UNIVERSIDAD DE EXTREMADURA

ESCUELA DE INGENIERÍAS INDUSTRIALES

SIMULACIÓN PARALELA DE LA EMISIÓN DE WIFI CON TRAZADO DE RAYOS

TRABAJO PRESENTADO PARA OPTAR AL TITULO DE

MÁSTER UNIVERSITARIO EN SIMULACIÓN EN CIENCIAS E INGENIERÍA

POR

DANIEL ALFONSO DURÁN GARCÍA

BADAJOZ, SEPTIEMBRE DE 2021

TRABAJO FIN DE GRADO/ TRABAJO FIN DE MÁSTER/ PROYECTO FIN DE CARRERA

Autor:

DANIEL ALFONSO DURÁN GARCÍA

Director o Directores:

CARLOS JAVIER GARCÍA ORELLANA

Tribunal de evaluación:

MIGUEL ÁNGEL JARAMILLO MORÁN

JUAN JESÚS RUIZ LORENZO

JOSE LUIS AUSÍN SÁNCHEZ

Índice

1. Introducción y objetivos **2**

2. Fundamento teórico y metodología **4**
2.1. Aproximación de la emisión electromagnética a rayos 4
2.1.1. Algoritmos de trazado de rayos 6
2.2. Geometría del trazado de rayos 8
2.2.1. Rayos 8
2.2.2. Paredes 8
2.2.3. Receptores 13
2.3. Electromagnetismo en los impactos 15
2.4. Antenas 18

3. Simulación en serie y en paralelo **22**
3.1. Elementos comunes 22
3.2. Implementación en serie 27
3.3. Implementación en paralelo 27

4. Entorno de pruebas **31**
4.1. Robot 31
4.1.1. Trayectoria 34
4.2. Escenarios 35
4.2.1. Escenario A 36
4.2.2. Escenario B 36
4.2.3. Escenario C 37

5. Resultados **38**
5.1. Resultados en las medidas y simulaciones 38
5.1.1. Escenario A 38
5.1.2. Escenario B 41
5.1.3. Escenario C 43
5.2. Aceleración del cómputo en paralelo 45
5.2.1. Discusión de los resultados 46

6. Conclusiones **48**

7. Bibliografía **50**

1. Introducción y objetivos

Las redes inalámbricas de WiFi se han popularizado en la última década hasta el punto de ser un protocolo barato de usar más a allá de su uso original, con hardware compatible en multitud de dispositivos.

Su diseño solo contempla su uso en entornos interiores, donde existirán infinitas combinaciones de configuraciones: distintos tamaños de salas, distintas posiciones de mobiliario u objetos y con infinidad de materiales distintos.

Para poder conocer con seguridad el desepeño de estas ondas es necesario conocer cómo se propagan en este tipo de escenarios, tarea que puede llegar a ser imposible con técnicas analíticas o estadísticas.

Para poder conocer las características de la radiación en las zonas de interés surge la idea de usar la aproximación de las ondas a un rayo que se deplaza con el frente de onda, de modo que es posible conocer su recorrido teniendo en cuenta los posibles rebotes que se puedan producir.

Esta técnica ha sido tradicionalmente usada en otros campos de la física como la óptica o en aplicaciones acústicas. En el campo de las ondas de radio su aplicación se ha aplicado principalmente para el desarrollo y posicionamiento de antenas de telefonía, principalmente para su uso en ciudades. En estos escenarios, los edificios harán las veces de obstáculos de modo que en ciertas zonas la señal puede ser más débil.

La simulaciones con trazado de rayos buscan encontrar este tipo de zonas antes de su implantación, de modo que sea posible corregir las posibles deficiencias y conseguir un posicionamiento óptimo de las antenas usadas.

En este trabajo se plantea su uso en un escenario local, en interiores. Como se comentaba al inicio, la señal de WiFi encontrará en los entornos donde se usa obstáculos en forma de paredes o grandes objetos que impidan su avance.

