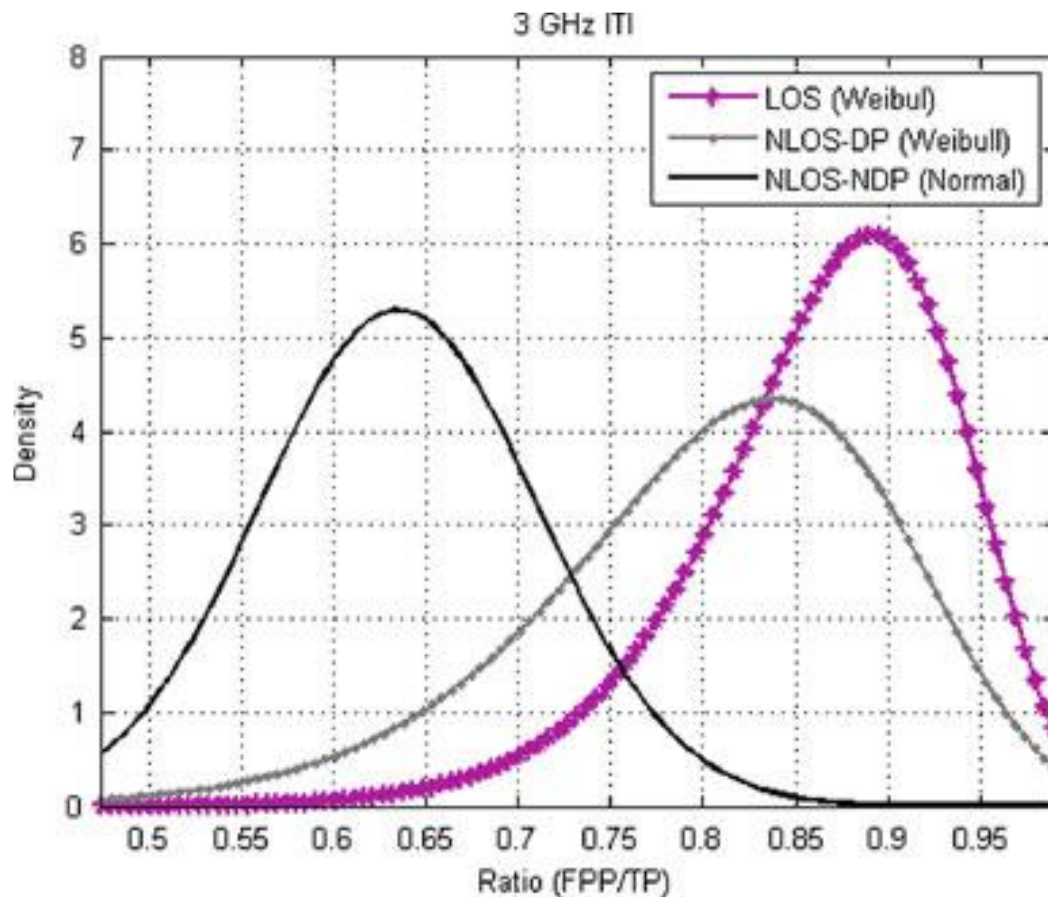


Figura 2-1: Ejemplo de distribución de tres casos en el ratio.

En este ejemplo se distinguen tres posibles casos:

- Curva LOS: aquellas medidas que tienen línea de visión directa.
- Curva NLOS/DDP (en la figura, NLOS-DP): aquellas medidas sin línea de visión directa pero en las que el rayo directo consigue llegar al receptor.
- Curva NLOS/UDP (en la figura, NLOS-NDP): aquellas medidas sin línea de visión directa en las cuales no se detecta el rayo directo.

Además, en esta gráfica se observa que las curvas LOS y NLOS/DDP siguen una distribución Weibull, mientras que la curva NLOS/UDP sigue una distribución normal [6].

En este proyecto solo consideraremos dos posibles casos, o bien la señal es LOS o bien NLOS.

Para realizar el cálculo de ratio, se requiere de la ecuación expresada aquí:

$$\rho = \frac{P_{FP}}{P_T} = \frac{|\alpha_1|^2}{\sum_{k=1}^{L_p} |\alpha_k|^2}$$

Ecuación 3: Ecuación ratio potencia primer camino frente a potencia total.

En esta ecuación,  $L_p$  representa la duración temporal del pulso y  $\alpha_k$  la amplitud del pulso en el instante  $k$ . Estos parámetros se rescatan de la ecuación que modela el canal multitrayecto, siendo esta la siguiente [9]:

$$h(t) = \sum_{k=1}^{L_p} \alpha_k * e^{j\varphi_k} * \delta(t - \tau_k)$$

Ecuación 4: Ecuación modelo canal multitrayecto.

## 2.3 RMS Delay Spread

Para el último apartado de teoría que recoge este documento, se sigue la siguiente bibliografía: [8] [9].

Esta técnica recoge una medida de cuánto se ha ensanchado una señal, en media cuadrática, a lo largo del camino que ha recorrido entre transmisor y receptor. La ecuación que describe esto matemáticamente es la que sigue [9]:

$$\tau_{rms} = \sqrt{\frac{\sum_{k=1}^{L_p} \alpha_k^2 * \tau_k^2}{\sum_{k=1}^{L_p} \alpha_k^2} - \tau_m^2}$$

Ecuación 5: Ecuación RMS Delay Spread.

Siendo  $\tau_m$  la expresión de la ecuación 6:

$$\tau_m = \frac{\sum_k \alpha_k^2 * \tau_k}{\sum_k \alpha_k^2}$$

Ecuación 6: Ecuación de  $\tau_m$ .

Para decidir si una señal es LOS o NLOS con este método, se requiere de un umbral de decisión:

$$H_0: \tau_{rms} > \tau_{th} \rightarrow NLOS$$

$$H_1: \tau_{rms} < \tau_{th} \rightarrow LOS$$

Ecuación 7: Umbrales de decisión para el RMS Delay Spread.

Por tanto, esta es una operación bastante más compleja de realizar computacionalmente que la ecuación del ratio que se presentó en el punto anterior, algo a tener en cuenta a la hora de decantarse por una u otra técnica.

Para poder observar de manera gráfica lo que mide la técnica del RMS Delay Spread, se cierra el apartado presentando la figura 2-2, donde se ilustra un ejemplo [8]:

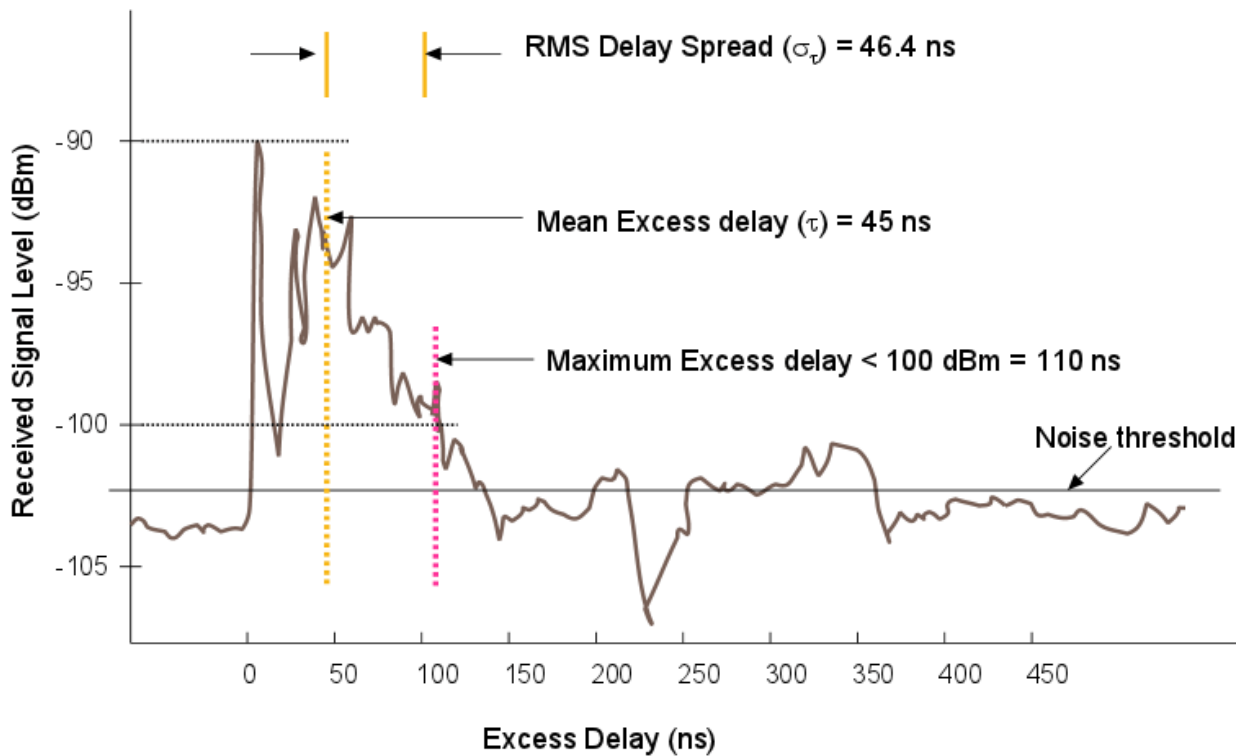


Figura 2-2: Ejemplo de gráfica de RMS Delay Spread.





## 3 ENTORNO DE TRABAJO

---

### 3.1 Introducción

En este apartado se describe el entorno de investigación en el cual se ha realizado este proyecto. Se han utilizado las siguientes referencias para el desarrollo de este apartado: [2] [3] [4] [5] [12].

### 3.2 Equipos

Para llevar a cabo este proyecto, se necesita contar con el kit EVK1000 Evaluation Kit para realizar las medidas oportunas. Este kit de evaluación incluye:

- 2 placas EVB1000.
- 2 antenas.
- 2 cables USB 2.0.
- Guía de inicio rápido.

La figura 3-1 ilustra la caja donde viene guardado el kit de evaluación:



Figura 3-1: Caja contenedora del kit de evaluación EVK1000 de DecaWave.

La placa de evaluación EVB1000 tiene unas medidas de 7 x 7cm. El frontal de la placa dispone de una pantalla LCD, donde se muestra información de las medidas de distancia, y también dispone de un interruptor (S1) que controla el modo de operación.

La parte trasera contiene el DW1000 IC, el ARM IC, el botón de ARM reset, dos interruptores DIP (S2 y S3), el conector JTAG, la conexión externa SPI y varios conectores y jumpers para configurar el modo de operación.

### 3.2.1 Transmisores

Este Kit incluye dos placas idénticas EVB1000, las cuales se ilustran en las figuras 3-2 y 3-3:



Figura 3-2: Placa kit de Evaluación EVK1000 (ancla).

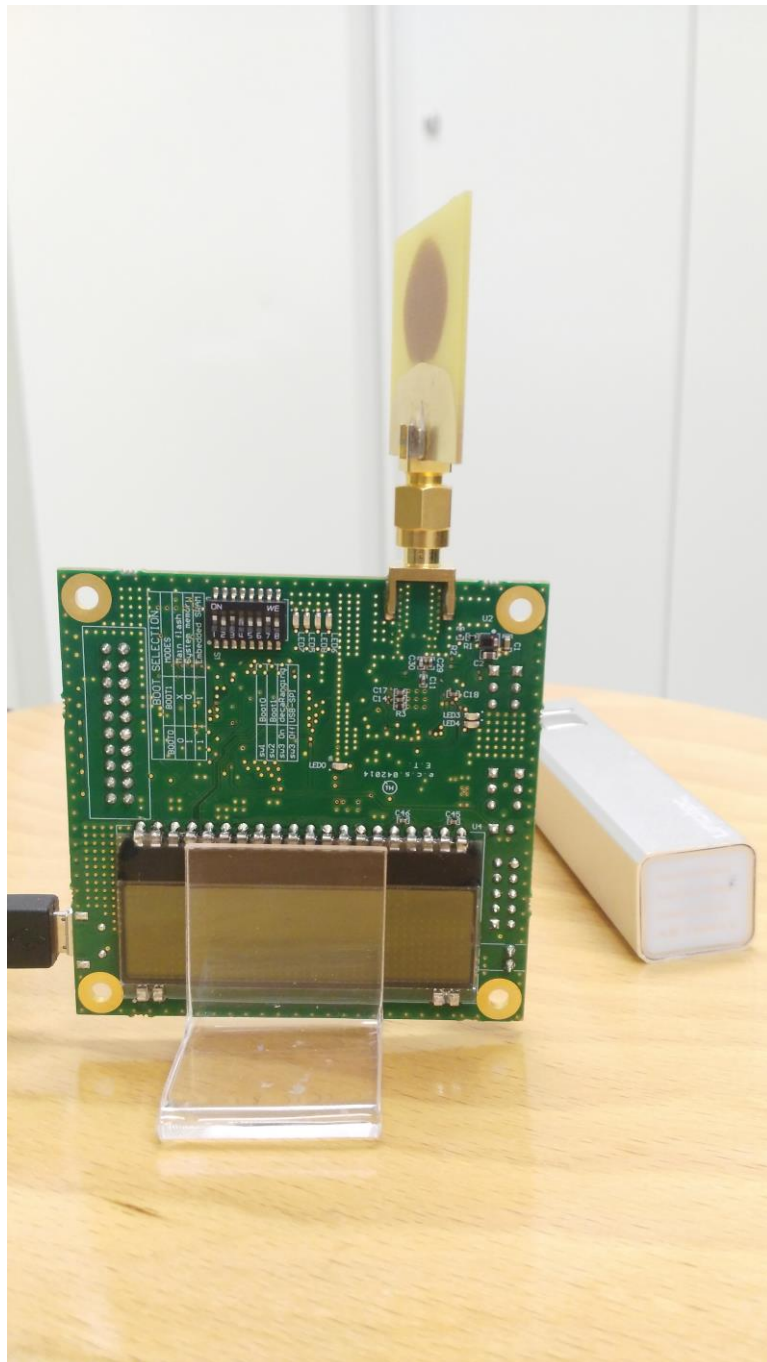


Figura 3-3: Placa del kit de Evaluación EVK1000 (tag).

### 3.2.1.1 Características

Las placas del kit de Evaluación EVK1000 presentan las siguientes características:

- Cumple IEEE 802.15.4-2011 UWB y las máscaras definidas por FCC y ETSI.

- Soporta 6 bandas RF de 3.5GHz a 6.5GHz.
- Programación de la potencia de salida.
- Soporta voltaje de 2.8V a 3.6V.
- Tasa de datos de 110kbps, 850 kbps y 6.8Mbps.
- Longitud máxima del paquete 1023 bytes para determinadas aplicaciones.
- Soporta Ranging TOA y Ranging TDOA.

### 3.2.1.2 Figuras de mérito

Las figuras de mérito de este sistema son las que siguen:

- Localización con una precisión de 10 cm en condiciones LOS.
- Rango de operación hasta de 290m con 110kbps de tasa de datos en estructuras RTLS.
- Alta inmunidad frente a desvanecimiento multitrayecto.

### 3.2.1.3 Ajuste

Este proyecto se basa principalmente en la comunicación, por tanto, en este apartado se explican las diferentes configuraciones referentes a la comunicación. Para este objetivo, el kit de Evaluación dispone del interruptor S1, que se divide en 8 interruptores a su vez:

<b>Interruptor</b>	<b>ON</b>	<b>OFF</b>	<b>¿Para qué sirve?</b>
<b>S1-1</b>	Activar ARM	Desactivar ARM	Funcionalidad ARM (si no se requiere, se puede dejar en OFF)
<b>S1-2</b>	Activar fast-ranging (5 ms)	Desactivar fast-ranging (150 ms)	Ajusta el tiempo de respuesta de las medidas
<b>S1-3</b>	Desactivar USB to SPI	Activar USB to SPI	Necesario para poder ejecutar la aplicación Decaranging
<b>S1-4</b>	Activa el <b>rol ancla</b>	Activa el <b>rol tag</b>	Permite cambiar el rol
<b>S1-5</b>	<b>Modos de operación</b>		
<b>S1-6</b>			
<b>S1-7</b>			
<b>S1-8</b>	Activa la configuración remota del tiempo de respuesta	Desactiva la configuración remota del tiempo de respuesta	Permite a la aplicación Decaranging cambiar el tiempo de respuesta

Tabla 3-1: Configuración de ajuste del kit de Evaluación.

Los modos de operación se exponen detalladamente en la tabla 2:

<b>S1-5</b>	<b>S1-6</b>	<b>S1-7</b>	<b>Modo</b>	<b>Canal</b>	<b>Tasa de datos</b>	<b>PRF</b>	<b>Preámbulo</b>	<b>Código de preámbulo</b>
OFF	OFF	OFF	1	2	110 kb/s	16	1024	3
ON	OFF	OFF	2	2	6,8 Mb/s	16	128	3
OFF	ON	OFF	3	2	110 kb/s	64	1024	9
ON	ON	OFF	4	2	6,8 Mb/s	64	128	9
OFF	OFF	ON	5	5	110 kbps	16	1024	3
ON	OFF	ON	6	5	6,8 Mb/s	16	128	3
OFF	ON	ON	7	5	110 kb/s	64	1024	9
ON	ON	ON	8	5	6,8 Mb/s	64	128	9

Tabla 3-2: Modos de operación del kit de Evaluación.

### 3.3 Funcionamiento de los equipos

Las dos placas EVB1000 que se han descrito anteriormente están configuradas a priori con la aplicación de prueba llamada *two-way Ranging*.

Estas placas tienen dos modos de funcionamiento:

- INT-INT: la aplicación Decaranging se ejecuta internamente en las dos placas y se intercambian mensajes. A partir de estos la aplicación calcula el tiempo de vuelo, estima la distancia y presenta por la pantalla LCD los resultados. Se requiere alimentación a las dos placas para este modo de funcionamiento.
- INT-EXT: en este modo de funcionamiento, una de las dos placas está conectada mediante USB a un microcontrolador externo (por ejemplo, un ordenador) el cual utiliza la aplicación Decaranging a través de la interfaz SPI.

Con este kit se pueden evaluar las siguientes características:

- Las medidas de distancia.
- El espectro de transmisión.
- El consumo de potencia y la corriente.
- La inmunidad a multitrayectos.
- Las opciones de la antena.

Para este proyecto, se utiliza el modo de funcionamiento INT-EXT, ya que se requiere de la aplicación Decaranging para obtener los logs de las medidas que se realicen.

Por tanto, y como se observa en la tabla 1, se debe de poner el interruptor S1-3 a OFF para activar la función USB to SPI.

El puerto USB virtual para Windows del que se requiere para la aplicación Decaranging se puede consultar en la página web de ST Microelectronics, aunque en el Apéndice A se explica detenidamente dónde descargarlo y cómo instalarlo correctamente.



### 3.4 Configuración de parámetros

En este apartado, se dispone a configurar el equipo teniendo en cuenta que se requiere para la identificación de medidas en interiores.

En interiores el rango aumenta frente a espacios al aire libre debido al multitrayecto. Por tanto, se debe conseguir una configuración que ofrezca el mayor rango de operación posible.

El rango operacional depende principalmente de tres factores:

- La tasa de datos de la trama.
- La longitud del preámbulo.
- El ancho de banda del canal y su frecuencia central.

A continuación, y teniendo todo lo anteriormente explicado en cuenta, se procede a configurar el equipo:

- **Tasa de datos:** una mayor tasa de datos se corresponde con un menor rango de operación, por lo que se debe elegir la menor tasa de datos disponible (110 kb/s).
- **Longitud del preámbulo:** para el objetivo de identificación, conviene tener un preámbulo largo de por sí para dar tiempo al receptor a procesar correctamente, pero además un preámbulo largo unido a una tasa de datos baja llevar a un mayor rango operativo. Por tanto, se elige el preámbulo más largo posible (1024).
- **Frecuencia central del canal:** en cuanto a la frecuencia central del canal, para el caso que nos ocupa se ha escogido la que nos proporciona el Canal 5, la cual es de 6,5 GHz.
- **Ancho de banda:** el ancho de banda correspondiente al Canal 5 de la configuración del kit de Decawave es de 500 MHz.
- **PRF:** la frecuencia de repetición de pulsos elegida en este proyecto es de 16 MHz.

A modo de resumen, la configuración del equipo será la que sigue: 110 kb/s de tasa de datos, longitud del preámbulo de 1024, uso del Canal 5 de frecuencias y ancho de banda y, por último, un PRF de 16 MHz.



## 4 MEDICIÓN EXPERIMENTAL

En este capítulo se explica todo lo relativo a las mediciones. Se exponen las distintas mediciones que se han realizado tanto para calcular el umbral como para ponerlo a prueba después. El objetivo de este punto es el de realizar medidas sabiendo a priori que son LOS o NLOS para el diseño de los criterios de identificación, ya sea mediante el ratio primer camino frente a potencia total o mediante el RMS Delay Spread, obtener un umbral de decisión a partir de sus resultados y, finalmente, poder probarlo con un conjunto de medidas independiente del primero. Además, se tratan también varios casos particulares que se han encontrado en las medidas durante la realización de estas.

### 4.1 Toma de medidas

La toma de medidas se ha llevado a cabo en diferentes puntos de la Escuela Técnica Superior de Ingeniería. Con este primer conjunto de medidas se consigue obtener un umbral de decisión para ambos métodos de identificación que se utilizan en este proyecto, para luego ponerlo a prueba en el segundo conjunto de medidas y llegar a un resultado.

#### 4.1.1 Medidas para calcular el umbral

Las medidas obtenidas en el despacho se han realizado desde trece puntos distintos de la Escuela Técnica Superior de Ingeniería y se han guardados en los archivos .log:

- 1) LOS.log: este archivo guarda una serie de medidas LOS a una distancia entre transmisor y receptor de 3 metros.
- 2) NLOS.log: en él están recogidas medidas NLOS a una distancia entre transmisor y receptor de 3 metros. En el camino directo se encuentra un armario metálico que lo obstaculiza.
- 3) NLOSlejos.log: igual que el anterior salvo que ahora el ancla se ha desplazado y se encuentra a 6 metros del tag. El camino directo sigue estando obstaculizado por el armario metálico.
- 4) Pared.log: en este caso, entre el ancla y el tag se encuentra una pared, lo que obstaculiza el rayo directo. Es un caso NLOS.

- 5) LOS16.log: medidas con línea de visión directa realizadas en la planta baja de la ETSI. Distancia de 16 metros.
- 6) LOS13.log: otro caso LOS, en otra zona de la planta baja de la ETSI, a una distancia de 13m.
- 7) NLOSparedv.log: caso NLOS con el ancla y el tag separados por una pared perpendicular al rayo directo distinta a la anterior. Distancia de 4,5 metros.
- 8) NLOSparedh.log: en este caso, ancla y tag están separados por la pared del log anterior pero esta vez el rayo directo se encuentra a la pared de forma oblicua. Distancia de 5 metros.
- 9) Pared3.log: nuevo caso de NLOS con pared entre ancla y tag, encontrándose estos a 3 metros.
- 10) Pared4.log: mismo caso que el anterior pero esta vez existiendo además un armario metálico obstaculizando el rayo directo. 3 metros de distancia.
- 11) NLOSarm2.log: medidas NLOS con ancla y tag separados exclusivamente de un armario metálico por su parte longitudinal. 2.5 metros de distancia.
- 12) LOS5.log: caso con visión directa a 5 metros de separación entre tag y ancla.
- 13) LOS3.log: igual que el anterior, pero acercando la distancia a 3 metros.

#### 4.1.2 Medidas para poner a prueba el umbral

Para el conjunto de medidas para poner a prueba el umbral se han realizado mediciones desde 6 puntos clave, de forma que se tengan muestras en todo tipo de situaciones posibles:

- 1) LOS14.log: esta serie de medidas han sido tomadas con línea de visión directa a lo largo de uno de los pasillos del departamento. La distancia es de 14 metros.
- 2) LOS8.log: estas medidas tienen línea de visión directa también. Se han tomado en otro pasillo del departamento, con una distancia de 8 metros.
- 3) NLOShab.log: medidas NLOS tomadas con el rayo directo interferido por varias paredes y habitaciones. Distancia de 14 metros.

- 4) NLOSarm.log: distancia de medida de 5 metros. El tag se encontraba escondido en el hueco entre 2 armarios metálicos. Rayo directo obstaculizado.
- 5) LOS4.log: la última posición con línea de visión directa. Estas medidas se han tomado en la intersección de los dos pasillos anteriormente comentados, donde se encuentra una pequeña sala de espera. Distancia de 4 metros.
- 6) NLOSpuerta.log: medidas NLOS tomadas con 5 metros de distancia, con una puerta metálica obstaculizando el rayo directo en el punto medio.

Todos estos logs se ubican en la carpeta de Google Drive referenciada por el siguiente enlace: <https://drive.google.com/drive/folders/1-K2NybCuo0hnhTME2LhLhuAJhpqmR1NV?usp=sharing>

## 4.2 Casos particulares

Una vez realizadas las mediciones, el paso siguiente era observar las gráficas de las medidas en MatLab para observar diferencias entre las medidas LOS y las NLOS que pudieran servir de ayuda a la hora de desarrollar los códigos de identificación más adelante. La diferencia entre ellas, como se puede observar en las figuras 4-1 y 4-2, es notoria:

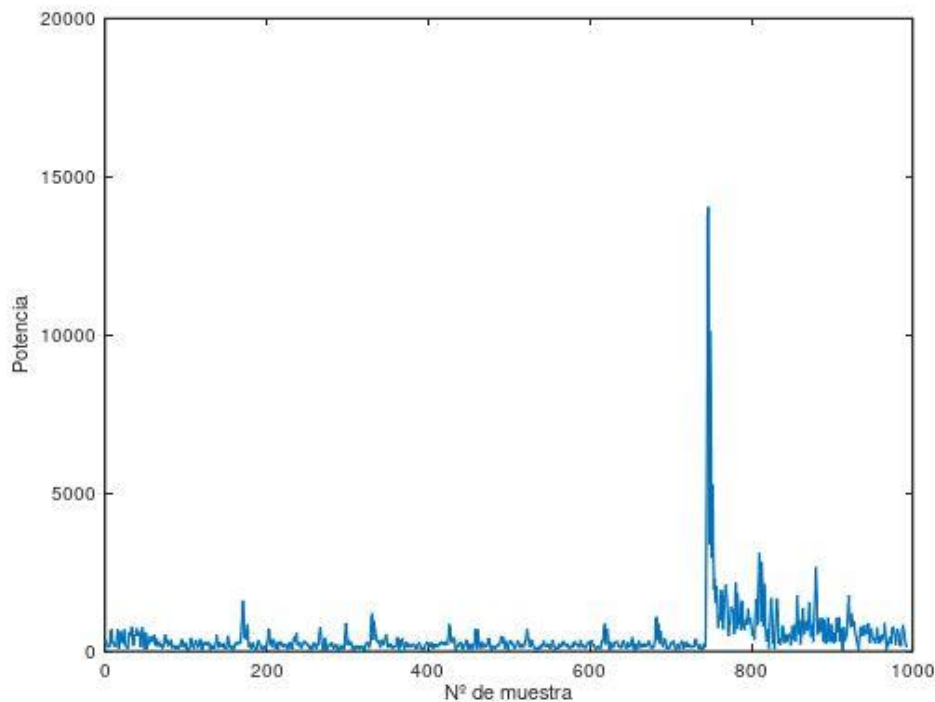


Figura 4-1: Ejemplo de señal LOS.

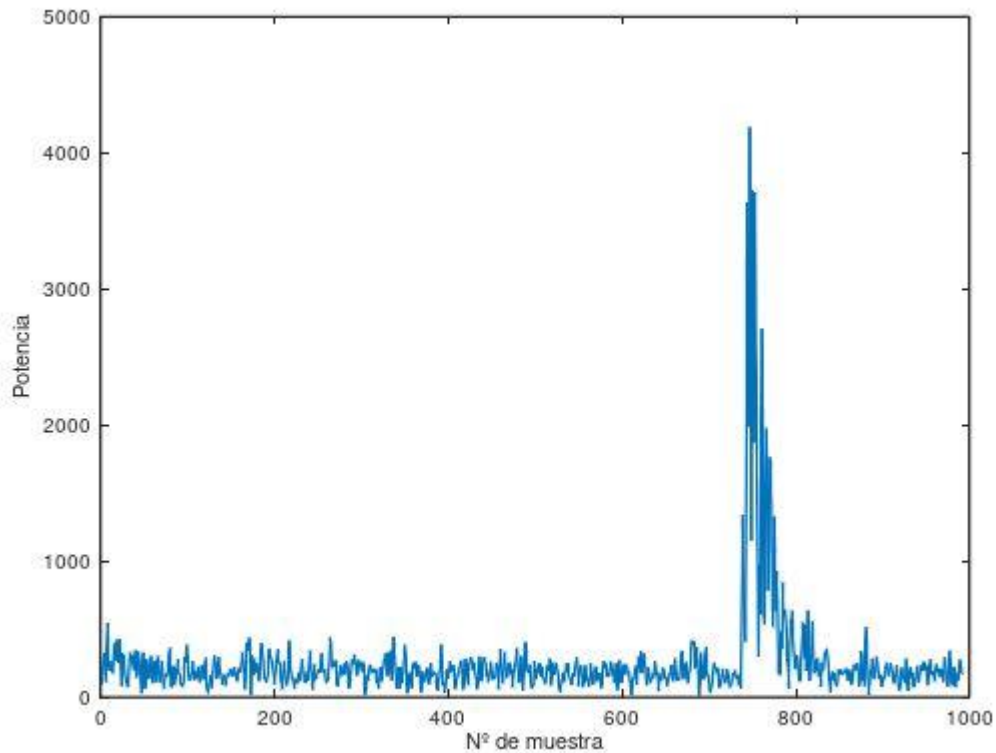


Figura 4-2: Ejemplo de señal NLOS.

Como se puede observar, la diferencia de potencia del pico máximo es muy grande. Esto tiene que ver con que la señal, a medida que va atravesando obstáculos, va perdiendo potencia. Además, el multitrayecto también se observa en tanto que la señal NLOS tiene varios picos de potencia bastante altos frente al ruido por detrás del pico máximo, cosa que en la señal LOS apenas se observa uno solo que cumpla esto.

Esto es lo esperable en señales de este tipo, pero durante el desarrollo del proyecto se han encontrado casos anómalos en algunas de las medidas realizadas que se explican aquí:

#### 4.2.1 Medidas corruptas

En el total de las 1126 medidas que se han realizado en los diferentes casos que se han explicado anteriormente, durante el desarrollo del código de identificación se han detectado que algunas medidas no cumplían con diferentes condiciones del código que supuestamente debían cumplir. Por tanto, se investigó el porqué y se dio con que algunas de las medidas estaban corruptas. Simplemente, el kit de DecaWave no recibió señal alguna en ese espacio de tiempo de muestras por algún motivo desconocido y solo detectó ruido, como se ilustra en la figura 4-3:

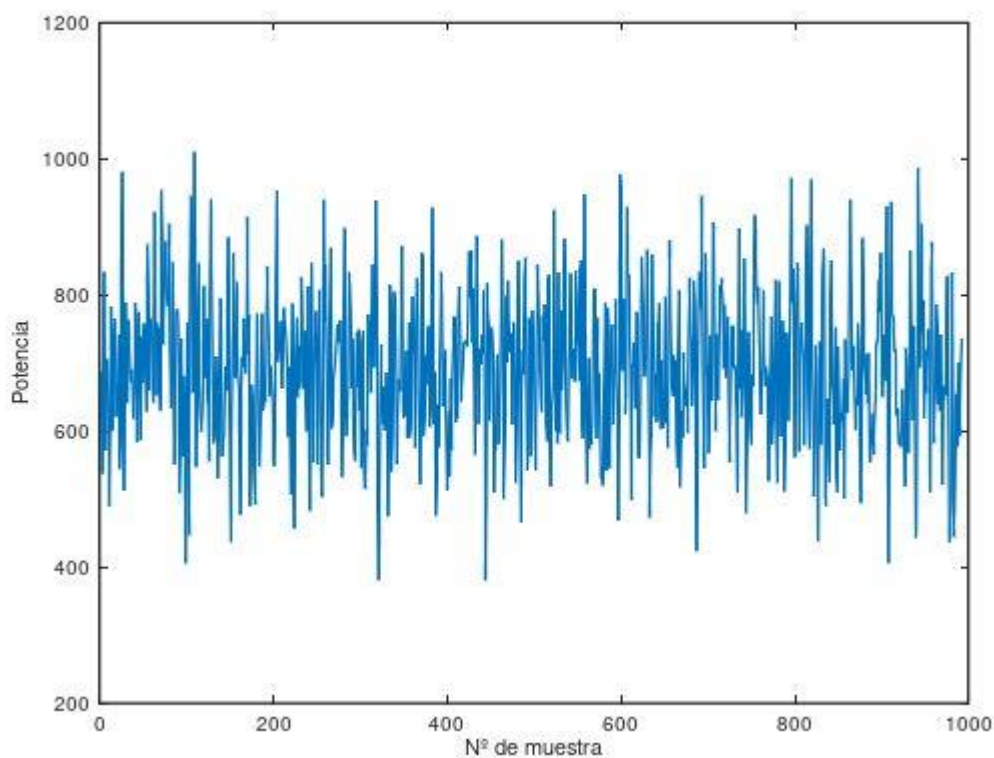


Figura 4-3: Ejemplo de medida corrupta.