Esto generará efectos de propagación multicamino, donde los rebotes de la señal en las paredes y obstáculos que se encuentren en su camino harán que en ciertos puntos lleguen, además de la señal original, una cierta potencia adicional fruto de estos rebotes. También encontraremos casos sin visión directa entre emisor y receptor, donde la señal recibida será exclusivamente la de estos rebotes.

La mayor ventaja de esta aproximación respecto a otros métodos de resolución de la ecuación de ondas es la posibilidad de paralelización. Si se consideran los rayos independientes entre sí –es decir, se obvian los posibles efectos de interferencias– se obtiene el caso idóneo de paralelización de código, donde no se necesitan barreras ni niguna otra técnica de programación paralela que pueda ralentizar la ejecución.

Por ello, en este trabajo se plantea el uso de GPUs –*Graphic Processing Unit*–. Este tipo de dispositivos cuentan con un gran número de núcleos ya que estan planteados para tareas muy paralelizables, por lo que son idóneas para este caso.

La programación en ese caso toma ciertas diferencias respecto a la programción para una CPU. Existen diferencias en la manera de tratar la ejecución y la memoria que requerirán un cierto cuidado a la hora de pasar de un dispositivo a otro.

De forma concreta, los objetivos de este Trabajo de Fin de Máster serán

- Diseñar e implementar un simulador de emisión de rayos que contemple su trayectoria en un entorno local.
- Implementar y estudiar la aceleración de la ejecución de la simulación en GPUs.
- Obtener datos en un entorno real con la ayuda de un robot autónomo y compararlos con los valores obtenidos en la simulación.

Con estos objetivos se busca aplicar las competencias adquiridos en el Máster en Simulación en Ciencias e Ingeniería en el ámbito de simulación de fenómenos naturales aplicando los conocimentos adquiridos sobre programación, especialmente la programación en paralelo.

2. Fundamento teórico y metodología

La proliferación del uso de ondas electromagnéticas como sustitución de conexiones cableadas ha aumentado su uso de forma significativa en las últimas décadas, especialmente entornos interiores.

Determinar la propagación de este tipo de ondas en estos escenarios resulta una parte fundamental en el estudio de su rendimiento y abre la puerta a aplicaciones adicionales además de la comunicación, como pueden ser algoritmos de posicionamiento en interiores.

De forma teórica existen técnicas de resolución de ecuaciones diferenciales aplicadas al campo electromagnético como el FDTD –*Finite-difference time-domain method*–, especialmente modificadas para adaptarse a la relación entre campos eléctrico y magnético.[1]

Este tipo de métodos numéricos, aunque precisos, requieren un alto coste computacional. Estan planteados como métodos generales, aplicables a cualquier tipo de onda electromagnética por lo que se pueden tomar aproximaciones que relajen los requisitos.

Teniendo en cuenta que las ondas a tratar en comunicaciones están en un régimen de alta frecuencia aparecen métodos que aproximan las ondas a rayos unidimensionales que se propagan en el espacio interactuando con todo tipo de objetos.

2.1. Aproximación de la emisión electromagnética a rayos

La aproximación de la emisión de ondas como propagación de rayos resulta una idea intuitiva ya explorada en otros ámbitos de la física relacionados con propagación de ondas como pueden ser fenómenos sismológicos o acústicos.

Dentro del ámbito de campos electromagnéticos, esta aproximación ha sido tradicionalmente usada en la óptica geométrica en aplicaciones como diseño de lentes o espejos.

En el caso de ondas electromagnéticas de forma general –más allá del ejemplo anterior de ondas de luz–, su uso principal es el de la propagación de ondas de radio. La proliferación de su uso desde el siglo pasado ha requerido técnicas de análisis para la arquitectura de sistemas de comunicación inalámbricos, de modo que sea posible determinar el camino seguido por las ondas de radio y tener comunicaciones fiables en condiciones sin visión directa.