El haber encontrado por primera vez una medida como esta, ha hecho que en el código se tenga en cuenta este fenómeno y se filtren este tipo de medidas. A su vez, en el código `Identifica.m` se han contabilizado el número de ocasiones que esto ocurre en el conjunto de todas las medidas tomadas para este proyecto. Solo hay 10 casos de 1126 medidas que se han realizado, lo que resulta algo más de un 0,88% del total.

#### 4.2.2 Caso de la pared

En este apartado, se trata el caso de las medidas realizadas sin línea de visión directa teniendo como obstáculo una pared.

El resultado esperado sería un caso intermedio entre el presentado como ejemplo en la figura 4-1 y el presentado en la figura 4-2, ya que no se está ante un camino directo libre pero tampoco se tiene como obstáculo un armario metálico.

Sin embargo, los resultados obtenidos en las gráficas son diferentes a lo que cabría esperar en un principio. La figura 4-4 ilustra una de las medidas tomadas con la configuración explicada:

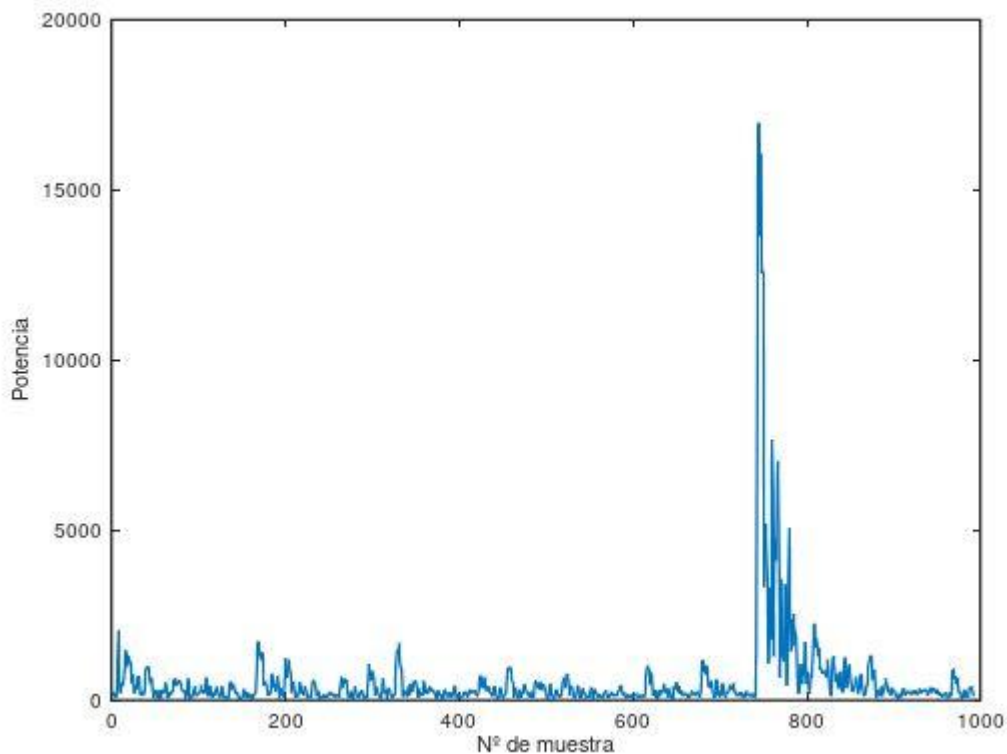


Figura 4-4: Ejemplo de medida NLOS con pared como obstáculo.

Se puede observar, comparándolo con las figuras 4-1 y 4-2 que este resultado está mucho más cerca de un caso LOS que de un caso NLOS cuando se esperaba algo más intermedio. Esto puede deberse a dos motivos:

- 1) La anchura de la pared: la pared que ejercía de obstáculo no era una pared de gran grosor, por lo que el obstáculo degradaría la señal poco.
- 2) La capacidad de penetración de la tecnología UWB: UWB tiene una capacidad de penetración de obstáculos muy buena, por lo que la señal es robusta cuando encuentra un obstáculo.

Por tanto, sumando los efectos de estos dos motivos, se puede dar una explicación lógica a este resultado inesperado.

### 4.2.3 Caso de la puerta

Este último caso anómalo nos lleva a las medidas realizadas teniendo como obstáculo la puerta metálica.

Estas medidas tenían como objetivo el obtener unas medidas NLOS con objeto metálico de por medio fuera del despacho donde se tomaron las medidas iniciales, pero una vez más el resultado al analizar las gráficas no fue el esperado, ni mucho menos.



En la figura 4-5 se ilustra la gráfica de una medida de este conjunto:

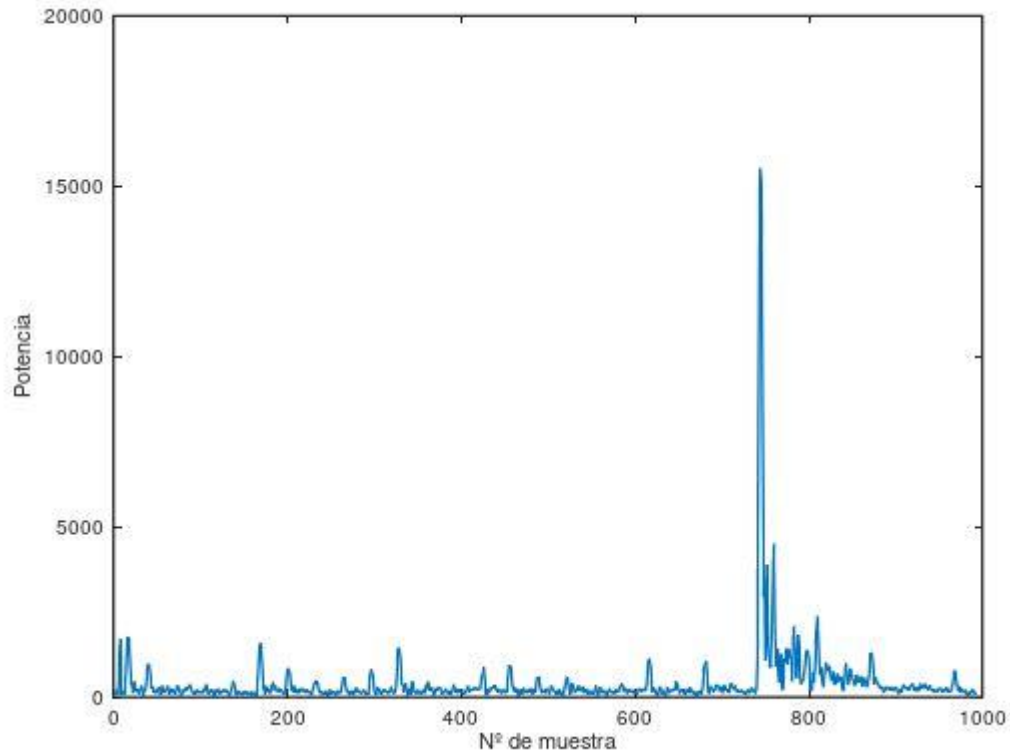


Figura 4-5: Ejemplo de medida NLOS con puerta como obstáculo.

Como puede observarse, se tiene una gráfica que no se parece en absoluto al caso NLOS de la figura 4-2. Este caso bien puede hacerse pasar por uno con línea de visión directa y, sin embargo, no lo es.

La única explicación a tan extraño suceso solo puede ser que ocurran estas dos cosas simultáneamente:

- 1) Que el rayo directo quede totalmente bloqueado.
- 2) Que un rayo indirecto, con un trayecto parecido al directo, consiga de algún modo salvar el obstáculo y llegar al receptor perdiendo muy poca potencia por el camino donde se haya reflejado.

De esta forma, el pico máximo de potencia que se observa en la gráfica de la figura 4-5 no sería el rayo directo, ya que este está totalmente bloqueado por la puerta, y un rayo con un camino indirecto consigue reflejarse con pocas pérdidas con algún objeto de forma que consigue llegar al receptor con un nivel de potencia muy bueno.



# 5 RATIO POTENCIA PRIMER CAMINO FRENTE A POTENCIA TOTAL

---

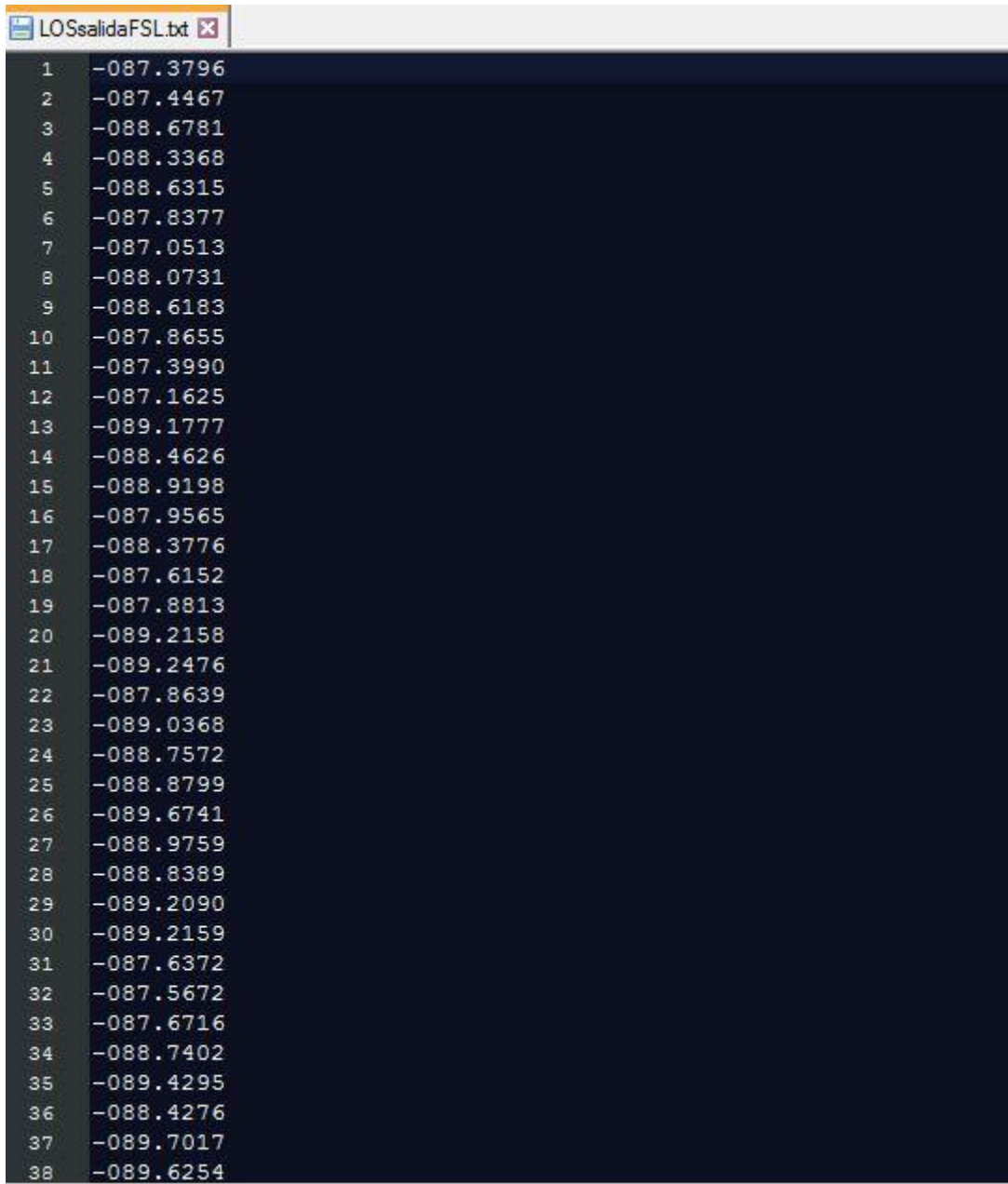
En este quinto punto tratamos el ratio potencia primer camino frente a potencia total. Para ello, se requiere de un código de MatLab/Octave que procese las medidas, un código ShellScript que vuelque los datos de los logs a MatLab/Octave y alguien que interprete los resultados que se obtengan.

En los logs de las medidas que nos proporciona DecaWave aparecen infinidad de datos sobre la señal, pero para este ratio que vamos a calcular solo necesitamos dos: el FSL, que hace referencia a la potencia del primer camino, y el RSL, que hace referencia a la potencia recibida.

## 5.1 Obtención de FSL y RSL a partir de los logs

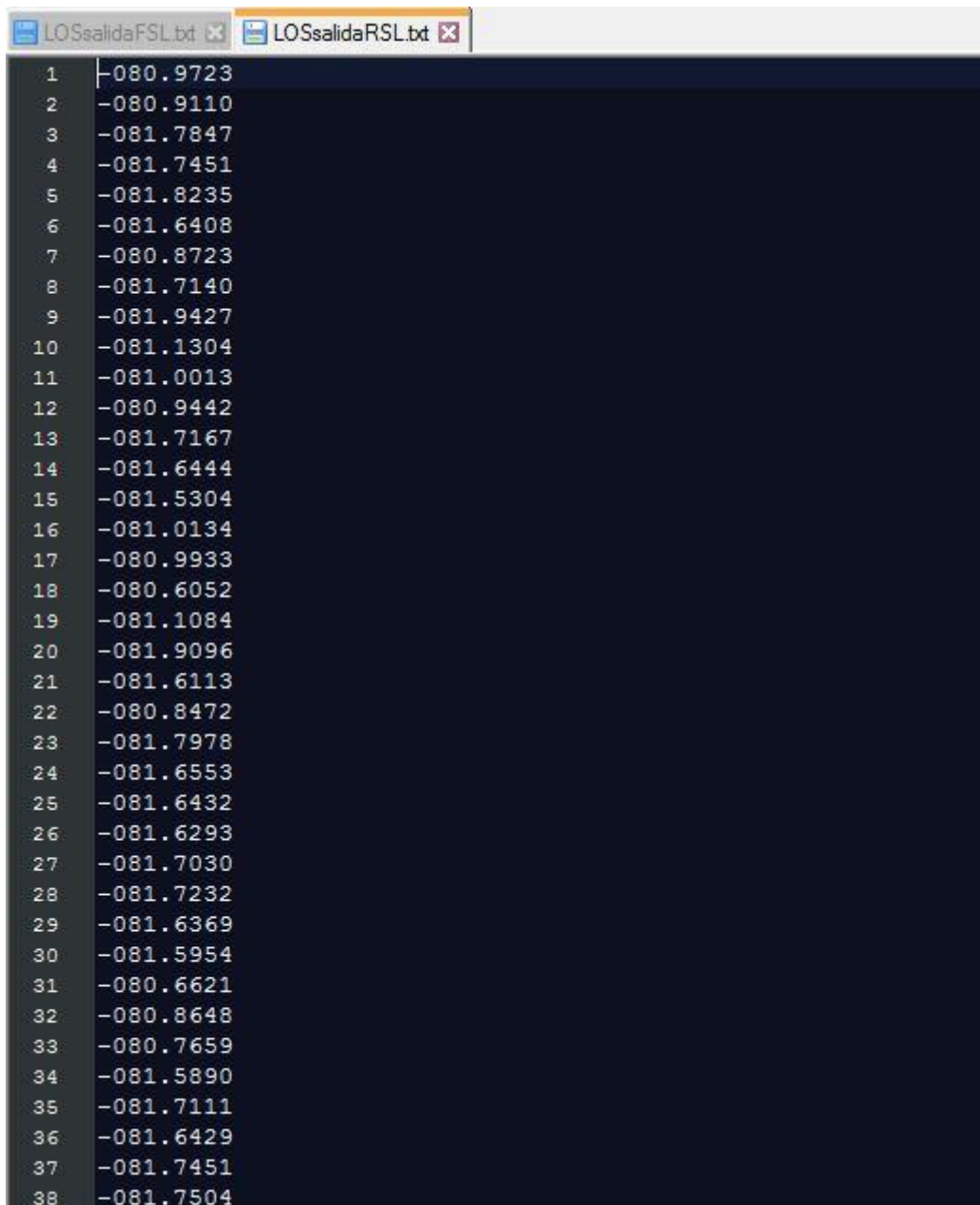
En los logs de las medidas que se han obtenido del software de DecaWave se encuentran los datos que necesitamos para calcular el ratio. Estos datos son FSL y RSL.

Para quedarnos solo con lo que nos interesa y filtrar lo demás, se utilizan los códigos ShellScript del Apéndice B *buscaFSL.sh* y *buscaRSL.sh* cuya salida es un fichero en el que nos muestra una lista con los FSL y RSL, respectivamente, de todas las medidas que existan en el log. Un ejemplo de resultado de ejecutar estos scripts es el siguiente:



```
LOSsalidaFSL.txt x
1 -087.3796
2 -087.4467
3 -088.6781
4 -088.3368
5 -088.6315
6 -087.8377
7 -087.0513
8 -088.0731
9 -088.6183
10 -087.8655
11 -087.3990
12 -087.1625
13 -089.1777
14 -088.4626
15 -088.9198
16 -087.9565
17 -088.3776
18 -087.6152
19 -087.8813
20 -089.2158
21 -089.2476
22 -087.8639
23 -089.0368
24 -088.7572
25 -088.8799
26 -089.6741
27 -088.9759
28 -088.8389
29 -089.2090
30 -089.2159
31 -087.6372
32 -087.5672
33 -087.6716
34 -088.7402
35 -089.4295
36 -088.4276
37 -089.7017
38 -089.6254
```

Figura 5-1: Contenido del fichero de salida de *buscaFSL.sh*.



```
LOSalidaFSL.bt x LOSalidaRSL.bt x
1 -080.9723
2 -080.9110
3 -081.7847
4 -081.7451
5 -081.8235
6 -081.6408
7 -080.8723
8 -081.7140
9 -081.9427
10 -081.1304
11 -081.0013
12 -080.9442
13 -081.7167
14 -081.6444
15 -081.5304
16 -081.0134
17 -080.9933
18 -080.6052
19 -081.1084
20 -081.9096
21 -081.6113
22 -080.8472
23 -081.7978
24 -081.6553
25 -081.6432
26 -081.6293
27 -081.7030
28 -081.7232
29 -081.6369
30 -081.5954
31 -080.6621
32 -080.8648
33 -080.7659
34 -081.5890
35 -081.7111
36 -081.6429
37 -081.7451
38 -081.7504
```

Figura 5-2: Contenido del fichero de salida de *buscaRSL.sh*.

## 5.2 Cálculo del ratio

Una vez obtenidos los valores de FSL y RSL de todos los logs de las medidas tomadas con anterioridad, solo queda cargarlos en MatLab/Octave y calcular el ratio con el código del Apéndice C *Ratio.m*.

Este código realiza el cálculo del ratio y además, genera dos histogramas para presentar los dos conjuntos de medidas independientes a considerar:

- Un primer histograma donde se presenta el conjunto de medidas para calcular el umbral. Gracias a este conjunto, se decide un umbral de decisión.
- El segundo histograma presenta el conjunto de medidas para poner a prueba el umbral. Una vez decidido el umbral en el primer histograma, es a este conjunto de medidas al cual le aplicaremos dicho umbral y sacar conclusiones al respecto mediante el cálculo de la probabilidad de error.

En las figuras 5-3 y 5-4 se ilustra el código donde se cargan en MatLab/Octave los valores de FSL y RSL de varios ficheros:

```
%Cargamos los datos obtenidos de los logs (cálculo umbral):  
  
load('logs/LOSsalidaRSL.txt');  
load('logs/LOSsalidaFSL.txt');  
  
load('logs/NLOSsalidaRSL.txt');  
load('logs/NLOSsalidaFSL.txt');  
  
load('logs/NLOSlejossalidaRSL.txt');  
load('logs/NLOSlejossalidaFSL.txt');  
  
load('logs/paredsalidaRSL.txt');  
load('logs/paredsalidaFSL.txt');  
  
load('logs/LOS16RSL.txt');  
load('logs/LOS16FSL.txt');  
  
load('logs/LOS13RSL.txt');  
load('logs/LOS13FSL.txt');  
  
load('logs/LOS5RSL.txt');  
load('logs/LOS5FSL.txt');
```

Figura 5-3: Código para exportar los datos de los ficheros .txt a MatLab/Octave

```

%Cargamos los datos obtenidos de los logs (prueba umbral):

load('logs/LOS14salidaRSL.txt');
load('logs/LOS14salidaFSL.txt');

load('logs/LOS8salidaRSL.txt');
load('logs/LOS8salidaFSL.txt');

load('logs/LOS4salidaRSL.txt');
load('logs/LOS4salidaFSL.txt');

load('logs/armsalidaRSL.txt');
load('logs/armsalidaFSL.txt');

load('logs/habsalidaRSL.txt');
load('logs/habsalidaFSL.txt');

load('logs/puertasalidaRSL.txt');
load('logs/puertasalidaFSL.txt');

```

Figura 5-4: Código para exportar los datos de los ficheros .txt a MatLab/Octave

Los valores obtenidos de los logs se encuentran en unidades logarítmicas (dBm) según ilustra el manual de Decaranging [13]. Por tanto, hay que realizar una conversión a unidades naturales previa a aplicar la Ecuación 3, la cual se recoge, como se dijo en el apartado 2, en la bibliografía [9].

Dicha conversión se realiza deshaciendo las operaciones de las ecuaciones 8 y 9 y así recuperar el valor en unidades naturales. Dichas ecuaciones son recogidas del manual correspondiente [4]:

$$10 \cdot \log\left(\frac{F_1^2 + F_2^2 + F_3^2}{N^2}\right) - A \text{ (dBm)}$$

Ecuación 8: Estimación nivel de potencia del primer camino.

$$10 \cdot \log\left(\frac{C + 2^{17}}{N^2}\right) - A \text{ (dBm)}$$

Ecuación 9: Estimación de potencia recibida.

Donde  $F_1, F_2$  y  $F_3$  hacen referencia a valores de amplitud de potencia que el software recoge en registros internos,  $C$  es la potencia correspondiente a la respuesta al impulso del canal,  $N$  es el valor del contador interno de preámbulo y  $A$  es una constante del software cuyo valor es de 113.77.

A continuación, se calcula el ratio de todos los datos con las líneas de código que se muestran en las figuras 5-

5 y 5-6:

```

%Calculamos ratios (umbral):
for i=1:length(LOSSalidaRSL)
    ratioLOS(i)=(FSL(i)/RSL(i));
endfor

for j=1:length(NLOSSalidaRSL)
    ratioNLOS(j)=(NFSL(j)/NRSL(j));
endfor

for k=1:length(NLOSlejossalidaRSL)
    ratioNLOSlejos(k)=(LFSL(k)/LRSL(k));
endfor

for l=1:length(paredsalidaRSL)
    ratiopared(l)=(PFSL(l)/PRSL(l));
endfor

for s=1:length(RSL16)
    ratioLOS16(s)=(FSL16(s)/RSL16(s));
endfor

for a=1:length(RSL13)
    ratioLOS13(a)=(FSL13(a)/RSL13(a));
endfor

for za=1:length(RSL5)
    ratioLOS5(za)=(FSL5(za)/RSL5(za));
endfor

for d=1:length(RSL3)
    ratioLOS3(d)=(FSL3(d)/RSL3(d));
endfor

```

Figura 5-5: Código que calcula el ratio de las medidas para calcular el umbral.



```
%Calculamos ratios (prueba umbral):  
  
for m=1:length(RSL14)  
    ratioLOS14(m)=(FSL14(m)/RSL14(m));  
endfor  
  
for n=1:length(RSL8)  
    ratioLOS8(n)=(FSL8(n)/RSL8(n));  
endfor  
  
for o=1:length(RSL4)  
    ratioLOS4(o)=(FSL4(o)/RSL4(o));  
endfor  
  
for p=1:length(RSLarm)  
    ratioarm(p)=(FSLarm(p)/RSLarm(p));  
endfor  
  
for q=1:length(RSLhab)  
    ratiohab(q)=(FSLhab(q)/RSLhab(q));  
endfor  
  
for r=1:length(RSLpuerta)  
    ratiopuerta(r)=(FSLpuerta(r)/RSLpuerta(r));  
endfor
```

Figura 5-6: Código que calcula el ratio de las medidas para poner a prueba el umbral.

Una vez calculado el ratio de todas las medidas, se juntan todos los correspondientes a las medidas tomadas para la obtención del umbral por un lado en un array, y en otro array se juntan los ratios que ponen a prueba el umbral para hacer la comparativa.

### 5.3 Comparativa de resultados

En este apartado se desarrolla y se ponen de relevancia los resultados obtenidos.

Para poder observar los resultados se hace uso del comando *hist* de MatLab/Octave, el cual generará los histogramas de los arrays con los ratios de los dos conjuntos de medidas que se han considerado anteriormente.

De esta manera, en la figura 5-7 se ilustra el histograma generado por MatLab/Octave del código *Ratio.m*, dicho código se puede encontrar detallado en el Apéndice C, que corresponde únicamente con el conjunto de medidas para obtener el umbral de decisión:

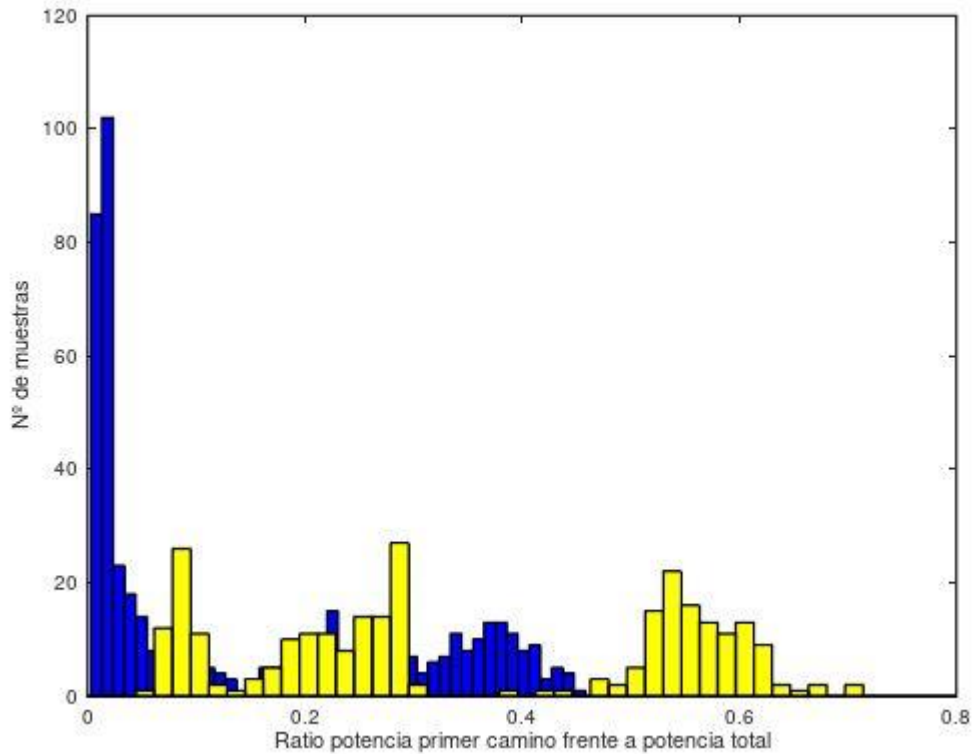


Figura 5-7: Histograma de los ratios de las medidas para buscar el umbral.

Código de colores de los histogramas de la figura 5-7:

- Azul: Medidas NLOS.
- Amarillo: Medidas LOS.

Como se puede observar en el histograma de la figura 5-7, no se encuentra una manera posible de hacer una diferenciación LOS/NLOS con el cruce de dos distribuciones normales, ya que hay muestras NLOS que entran en el terreno de las medidas LOS. Por tanto, más adelante no se calculará la probabilidad de error como tal, sino que se enfocará de manera diferente.

Para completar la definición del criterio de identificación, se escogerá un umbral de decisión que contenga la primera región de medidas NLOS con ratio pequeño. Si se compara este histograma con la figura 2-1, se observa que esta región se corresponde probablemente en gran parte con las medidas NLOS/UDP, las cuales son las más perjudiciales ya que no se detecta el trayecto directo.

Ese umbral corresponde con un ratio potencia primer camino frente a potencia total de **0.15**.

Una vez obtenido el umbral, se procede de la misma forma con el conjunto de medidas para poner a prueba dicho umbral y se genera el histograma correspondiente, como se ilustra en la figura 5-8:

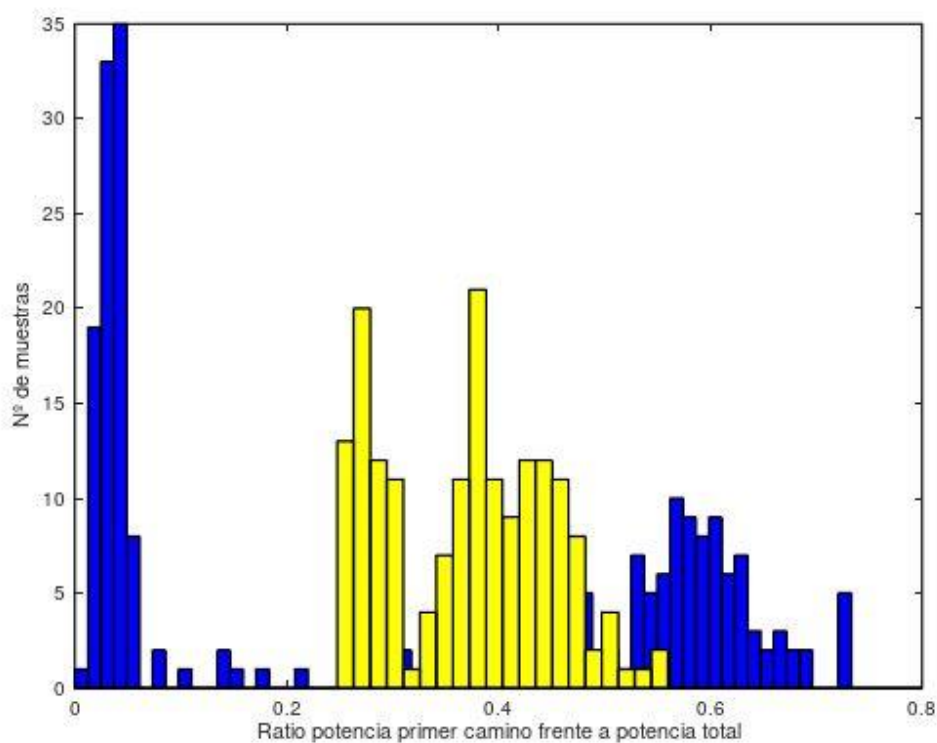


Figura 5-8: Histograma de los ratios de las medidas para poner a prueba el umbral.

Código de colores de los histogramas de la figura 5-8:

- Azul: Medidas NLOS.
- Amarillo: Medidas LOS.

En el histograma se observa que, de nuevo, existen medidas NLOS en la zona LOS.

Con el umbral de 0,15 escogido con la primera batería de muestras, se observa que para este segundo conjunto la forma del histograma para la zona NLOS con bajo ratio se puede corresponder con la zona NLOS-UDP de la figura 2-1, al igual que el histograma anterior. Esto apunta a que el sistema de identificación es relativamente fiable en cuanto a la migración a otros lugares y puesta en marcha.



---

## 6 RMS DELAY SPREAD

---

En este apartado se explica qué hace el RMS Delay Spread y cómo se implementa en un código para conseguir realizar la identificación LOS/NLOS que se busca.

El RMS Delay Spread lo que pretende es, a grandes rasgos, cuantificar cuánto se ha ensanchado en el tiempo la señal que hemos enviado debido al canal multitrayecto.

Por tanto, para una señal con línea de visión directa, es decir, LOS, tendremos un valor de RMS Delay Spread relativamente bajo; mientras que para una señal sin línea de visión directa, o NLOS, cabría esperar un valor de RMS Delay Spread relativamente alto.

Para llevar a cabo el código de MatLab/Octave que calcule el RMS Delay Spread necesitamos primero las gráficas que proporcionan los logs de todas las medidas.

Después, una vez se tienen las gráficas necesitamos averguar dos puntos clave: el pico máximo de potencia y el pico *final de señal*. Este pico de final de señal es clave porque es el punto de la gráfica en el cual se decide que la señal que enviamos termina y más allá se encuentra ruido.

Un ejemplo de gráfica de una medida puede ser el que ilustra la figura 6-1:

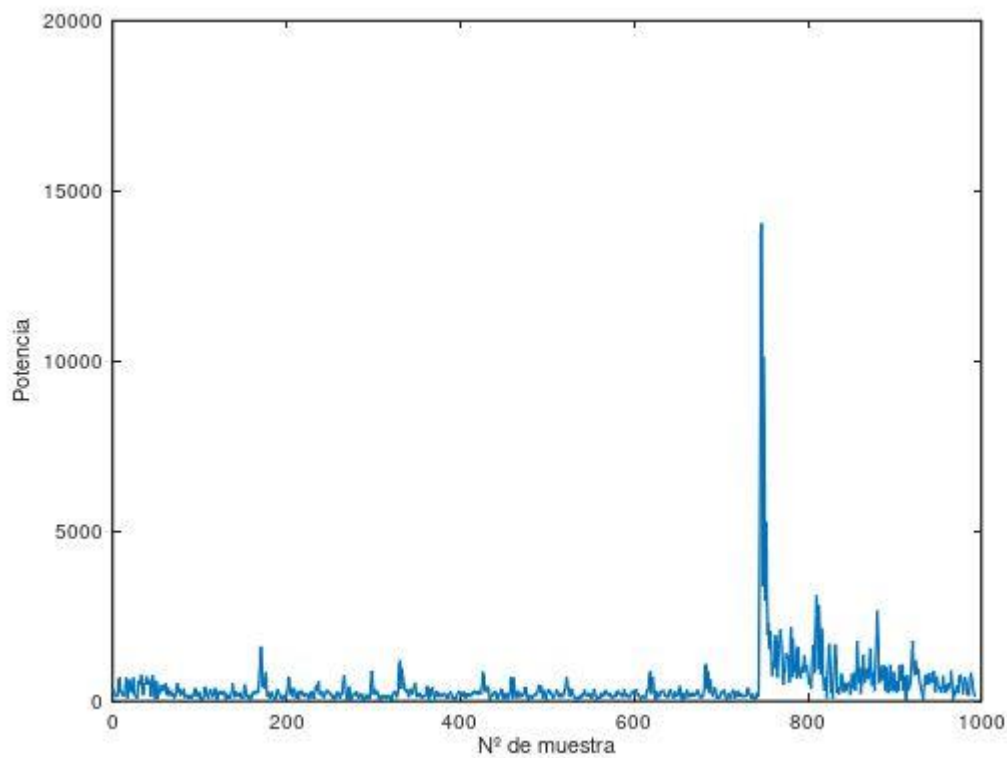


Figura 6-1: Ejemplo de gráfica de una medida.

Como se puede observar, es posible dividir esta gráfica en tres partes diferenciadas, de izquierda a derecha:

- Un fondo de ruido.
- La señal deseada, la cual hay que determinar en qué momento acaba.
- De nuevo, un fondo de ruido.

En el código de MatLab/Octave del archivo *Identifica.m*, el cual se encuentra en el Apéndice C, apartado 11.2.1, se realizan una serie de algoritmos para estimar el pico máximo y un punto final de señal, con tal de delimitarla y poder así ejecutar el algoritmo del RMS Delay Spread posteriormente.

## 6.1 Cálculo del máximo de la señal

El primer paso para el cálculo del RMS Delay Spread es la obtención del pico máximo de la señal recibida. Para ello debemos de extraer los datos de la gráfica de la señal del log antes de exportarlos a MatLab/Octave. Esto se consigue mediante un script en una consola de Linux y se explica cómo se consigue en el Apéndice A.

Una vez se tiene en MatLab/Octave cargados los datos de las gráficas de todas las medidas realizadas en este proyecto, en el código del Apéndice C llamado *Identifica.m* se encuentran las siguientes líneas que consiguen extraer tanto el valor del máximo de la señal como la posición dentro de la gráfica donde se encuentra. En la figura 6-2 se recoge el extracto de código que consigue esto:

```
%Busqueda de pico máximo:
i=1;
for i=1:992
    if valor<real(grafica(i))
        if x2(i) != -Inf
            valor=real(grafica(i));
            pulso=i;
        end
    end
endfor
```

Figura 6-2: Extracto de código de búsqueda del pico máximo.

Es un bucle sencillo en el cual se recorren los 992 puntos que tienen todas las gráficas que proporciona DecaWave en sus logs. Dentro del bucle, si el valor de la gráfica es mayor que el recogido en la variable *valor* pues esta se actualiza, y además en la variable *pulso* se recoge la posición en la que se encuentra ese nuevo máximo.

Cuando termine de recorrer toda la gráfica quedarán recogidos tanto el punto máximo como su posición, algo que necesitaremos en el paso siguiente.

## 6.2 Algoritmo RMS Delay Spread

Una vez calculado el pico máximo de potencia se dispone al cálculo del punto final de la señal y al cálculo del RMS Delay Spread.

### 6.2.1 Algoritmo fin de señal

Para encontrar el punto donde termina la señal recibida y el resto vuelve a ser ruido, por la naturaleza de las gráficas obtenidas en este proyecto se ha decidido por encontrar este punto mediante un umbral.

Las señales LOS tienen un pico muy abrupto y rápidamente cae el nivel de señal hacia el ruido. Sin embargo, las señales NLOS no tienen un pico máximo muy alto y además presentan muchas reflexiones que hacen que la señal se ensanche temporalmente.

Por tanto, para conseguir un umbral que encaje bien en las diferentes situaciones, se ha optado por el umbral valor del pico máximo de potencia dividido entre 9.

El umbral para el final de la señal es obtenido mediante el estudio y la observación de las medidas recogidas en este trabajo, ya que estas detecciones se encuentran fuera del objeto de este.

### 6.2.2 Ecuación RMS Delay Spread

Una vez se han estimado los puntos necesarios (punto máximo y fin de señal) se dispone a ejecutar el RMS Delay Spread.

En este apartado se explica el algoritmo realizado únicamente para el cálculo del RMS Delay Spread siguiendo la ecuación recogida en el apartado teórico.

Los pasos que sigue el algoritmo son los siguientes:

- Se obtiene el valor de  $\tau_m$ .
- Se sustituyen los valores en la fórmula del RMS Delay Spread.

## 6.3 Representación de resultados

Para finalizar el apartado del RMS Delay Spread, se representan los histogramas referentes al cálculo de este.

Al igual que en el apartado 4, se recurre una vez más al comando *hist* de MatLab/Octave para llevar a cabo esta representación y se hará la misma división de medidas anterior: primero se muestra el histograma con las medidas para calcular el umbral y después se expone el histograma con las medidas realizadas para poner a prueba dicho umbral, donde se observarán las diferencias.

El código donde se generan estos histogramas se encuentra dentro del script *Identifica.m*, el cual se encuentra en el Apéndice C de este documento.

El histograma que ilustra la figura 6-3 es el que se corresponde con las medidas tomadas para el cálculo del



umbral, y el que ilustra la figura 6-4, el de las medidas recogidas para ponerlo a prueba.

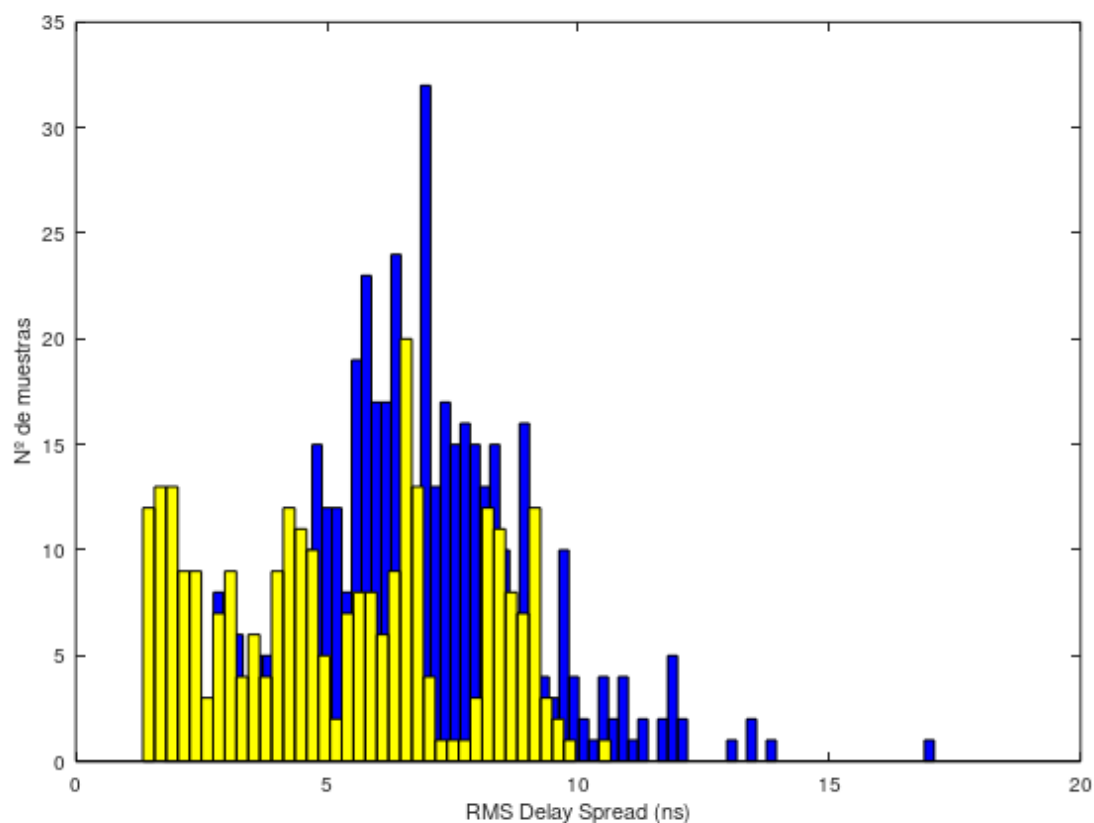


Figura 6-3: Histograma del RMS Delay Spread de las medidas para calcular el umbral.

Código de colores de los histogramas de la figura 6-3:

- Azul: Medidas NLOS.
- Amarillo: Medidas LOS.

Si se examina el histograma, se ve cómo para esta técnica nos encontramos con medidas LOS en zona NLOS y viceversa. Esto indica que no se puede esperar la identificación de la totalidad de los casos, y además que la probabilidad de error que se pueda conseguir con el RMS Delay Spread sea algo peor que la del ratio.

Para este caso, como a partir del histograma no se ve un valor óptimo del umbral, he elegido uno por inspección que abarca al menos la primera de las regiones LOS, de manera que se puedan calcular probabilidades de identificación. Este umbral de decisión para la identificación tiene un valor de **6 ns**.

A la vista del histograma de la figura 6-3, se puede decir que este criterio presenta problemas en la práctica, ya que las regiones se solapan. Por tanto, no se podrán obtener probabilidades de identificación correcta altas utilizando de forma aislada este criterio.

A continuación, se presenta el histograma del RMS Delay Spread de las medidas obtenidas para poner a

prueba al umbral mediante la figura 6-4:

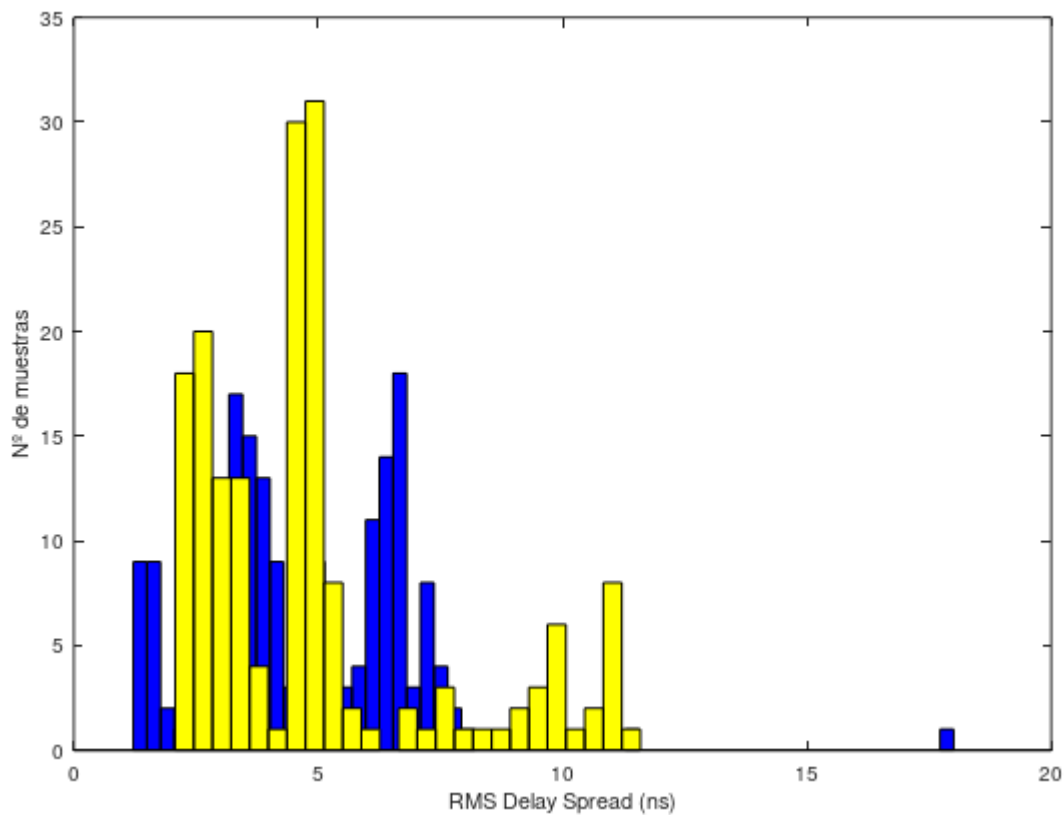


Figura 6-4: Histograma del RMS Delay Spread de las medidas para poner a prueba el umbral.

Código de colores de los histogramas de la figura 6-4:

- Azul: Medidas NLOS.
- Amarillo: Medidas LOS.

Con el umbral elegido anteriormente, se observa que para este conjunto de medidas, independiente del anterior, también nos encontramos medidas LOS en zona NLOS y viceversa. Esto parece indicar que la identificación con esta técnica sea más complicada, pero para cuantificar esto, aplicado al conjunto de medidas usadas en este proyecto, se han de calcular las probabilidades de error en el siguiente punto.



## 7 CÁLCULO DE LA PROBABILIDAD DE ERROR

---

En este punto se trata de intentar cuantificar el grado de eficacia que tienen las técnicas de identificación que se han explicado a lo largo de este proyecto. Con los histogramas observados en los puntos 5 y 6 de esta memoria, uno puede intuir, por la posición de las distribuciones normales que allí se muestran, que el ratio puede tener una mejor probabilidad de error frente al RMS Delay Spread, pero aquí se comprueba realmente cuál de los dos ofrece mejor probabilidad para el conjunto de medidas utilizados en este proyecto.

### 7.1 Probabilidad de error del ratio potencia primer camino frente a potencia total

Teniendo en cuenta el resultado obtenido en el punto 5, no es posible obtener una diferenciación buena para LOS/NLOS. Sin embargo, el umbral elegido anteriormente nos servirá para poder identificar parte de los casos LOS y parte de los casos NLOS.

Por tanto, las probabilidades que aquí se calculan son dos:

- Probabilidad de detectar LOS sabiendo que es una medida NLOS:  $p(\text{LOS}|\text{NLOS})$ .
- Probabilidad de detectar NLOS sabiendo que es una medida LOS:  $p(\text{NLOS}|\text{LOS})$ .

Para calcular la probabilidad de error de este ratio, se requiere de un algoritmo que realice este cálculo. En la figura 7-1 se ilustra el contenido de dicho algoritmo, el cual se puede encontrar además en el Apéndice C, concretamente en el archivo *Ratio.m*.

```

%Probabilidad de que se detecte LOS sabiendo que realmente es NLOS:

for bc=1:length(resultado_ratio)

    if ((resultado_ratio(bc)== 1) & (resultado_priori(bc)==0))
        falso_LOS++;
    endif
endfor

%Probabilidad de que se detecte NLOS sabiendo que realmente es LOS:

for c=1:length(resultado_ratio)
    if ((resultado_ratio(c)== 0) & (resultado_priori(c)==1))
        falso_NLOS++;
    endif
endfor

%Inicializamos variables de probabilidad:

pLOS_NLOS=0;
pNLOS_LOS=0;

%Calculamos las probabilidades:

pLOS_NLOS=(falso_LOS/204) %205 - una trama corrupta que se encuentra en este grupo
pNLOS_LOS=(falso_NLOS/173)

```

Figura 7-1: Extracto del código para la probabilidad de error del ratio.

Las probabilidades de error obtenidas con el umbral de 0,15 elegido a la luz de los resultados de las gráficas son las siguientes:

- $p(\text{LOS}|\text{NLOS}) = 50,490 \%$
- $p(\text{NLOS}|\text{LOS}) = 0\%$

Con este umbral se ha conseguido identificar correctamente todas las medidas LOS y prácticamente la mitad de las NLOS.

## 7.2 Probabilidad de error del RMS Delay Spread

Para el caso del RMS Delay Spread se sigue el mismo procedimiento que en el punto anterior. Las probabilidades obtenidas de las gráficas resultantes de ejecutar la técnica RMS Delay Spread son las que siguen con el umbral escogido en el punto 6:

- $p(\text{LOS}|\text{NLOS}) = 68,966 \%$
- $p(\text{NLOS}|\text{LOS}) = 19,075 \%$

Para calcular la probabilidad de error del RMS Delay Spread, existe un algoritmo dentro del script *Identifica.m* del Apéndice C específico para este fin. En la figura 7-2 se ilustran las líneas de código dedicadas a ese cálculo:

```
%Probabilidad de que se detecte LOS sabiendo que realmente es NLOS:

for bc=1:length(resultado_rms)
    if ((resultado_rms(bc)== 1) & (resultado_priori(bc)==0))
        falso_LOS++;
    endif
endfor

%Probabilidad de que se detecte NLOS sabiendo que realmente es LOS:

for bd=1:length(resultado_rms)
    if ((resultado_rms(bd)== 0) & (resultado_priori(bd)==1))
        falso_NLOS++;
    endif
endfor

%Inicializamos variables de probabilidad:

pLOS_NLOS=0;
pNLOS_LOS=0;

%Calculamos las probabilidades:

pLOS_NLOS=(falso_LOS/200)
pNLOS_LOS=(falso_NLOS/173)
```

Figura 7-2: Extracto del código para la probabilidad de error del RMS Delay Spread.

A la luz de estas probabilidades, se puede observar que para el ratio se consiguen identificar la totalidad de las medidas LOS y se comprueba que el resultado de identificación NLOS es bastante mejor que el que se consigue con la técnica del RMS Delay Spread para el mismo caso.



## 8 CONCLUSIONES

---

En este apartado se exponen las conclusiones de este proyecto.

En este documento se ha descrito el proceso que se ha realizado de toma de medidas reales realizadas en el Departamento de teoría de la señal y comunicaciones. Se trata de medidas de respuesta de un canal UWB para diferentes posiciones de los transmisores y distancias entre ellos. A partir de esta base común, en todo momento se ha avanzado de manera independiente por dos vertientes relativas a dos criterios de identificación de la condición LOS o NLOS de dichas medidas: la del ratio potencia primer camino frente a potencia total, y la del RMS Delay Spread.

Una vez se consigue obtener unos resultados concretos por cada criterio de manera independiente, fueran estos los que fueran, se llega a este punto, donde estos dos caminos trazados durante el proyecto convergen y se comparan los resultados conseguidos por cada uno.

Por tanto, estas conclusiones se pueden dividir en tres partes a valorar: la toma de las medidas, la comparativa de las técnicas de identificación expuestas en el proyecto y el análisis de los resultados.

En cuanto a toma de las medidas, se ha optado por una toma de medidas separada: un primer conjunto de medidas encaminado a la búsqueda de un umbral de decisión y un segundo conjunto, con medidas totalmente independientes, para poner a prueba ese umbral elegido para comprobar si funciona con una probabilidad aceptablemente buena.

De esta forma se podría observar si, gracias a estas técnicas, el algoritmo diseñado era fácilmente transportable o no en medida de las diferencias que se observarían en la comparación de los dos conjuntos de medidas realizados mediante sus histogramas.

Una vez terminado el proyecto, se puede intuir que el hecho de haber realizado medidas en diferentes lugares de la Escuela Técnica Superior de Ingeniería ha aportado variabilidad al conjunto de medidas, lo cual ha sido una información muy valiosa para esta investigación.

En cuanto a la comparativa de las técnicas de identificación hay varios puntos a examinar:

- La forma en la que se han obtenido el ratio potencia primer camino frente a potencia total y el RMS Delay Spread han sido diferentes: mientras que el ratio se ha calculado de datos proporcionados directamente de DecaWave, es decir, calculados por el algoritmo interno del software, en el caso del RMS Delay Spread se ha servido de información proporcionada por el software pero los cálculos realizados a partir de esa información han sido llevados a cabo por un algoritmo realizado por mí.
- En cuanto al coste computacional de ambas técnicas, el ratio ofrece mucho mejores prestaciones que el RMS Delay Spread. El RMS Delay Spread se sirve de cálculos mucho más complejos que el ratio, ya que la ecuación que lo describe es más compleja. Esto se puede observar en el apartado teórico.

Parte de la diferencia en cuanto coste computacional se refiere, posiblemente se deba a que el



algoritmo interno de DecaWave ya calcula la potencia del primer camino y la potencia total recibida y, en contrapartida, el algoritmo creado para determinar algunos de los parámetros incluidos en el RMS Delay Spread, creado por mí, ha de tratar con los datos que DecaWave proporciona de las gráficas de señal.

- Sobre la transportabilidad de estos criterios de identificación, teniendo en cuenta la comparativa entre los dos histogramas obtenidos para cada uno de ellos (figuras 5-7 y 5-8 para el ratio y figuras 6-3 y 6-4 para el RMS Delay Spread) y la variabilidad de lugares donde se han obtenido las muestras, se puede decir que cabría esperar unos resultados relativamente similares a los obtenidos en este proyecto si se pusiera a prueba en otro lugar. Esta transportabilidad podría mejorarse aumentando la base de datos de medidas incluyendo una mayor variedad de localizaciones.
- Los resultados obtenidos para el ratio apuntan a que si se consideran por separado los casos NLOS/DDP y los NLOS/UDP, la región NLOS-UDP (la más perjudicial) quizás se podría identificar con una mayor probabilidad, teniendo en cuenta las curvas de la figura 2-1 en comparación con los histogramas de las figuras 5-7 y 5-8.
- En el apartado de la probabilidad de error, las prestaciones del ratio son mejores que las de la versión que se ha implementado del RMS Delay Spread, como se observó en el apartado 7.

Se comprobó que la probabilidad de error en cuanto la cantidad de medidas LOS detectadas como NLOS y la cantidad de medidas NLOS detectadas como LOS es más bajo en el caso del ratio que en el RMS Delay Spread.

A la vista de los resultados obtenidos con estas dos técnicas de identificación, se puede observar que abordar este problema es difícil, ya que llevados a la práctica estos algoritmos de identificación, de forma aislada, no garantizan probabilidades de identificación correcta muy buenas.

Para finalizar, algunas ideas para proyectos futuros que me han surgido durante la realización de este proyecto serían:

- Un diseño más pulido y eficiente del RMS Delay Spread en cuanto a la obtención del principio y fin de la señal, los cuales son difíciles de conseguir de manera precisa. Y esto afecta directamente al resultado del RMS Delay Spread.
- Ampliar la base de datos de medidas para incluir una mayor variedad de casos.
- Investigar y analizar otros métodos de identificación más sofisticados para comprobar su rendimiento, como por ejemplo el método de kurtosis, o realizar la identificación mediante una combinación de métodos sencillos para comprobar si existe así una mejora de resultados respecto a usarlos aisladamente.
- Plantear un algoritmo que identificara únicamente las medidas NLOS/UDP con el objetivo de descartarlas, mientras que las medidas NLOS/DDP, al ser relativamente buenas, podrían conservarse, por lo que no importaría tanto que se solapen con las medidas LOS.



## 9 APÉNDICE A: INSTALACIÓN Y USO DEL SOFTWARE

---

La estructura y presentación de los distintos apartados de este Apéndice se han inspirado del Trabajo Fin de Grado «Evaluación de dispositivos UWB para localización en interiores con medidas de distancia» de Luis de la Calle [12], ya que él trata con el mismo dispositivo DecaWave que el que se ha utilizado en este proyecto.

### 9.1 Instalación

Para realizar todas las medidas, se ha usado un software incluido en el paquete EVK1000. Ahora se explicará un breve resumen con los pasos a seguir para su instalación:

#### 1) USB virtual COM

Lo primero que se hace es instalar el driver para conectar el puerto EVB1000 USB. De esta manera, se puede usar el EVB1000 ARM software USB-to-SPI. El driver se puede conseguir en la página web de ST Microelectronics:

<https://www.st.com/en/development-tools/stsw-stm32102.html>



QUICK VIEW		RESOURCES		GET SOFTWARE	
<b>Product Specifications</b>					
Description		Version		Size	
	DB3529: STM32 Virtual COM port driver	1.0		116 KB	
<b>Legal</b>					
<b>License Agreement</b>					
Description		Version		Size	
	SLA0048: Mix Ultimate Liberty+OSS+3rd-party V1 - SOFTWARE LICENSE AGREEMENT	4.10		86 KB	
<b>GET SOFTWARE</b>					
Part Number ▲	Software Version ▼	Marketing Status ▼	Supplier ▼	Order from ST ▼	
STSW-STM32102	1.5.0	Active	ST	<a href="#">Get Software</a>	

Figura 9-1: Página web del software del driver.

- 2) Se descarga el fichero *stsw-stm32102.zip*. A continuación se ejecuta y se siguen las instrucciones de instalación del software.
- 3) Una vez instalado, ya podemos ejecutar el fichero *Decaranging.exe*.

## 9.2 Uso

En esta sección se explica cómo se utiliza el software de DecaWave. Con el programa ya instalado, los pasos a seguir son los siguientes:

Por un lado, se tiene un nodo configurado como ancla, el cual está conectado vía USB a un ordenador con la aplicación DecaRanging, y por otro lado se tiene un nodo configurado como tag.

A continuación, se emparejan los nodos. Entonces, el tag comienza a enviar blink a la espera de ser escuchado y respondido por otro nodo.

Se ha de configurar el ancla para que esta pueda escuchar los mensajes blink.

Para conseguir esto, se configuran los parámetros para que operen en el mismo canal y con la misma configuración.

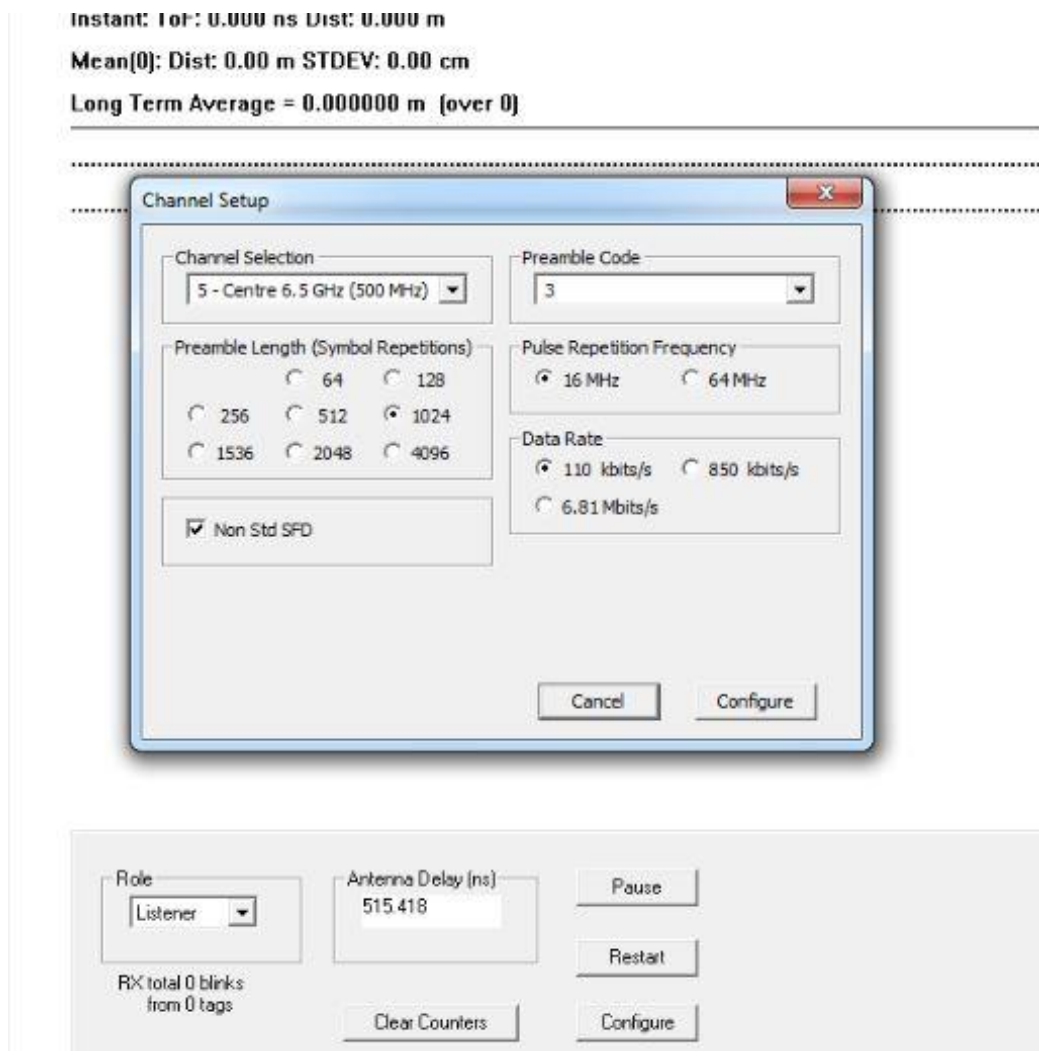


Figura 9-2: Configuración del canal de comunicación.

Como el tag opera con estos parámetros, se escoge esta configuración.

Además, como ilustra la figura 9-2, el ancla debe estar en modo escucha (listener) para mantenerse a la escucha esperando a ser emparejada.

Si se ha configurado correctamente, en pantalla se observará como llegan los mensajes blink. Una vez esto pase, se cambia el rol a 'anchor' (ancla). Se mostrará un mensaje por pantalla con la dirección del tag y se acepta para continuar.

Una vez se acepta, se termina el emparejamiento.

En la pantalla se pueden visualizar distintas opciones, por ejemplo, la figura 9-3 ilustra la gráfica de las muestras que se van recibiendo por pantalla:

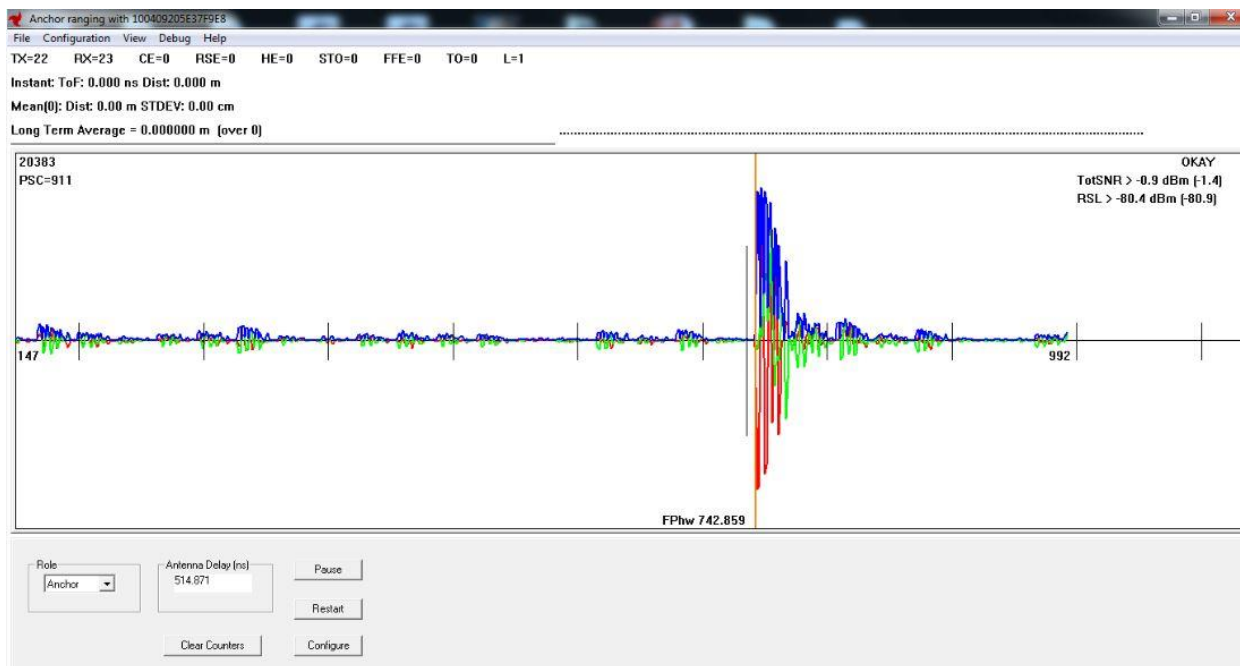


Figura 9-3: pantalla del software DecaWave con la opción de visualizar gráfica.

## 9.2.1 Logs

En esta subsección se explican los diferentes tipos de logs que el software Decaranging, de DecaWave, nos permite crear.

Para crear un log, se tiene en la barra superior de opciones una opción llamada 'Debug'. Aquí se encuentran los dos distintos logs posibles.

La primera opción disponible es *Log Channel Responses*: en esta opción el software recoge toda la información sobre las tramas enviadas y la respuesta del canal.

La segunda opción es *Log SPI Activity*: en ella se almacena todo el tráfico SPI.

Para este proyecto solo se requiere de los logs con la respuesta del canal, es decir, la primera opción.

Cuando se pulsa en cualquiera de las dos opciones, el software empieza el debug. Cuando se quiera terminar con él, se pulsa el botón *Stop Logging*. Hecho esto, se para el debug y el software genera un fichero con la información y se guarda en la carpeta raíz de la aplicación. El nombre por defecto con el que se guarda es xxxSPI.log para la segunda opción y xxxAccum.log para la primera.

Como en este proyecto solo se requiere de los logs de respuesta del canal, se describe resumidamente cómo es esta opción.

### 9.2.1.1 Log Channel Responses

En las tramas que se encuentran en este log, tenemos tanto las tramas recibidas como las enviadas. El intervalo de envío de tramas es variable y se puede configurar en la opción *Configuration* -> *Time Setting*.

La figura 9-4 ilustra un ejemplo de trama recogido en el log:

```
TX Frame Timestamp Raw = 85 D368E641
  Adding Antenna Delay = 0085 D368E641
18 Tx time = 8.995291178901868e+000
21 ED Rx time = 9.431259866426908e+000 8C4FD65EA9
21 ED Rx time(un) = 9.431260120192308e+000 8C4FD69E00
txdly 4041 rxdly 4041
RX DATA: 41ccedcade82f9375e20900310e8f9375e20090410210000e071
RX OK wInd(0735), HLP(0736.5313), PSC(1019), SLP(0000.0000), RC(008C 4FD65EA9), DCR(0), DCI(0), NTH(026C), T(86AB), RSL(-095.7267), FSL(-109.9741), RSMPL(11)
Accum Len 992
```

Figura 9-4: Ejemplo de trama recogido en el *Log Channel Responses*.

Las tramas van acompañadas de la respuesta impulsiva del canal en ese instante. Esta respuesta impulsiva se encuentra en una tabla con dos columnas: la columna izquierda se corresponde con los valores imaginarios y la columna derecha se corresponde con los valores reales. Esto se puede observar en la figura 9-5:

```
204, -260
263, -272
165, -69
55, 32
83, -15
19, 38
27, 81
4, -33
-48, -243
-7, -73
42, 7
476, 319
1521, 769
1645, 836
989, 742
-276, -277
-510, -1223
-602, -652
-1034, 514
-1232, 1341
-271, 1688
1270, 992
1868, 132
1576, -488
517, -792
-721, -454
-617, 401
276, 1411
155, 1244
-636, 505
-642, 278
-41, 423
-119, 927
-366, 795
-147, 317
191, 188
349, 380
171, 201
126, 91
296, 401
623, 638
461, 409
104, 82
-235, 11
-531, -233
-445, -464
-145, -436
96, 213
111, 729
402, 506
765, -6
```

Figura 9-5: Ejemplo de respuesta impulsiva en el *Log Channel Responses*.

De todos los datos que se pueden extraer del log, los necesarios para llevar a cabo este proyecto son la respuesta impulsiva del canal para el caso del RMS Delay Spread y los valores de FSL y RSL para el caso del ratio potencia primer camino frente a potencia total.



## 9.2.2 Exportar datos con MatLab/Octave

Ahora, se expone cómo poder exportar esos datos a MatLab para posteriormente poder tratarlos.

Se parte un fichero generado con *Log Channel Responses*, y en una consola de Linux con alguno de los scripts descritos en el Apéndice B, según el tipo de dato que queramos extraer del log, ejecutamos el script correspondiente.

Para invocar el script, se llama de la siguiente forma: **`./buscaFSL.sh {nombre fichero log}`**

Se ha puesto de ejemplo el script *buscaFSL.sh*, pero todos los scripts se llaman de la misma forma por línea de comandos.

Los scripts del Apéndice B requieren solamente de un parámetro, que será el nombre del log del que queremos extraer la información. Una vez consiga la información que se quiere, el mismo script generará un archivo nuevo donde guarda esa información, manteniendo así el archivo log intacto por si se requiere en un futuro de otra información relevante de ese archivo.

Una vez extraída la información necesaria, solo tenemos que volcar los datos en MatLab/Octave. La figura 9-6 contiene un extracto del código *Ratio.m*:

```
load('LOSsalidaRSL.txt');  
load('LOSsalidaFSL.txt');  
  
load('NLOSsalidaRSL.txt');  
load('NLOSsalidaFSL.txt');
```

Figura 9-6: Ejemplo de vuelco de datos en MatLab/Octave.

Un ejemplo de un fichero de salida resultante de la ejecución de un script es el que ilustra la figura 9-7:

```
-086.4157  
-086.9605  
-085.8004  
-087.4623  
-087.0470  
-086.6584  
-085.8802  
-085.9146  
-086.1013  
-087.7431  
-086.5418  
-085.9417  
-086.8421  
-086.9362  
-087.1155  
-087.8402  
-087.2093  
-087.1893  
-086.1872  
-087.1965  
-085.9844  
-086.4697  
-086.3126  
-086.4353  
-086.9079
```

Figura 9-7: Ejemplo de fichero de salida de script.



# 10

# A

## PÉNDICE B: FICHEROS SHELLSCRIPT

---

### 10.1 Ratio potencia primer camino frente a potencia total

Para calcular el ratio, debemos obtener los valores FSL y RSL del log generado por el software de DecaWave. Los códigos que aquí se presentan son una solución mediante ShellScript que genera un fichero con una lista de todos los valores FSL y RSL para luego exportar estos valores al código MatLab correspondiente al cálculo del ratio.

#### 10.1.1 buscaRSL.sh

Este código es el responsable de generar el fichero con la lista de los valores RSL de los logs:

```
#!/bin/bash
# -*- ENCODING: UTF-8 -*-
> salida
sudo cat $1 | grep -ao "RSL(.*)" >> salida
sed -i 's/RSL(/g' salida
sed -i 's/),/g' salida
sed -i 's/FSL.*/g' salida
sed -i 's/RSMPL.*/g' salida
```

#### 10.1.2 buscaFSL.sh

El siguiente código se encarga de generar el fichero con la lista de valores FSL de los logs:

```
#!/bin/bash
```

```
# -*- ENCODING: UTF-8 -*-  
> salida  
sudo cat $1 | grep -ao "FSL(.*)" >>salida  
sed -i 's/FSL(/g' salida  
sed -i 's/),/g' salida  
sed -i 's/RSMPL.*//g' salida  
sed -i 's/)//g' salida
```

## 10.2 RMS Delay Spread

Antes de calcular el RMS Delay Spread, necesitamos obtener los puntos de las gráficas que nos proporciona DecaWave a partir de los log, aquí se expone una solución en ShellScript donde se consigue extraer una lista con dichos puntos:

### 10.2.1 buscagrafica.sh

```
#!/bin/bash  
# -*- ENCODING: UTF-8 -*-  
clear  
> salidagrafica  
sudo cat $1 | grep -ao ".*,.*" >>salidagrafica  
sed -i 's/File:.*//g' salidagrafica  
sed -i 's/Mode.*//g' salidagrafica  
sed -i 's/RX.*//g' salidagrafica
```



## 11

## A

## PÉNDICE C: FICHEROS DE MATLAB/OCTAVE

---

### 11.1 Ratio potencia primer camino frente a potencia total

El código en el cual se encuentran todos los algoritmos utilizados en el ámbito de este ratio: algoritmo del ratio, importación de datos, probabilidad de error, etc. se encuentran recogidos en un fichero de MatLab/Octave llamado *Ratio.m*, el contenido del cual se expone en la siguiente subsección.

#### 11.1.1 Ratio.m

```
%Autor: Noel Cortés García
```

```
%Fichero Ratio.m donde se desarrolla el código referente al ratio potencia primer camino frente a potencia total.
```

```
clc
```

```
%Cargamos los datos obtenidos de los logs (cálculo umbral):
```

```
load('logs/LOSsalidaRSL.txt');
```

```
load('logs/LOSsalidaFSL.txt');
```

```
load('logs/NLOSsalidaRSL.txt');
```

```
load('logs/NLOSsalidaFSL.txt');
```

```
load('logs/NLOSlejossalidaRSL.txt');
```

```
load('logs/NLOSlejossalidaFSL.txt');
```

```
load('logs/paredsalidaRSL.txt');
load('logs/paredsalidaFSL.txt');

load('logs/LOS16RSL.txt');
load('logs/LOS16FSL.txt');

load('logs/LOS13RSL.txt');
load('logs/LOS13FSL.txt');

load('logs/LOS5RSL.txt');
load('logs/LOS5FSL.txt');

load('logs/LOS3RSL.txt');
load('logs/LOS3FSL.txt');

load('logs/paredvRSL.txt');
load('logs/paredvFSL.txt');

load('logs/paredhRSL.txt');
load('logs/paredhFSL.txt');

load('logs/pared3RSL.txt');
load('logs/pared3FSL.txt');

load('logs/pared4RSL.txt');
load('logs/pared4FSL.txt');

load('logs/arm2RSL.txt');
load('logs/arm2FSL.txt');

%Cargamos los datos obtenidos de los logs (prueba umbral):

load('logs/LOS14salidaRSL.txt');
load('logs/LOS14salidaFSL.txt');

load('logs/LOS8salidaRSL.txt');
load('logs/LOS8salidaFSL.txt');
```



```
load('logs/LOS4salidaRSL.txt');
load('logs/LOS4salidaFSL.txt');

load('logs/armsalidaRSL.txt');
load('logs/armsalidaFSL.txt');

load('logs/habsalidaRSL.txt');
load('logs/habsalidaFSL.txt');

load('logs/puertasalidaRSL.txt');
load('logs/puertasalidaFSL.txt');

%Convertimos de dBm a naturales los resultados de los logs según la fórmula del
manual de DecaWave:

RSL=10.^(((LOSsalidaRSL(:,1)+113.77)/10));
FSL=10.^(((LOSsalidaFSL(:,1)+113.77)/10));

NRSL=10.^(((NLOSsalidaRSL(:,1)+113.77)/10));
NFSL=10.^(((NLOSsalidaFSL(:,1)+113.77)/10));

LRSL=10.^(((NLOSlejossalidaRSL(:,1)+113.77)/10));
LFSL=10.^(((NLOSlejossalidaFSL(:,1)+113.77)/10));

PRSL=10.^(((paredsalidaRSL(:,1)+113.77)/10));
PFSL=10.^(((paredsalidaFSL(:,1)+113.77)/10));

RSL16=10.^(((LOS16RSL(:,1)+113.77)/10));
FSL16=10.^(((LOS16FSL(:,1)+113.77)/10));

RSL13=10.^(((LOS13RSL(:,1)+113.77)/10));
FSL13=10.^(((LOS13FSL(:,1)+113.77)/10));

RSL5=10.^(((LOS5RSL(:,1)+113.77)/10));
FSL5=10.^(((LOS5FSL(:,1)+113.77)/10));

RSL3=10.^(((LOS3RSL(:,1)+113.77)/10));
FSL3=10.^(((LOS3FSL(:,1)+113.77)/10));

RSLv=10.^(((paredvRSL(:,1)+113.77)/10));
```

```

FSLv=10.^(((paredvFSL(:,1)+113.77)/10));

RSLh=10.^(((paredhRSL(:,1)+113.77)/10));
FSLh=10.^(((paredhFSL(:,1)+113.77)/10));

RSLp3=10.^(((pared3RSL(:,1)+113.77)/10));
FSLp3=10.^(((pared3FSL(:,1)+113.77)/10));

RSLp4=10.^(((pared4RSL(:,1)+113.77)/10));
FSLp4=10.^(((pared4FSL(:,1)+113.77)/10));

RSLarm2=10.^(((arm2RSL(:,1)+113.77)/10));
FSLarm2=10.^(((arm2FSL(:,1)+113.77)/10));

RSL14=10.^(((LOS14salidaRSL(:,1)+113.77)/10));
FSL14=10.^(((LOS14salidaFSL(:,1)+113.77)/10));

RSL8=10.^(((LOS8salidaRSL(:,1)+113.77)/10));
FSL8=10.^(((LOS8salidaFSL(:,1)+113.77)/10));

RSL4=10.^(((LOS4salidaRSL(:,1)+113.77)/10));
FSL4=10.^(((LOS4salidaFSL(:,1)+113.77)/10));

RSLarm=10.^(((armsalidaRSL(:,1)+113.77)/10));
FSLarm=10.^(((armsalidaFSL(:,1)+113.77)/10));

RSLhab=10.^(((habsalidaRSL(:,1)+113.77)/10));
FSLhab=10.^(((habsalidaFSL(:,1)+113.77)/10));

RSLpuerta=10.^(((puertasalidaRSL(:,1)+113.77)/10));
FSLpuerta=10.^(((puertasalidaFSL(:,1)+113.77)/10));

%Inicializamos variables ratios:

ratioLOS=0;
ratioNLOS=0;
ratioNLOSlejos=0;
ratiopared=0;
ratioLOS16=0;

```

```
ratioLOS14=0;
ratioLOS13=0;
ratioLOS5=0;
ratioLOS4=0;
ratioLOS3=0;
ratioLOS8=0;
ratioarm=0;
ratiohab=0;
ratiopuerta=0;
ratiov=0;
ratioh=0;
ratio3=0;
ratio4=0;
ratioarm2=0;

%Calculamos ratios (umbral):

for i=1:length(LOSsalidaRSL)
    ratioLOS(i)=(FSL(i)/RSL(i));
endfor

for j=1:length(NLOSsalidaRSL)
    ratioNLOS(j)=(NFSL(j)/NRSL(j));
endfor

for k=1:length(NLOSlejossalidaRSL)
    ratioNLOSlejos(k)=(LFSL(k)/LRSL(k));
endfor

for l=1:length(paredsalidaRSL)
    ratiopared(l)=(PFSL(l)/PRSL(l));
endfor

for s=1:length(RSL16)
    ratioLOS16(s)=(FSL16(s)/RSL16(s));
endfor

for a=1:length(RSL13)
    ratioLOS13(a)=(FSL13(a)/RSL13(a));
endfor
```

```
for za=1:length(RSL5)
    ratioLOS5(za)=(FSL5(za)/RSL5(za));
endfor

for d=1:length(RSL3)
    ratioLOS3(d)=(FSL3(d)/RSL3(d));
endfor

for aa=1:length(RSLp3)
    ratioop3(aa)=(FSLp3(aa)/RSLp3(aa));
endfor

for bb=1:length(RSLp4)
    ratioop4(bb)=(FSLp4(bb)/RSLp4(bb));
endfor

for cc=1:length(RSLarm2)
    ratioarm2(cc)=(FSLarm2(cc)/RSLarm2(cc));
endfor

for dd=1:length(RSLh)
    ratioh(dd)=(FSLh(dd)/RSLh(dd));
endfor

for ee=1:length(RSLv)
    ratiov(ee)=(FSLv(ee)/RSLv(ee));
endfor

%Calculamos ratios (prueba umbral):

for m=1:length(RSL14)
    ratioLOS14(m)=(FSL14(m)/RSL14(m));
endfor

for n=1:length(RSL8)
    ratioLOS8(n)=(FSL8(n)/RSL8(n));
endfor
```

```
for o=1:length(RSL4)
    ratioLOS4(o)=(FSL4(o)/RSL4(o));
endfor

for p=1:length(RSLarm)
    ratioarm(p)=(FSLarm(p)/RSLarm(p));
endfor

for q=1:length(RSLhab)
    ratiohab(q)=(FSLhab(q)/RSLhab(q));
endfor

for r=1:length(RSLpuerta)
    ratiopuerta(r)=(FSLpuerta(r)/RSLpuerta(r));
endfor

%Pintamos histogramas

umbral_los=horzcat(ratioLOS, ratioLOS16, ratioLOS13, ratioLOS5, ratioLOS3);
umbral_nlos=horzcat(ratioNLOS, ratioNLOSlejos, ratiopared, ratiop3, ratiop4,
ratioh, ratiov, ratioarm2);

prueba_los=horzcat(ratioLOS14, ratioLOS8, ratioLOS4);
prueba_nlos=horzcat(ratioarm, ratiohab, ratiopuerta);

aux=horzcat(prueba_los, prueba_nlos);

control=1;
valores_hist1=0;
valores_hist2=0;
corrupta=0;
tramas_corruptas=0;

for t=1:length(prueba_los)
    if (prueba_los(t)>0 & prueba_los(t)<1)
        valores_hist1(control-corrupta)=prueba_los(t);
```

```
        control++;
    else
        corrupta++;
    end
endfor

control=1;
tramas_corruptas=tramas_corruptas+corrupta;
corrupta=0;

for s=1:length(prueba_nlos)
    if (prueba_nlos(s)>0 & prueba_nlos(s)<1)
        valores_hist2(control-corrupta)=prueba_nlos(s);
        control++;
    else
        corrupta++;
    end
endfor

control=1;
tramas_corruptas=tramas_corruptas+corrupta;
corrupta=0;
valores_umbral_los=0;
valores_umbral_nlos=0;

for u=1:length(umbral_los)
    if (umbral_los(u)>0 & umbral_los(u)<1)
        valores_umbral_los(control-corrupta)=umbral_los(u);
        control++;
    else
        corrupta++;
    end
endfor

control=1;
tramas_corruptas=tramas_corruptas+corrupta;
corrupta=0;

for v=1:length(umbral_nlos)
    if (umbral_nlos(v)>0 & umbral_nlos(v)<1)
```

```
    valores_umbral_nlos(control-corrupta)=umbral_nlos(v);
    control++;
else
    corrupta++;
end
endfor

tramas_corruptas=tramas_corruptas+corrupta;

figure(3)
hist(valores_umbral_nlos, 50, 'b')
ylabel("N° de muestras")
hold on
hist(valores_umbral_los, 40, 'y')
xlabel("Ratio potencia primer camino frente a potencia total")
hold off

figure(4)
hist(valores_hist2, 60, 'b')
ylabel("N° de muestras")
hold on
hist(valores_hist1, 20, 'y')
xlabel("Ratio potencia primer camino frente a potencia total")
hold off

%Cálculo de la probabilidad de error del ratio:

resultado_priori=vertcat(ones(173,1), zeros(205,1));
resultado_ratio=0;
cadena=0;
falso_LOS=0;
falso_NLOS=0;

for ac=1:length(aux)
    if aux(ac)>0.15
        resultado_ratio(ac)=1; %1 para LOS
    else
        resultado_ratio(ac)=0; %0 para NLOS
    endif
endfor
```

```
endfor

%Probabilidad de que se detecte LOS sabiendo que realmente es NLOS:

for bc=1:length(resultado_ratio)

    if ((resultado_ratio(bc)== 1) & (resultado_priori(bc)==0))
        falso_LOS++;
    endif
endfor

%Probabilidad de que se detecte NLOS sabiendo que realmente es LOS:

for c=1:length(resultado_ratio)
    if ((resultado_ratio(c)== 0) & (resultado_priori(c)==1))
        falso_NLOS++;
    endif
endfor

%Inicializamos variables de probabilidad:

pLOS_NLOS=0;
pNLOS_LOS=0;

%Calculamos las probabilidades:

pLOS_NLOS=(falso_LOS/204) %205 - una trama corrupta que se encuentra en este
grupo
pNLOS_LOS=(falso_NLOS/173)
```



## 11.2 RMS Delay Spread

Para todos los cálculos que se llevan a cabo para desarrollar el algoritmo del RMS Delay Spread, así como aquellos referidos a probabilidad de error o tratamiento de las gráficas importadas de los log de DecaWave, se ha escrito el script de MatLab/Octave llamado *Identifica.m*, el cual se expone en la subsección que viene a continuación.

### 11.2.1 Identifica.m

```
% Autor: Noel Cortés García
%Programa identificación de medidas UWB mediante RMS Delay Spread: LOS o NLOS
%%

%Limpiamos la ventana:
clc
clear all

%Variable para recorrer los 992 puntos de cada señal recogida en DecaWave
t=1:992;

%Inicializamos variables:
i=1;
pulso=0;
valor=-200;
tramas_corruptas=0;

%Muestras tomadas para definir el umbral:           Acumulado de muestras para guía:

load('logs/LOSsalidagrafica.txt'); %43 muestras           43
load('logs/LOS16grafica.txt'); %56 muestras               99
load('logs/LOS13grafica.txt'); %63 muestras               162
load('logs/LOS5grafica.txt'); %54 muestras                216
load('logs/LOS3grafica.txt'); %61 muestras                277
load('logs/NLOSsalidagrafica.txt'); %51 muestras          328
```

```

load('logs/Lejossalidagrafica.txt'); %63 muestras      391
load('logs/paredsalidagrafica.txt'); %46 muestras      437*
load('logs/pared3grafica.txt'); %54 muestras          491*
load('logs/pared4grafica.txt'); %50 muestras          541
load('logs/paredhgrafica.txt'); %48 muestras          589*
load('logs/paredvgrafica.txt'); %69 muestras          658
load('logs/arm2grafica.txt'); %90 muestras            748

%Muestras tomadas para poner a prueba el umbral:

load('logs/LOS14salidagrafica.txt'); %58 muestras      806
load('logs/LOS8salidagrafica.txt'); %55 muestras      861
load('logs/LOS4salidagrafica.txt'); %60 muestras      921
load('logs/armsalidagrafica.txt'); %74 muestras      995*
load('logs/habsalidagrafica.txt'); %66 muestras     1061
load('logs/puertasalidagrafica.txt'); %65 muestras   1126*

%Array con todos los valores de los logs:

concatena=vertcat(LOSsalidagrafica,  LOS16grafica,  LOS13grafica,  LOS5grafica,
LOS3grafica, NLOSsalidagrafica,
Lejossalidagrafica,  paredsalidagrafica,  pared3grafica,  pared4grafica,
paredhgrafica, paredvgrafica, arm2grafica,
LOS14salidagrafica,  LOS8salidagrafica,  LOS4salidagrafica,  armsalidagrafica,
habsalidagrafica, puertasalidagrafica);

%Cada medida contiene 992 puntos a representar en la gráfica:
final=(length(concatena))/992;
final
grafica=0;

%Variables para capturar cada columna de los logs por separado
x3=(concatena(:,1));
x4=(concatena(:,2));

%Raíz de la suma de los cuadrados para obtener el módulo
x2=sqrt(x3.^2+x4.^2);

%Variable tau_rms
tau_rms=0;

%Bucle principal que recorre todas las medidas capturadas en los logs

```

```
for e=1:final

    %En este bucle se obtienen los valores de los 992 puntos de la grafica de cada
    medida para realizar los cálculos posteriormente:
    for f=1:992

        grafica(f)=x2(f+((e-1)*992));

    endfor

    pulso=0;
    valor=-200;

    %Busqueda de pico máximo:
    i=1;
    for i=1:992
        if (valor<grafica(i))
            if x2(i) != -Inf
                valor=grafica(i);
                pulso=i;
            end
        end
    endfor

    valor; %valor del pico máximo
    pulso; %posición del pico máximo

    if(valor>1200)%%Condición descarte de tramas corruptas del bucle (caso
    particular para el conjunto de muestras
        %del proyecto)

    %RMS Delay Spread

    if(pulso!=0)

        p=pulso+2; %Como pulso es la posición del pico máximo, la posición pulso+1
        no puede ser máximo relativo
        ultimo=0; %Posición del último máximo que sobrepasa el umbral
```

```

    umbral=valor/9; %Umbral definido mediante la observación y el estudio del
conjunto de muestras

```

```

    %En este bucle se recorre la gráfica desde el pico máximo hasta el final
para encontrar el primer máximo relativo

```

```

    %que no sobrepase el umbral establecido. Cuando lo encuentra, se sale del
bucle.

```

```

    %La variable ultimo recoge la posición del máximo relativo más alejado del
punto máximo que sobrepasa el umbral.

```

```

while p<=991
    if grafica(p-1)<grafica(p)
        if grafica(p+1)<grafica(p)
            if grafica(p)>=umbral
                ultimo=p;
            else
                p=1000;
            endif
        endif
    endif
endif

```

```

    p++;
endwhile

```

```

secuencia=0;
pos_secuencia=0;

```

```

    %Vector secuencia, que determina el valor y la posición de todos los puntos
de la señal desde el punto

```

```

    %máximo hasta el ultimo

```

```

for h=pulso:ultimo
    secuencia(h-pulso+1)=grafica(h);
    pos_secuencia(h-pulso+1)=h-pulso+1;
endfor

```

```

    %Parámetros de las ecuaciones de tau_m y tau_rms

```

```

media_todo=0;
aux_media=0;
media=0;

```

```
div=0;

%Resolución de los parámetros de las ecuaciones de tau_m y tau_rms

for z=1:length(secuencia)
    media=media+(((secuencia(z))^2)*((pos_secuencia(z))^2));%numerador
    div=div+((secuencia(z))^2); %denominador

    media_todo= media_todo+((secuencia(z)^2)*(z)); %numerador tau_m
    aux_media=aux_media+((secuencia(z))^2); %denominador tau_m
endfor

%Cálculo de tau_m y tau_rms
tau_m=media_todo/aux_media;
tau_rms(e-tramas_corruptas)=sqrt((media/div)-(tau_m^2));

end

else
    %Si no cumple el requisito, se suma a las tramas corruptas
    tramas_corruptas++;
    %e
    % figure(e)
    % plot(grafica)

endif

clc
e

endfor

control=1;
umbral_los=0;
umbral_nlos=0;

y=1;

%Recorrido de tau_rms para rellenar los arrays de los histogramas.
```

```
%Tomando el acumulado de muestras del inicio del código y la situación de las  
tramas corruptas,
```

```
% se acotan los bucles de forma correcta.
```

```
%Bucle que recorre las muestras LOS de cálculo del umbral:
```

```
for t=1:276  
    umbral_los(control)=tau_rms(t);  
    control++;  
endfor
```

```
control=1;
```

```
%Bucle que recorre las muestras NLOS de cálculo del umbral:
```

```
for s=277:743  
    umbral_nlos(control)=tau_rms(s);  
    control++;  
endfor
```

```
control=1;  
prueba_los=0;  
prueba_nlos=0;
```

```
%Bucle que recorre las muestras LOS para poner a prueba el umbral:
```

```
for v=744:916  
    prueba_los(control)=tau_rms(v);  
    control++;  
endfor
```

```
control=1;
```

```
%Bucle que recorre las muestras NLOS para poner a prueba el umbral:
```

```
for w=917:length(tau_rms)  
    prueba_nlos(control)=tau_rms(w);  
    control++;  
endfor
```

```
%Histograma de muestras de cálculo del umbral
figure(1)
hist(umbral_nlos, 80, 'b')
ylabel("N° de muestras")
xlabel("RMS Delay Spread (ns)")
hold on
hist(umbral_los, 40, 'y')
hold off

%Histograma de muestras para poner a prueba el umbral
figure(2)
hist(prueba_nlos, 60, 'b')
ylabel("N° de muestras")
xlabel("RMS Delay Spread (ns)")
hold on
hist(prueba_los, 25, 'y')
hold off

%Código del cálculo de la probabilidad de error:

%Las longitudes de los arrays se obtienen del acumulado de muestras y las tramas
corruptas que se encuentran

%Para la probabilidad de error que se busca, es necesario tener un vector que
nos diga cómo es realmente la señal a priori

resultado_priori=vertcat(ones(173,1), zeros(200,1));

resultado_rms=0;
acierto=0;
error=0;

%En este bucle obtenemos el resultado de la identificación de la señal mediante
el RMS Delay Spread

for a=744:length(tau_rms)

    if (tau_rms(a)<=6) %Umbral obtenido mediante observación de la gráfica
        resultado_rms(a-743)=1;
    else
        resultado_rms(a-743)=0;
    end
end
```

```
endif

endfor

falso_LOS=0;
falso_NLOS=0;

%Probabilidad de que se detecte LOS sabiendo que realmente es NLOS:

for bc=1:length(resultado_rms)
    if ((resultado_rms(bc)== 1) & (resultado_priori(bc)==0))
        falso_LOS++;
    endif
endfor

%Probabilidad de que se detecte NLOS sabiendo que realmente es LOS:

for bd=1:length(resultado_rms)
    if ((resultado_rms(bd)== 0) & (resultado_priori(bd)==1))
        falso_NLOS++;
    endif
endfor

%Inicializamos variables de probabilidad:

pLOS_NLOS=0;
pNLOS_LOS=0;

%Calculamos las probabilidades:

pLOS_NLOS=(falso_LOS/200)
pNLOS_LOS=(falso_NLOS/173)
```





# REFERENCIAS

- [1] Wikipedia, «Ultrawideband,» Available: <https://es.wikipedia.org/wiki/Ultrawideband>.
- [2] DecaWave, «DW1000 DATASHEET» Available: [https://www.decawave.com/wp-content/uploads/2018/09/dw1000\\_datasheet\\_v2.17.pdf](https://www.decawave.com/wp-content/uploads/2018/09/dw1000_datasheet_v2.17.pdf)
- [3] DecaWave, «DW1000 PRODUCT BRIEF» Available: [https://www.decawave.com/wp-content/uploads/2018/09/dw1000-product-brief\\_0.pdf](https://www.decawave.com/wp-content/uploads/2018/09/dw1000-product-brief_0.pdf).
- [4] DecaWave, «DW1000 USER MANUAL» Available: [https://www.decawave.com/wp-content/uploads/2018/09/dw100020user20manual\\_0.pdf](https://www.decawave.com/wp-content/uploads/2018/09/dw100020user20manual_0.pdf).
- [5] DecaWave, «EVK1000 USER MANUAL» Available: [https://www.decawave.com/wp-content/uploads/2018/09/evk1000\\_user\\_manual.pdf](https://www.decawave.com/wp-content/uploads/2018/09/evk1000_user_manual.pdf).
- [6] Wikipedia, «Distribución Normal,» Available: [https://es.wikipedia.org/wiki/Distribuci%C3%B3n\\_normal](https://es.wikipedia.org/wiki/Distribuci%C3%B3n_normal).
- [7] Wikipedia, «Número complejo,» Available: [https://es.wikipedia.org/wiki/N%C3%BAmero\\_complejo](https://es.wikipedia.org/wiki/N%C3%BAmero_complejo).
- [8] Gaussian Waves, «RMS Delay Spread» <https://www.gaussianwaves.com/tag/rms-delay-spread/>
- [9] Gentile, C.; Alsindi, N.; Raulefs, R. y Teolis, C. «Geolocation Techniques: Principles and Applications», Springer 2012
- [10] Pahlavan, K; Akgül, F. O.; Heidari, M; Hatami, A.; Elwell, J.M. y Tingley, R. D. «Indoor geolocation in the absence of direct path», 2009
- [11] Verdone, R.; Conti, A., Dardari, D. y Mazzini, G. «Wireless sensor and actuator networks», AP 2008
- [12] De la Calle, L. «Evaluación de dispositivos UWB para localización en interiores con medidas de distancia», Trabajo Fin de Grado, 2017
- [13] DecaWave, «Decaranging PC User Manual» Available: <https://www.decawave.com/software/>