Los primeros estudios aparecen en la estregia para aprovechar la capa de la ionosfera de la atmósfera para la comunicación entre puntos de la Tierra sin visión directa por la geometría esférica.

En estos casos los distintos índices de refracción modificarán la trayectoria antes y después de los rebotes buscados complicando la predicción del camino seguido de forma general. Por ello, se usa la aproximación de rayos para determinar las posiciones de los receptores y los ángulos de emisión requeridos.

La misma estrategia es seguida en propagación de ondas de telefonía móvil. En este caso el problema a resolver es comprobar las zonas con una mayor atenuación en zonas con una alta densidad urbanística, de modo que sea posible colocar emisores en posiciones óptimas de modo que estas zonas tengan el menor área posible.

En este trabajo se retoma esta aproximación en un entorno local. La estrategia de emisión de rayos no es escasa en la bibliografía en este ámbito: al igual que en los casos expuestos anteriormente, la parametrización a priori de los obstáculos en la propagación hace prácticamente imposible aplicar métodos analíticos o estadísticos para evaluar de forma fiable la recepción de las ondas emitidas.

La correcta determinación de la propagación en entornos controlados habilita la predicción con una alta precisión del rendimiento de los receptores en cualquier zona del mismo, lo que permite no solo una correcta colocación de los mismos –en caso de que se quiera garantizar una correcta recepción en cualquier zona– si no que abre la puerta a posibles aplicaciones como las de posicionamiento local.

La alternativa a la simulación de esta propagación es la medición sobre el terreno de los valores de la onda, bien eligiendo ciertos puntos de forma homogénea o buscando lugares concretos de interés. Esta técnica, conocida como *fingerprinting*, es muy lenta y obliga a desplazarse a la zona de interés, por lo que la opción de realizar una simulación presenta una gran ventaja a la hora de ahorrar tiempo y costes, además de permitir una precisón mayor pudiendo obtener los valores de recepción en cualquier punto de la zona a medir.[2]

La aproximación de ondas electromagnéticas a rayos proviene del concepto de frente de ondas, en el que se considera un cierto plano de propagación de las ondas. Según las ecuaciones de Maxwell los campos eléctrico y magnético son perpendicualares a este frente, y con ellos aparece el vector de Poynting.[1]

Este vector, definido como el producto vectorial de ambos campos, contiene en su módulo la potencia de la onda –tomando medias temporales–, y en su dirección la dirección de propagación. En ausencia de cambios en el índice de refracción del medio –es decir, en medios

homogéneos–, la dirección del frente de ondas y del vector de Poyinting no se modificará, así que la onda seguirá su trayectoria en línea recta.

Es este vector el utilizado para modelizar los rayos, de modo que cumplen tres características:

- Los rayos se transmitirán en línea recta en un medio homogéneo.
- Los rayos contienen información sobre la potencia de la onda transmitida.
- Los rayos sufrirán fenómenos de reflexión y refracción a su encuentro con los obstáculos del medio.

2.1.1. Algoritmos de trazado de rayos

En general, todos los algoritmos buscarán encontrar la manera de determinar impactos entre los rayos y los obstáculos del entorno, pero existen distintas estrategias para determinar estas colisiones.

Divisiones uniformes

En esta estrategia se busca la división de la zona de interés en cuadrados o cubos de forma unifome.

Con estas divisiones se busca cuál es la posición del rayo en cada momento de modo que se puedan determinar las propiedades buscadas en cada división, pudiendo obtener dichos valores en cada parte del mapa.

Si la geometría de la zona no es regular es posible definir zonas con distintos tamaños y así poder tener zonas con una granularidad mayor.

Bounding box

Otra de las técnicas viene heredada de la resolución del problema de colisiones habitualmente usando en videojuegos.

Aquí se busca modelizar los objetos con un cierto volumen, de forma que es posible determinar la existencia de una cierta colisión en caso de que alguno de ellos llegan a compartir cierto espacio.

Para mejorar el rendimiento es común usar una jerarquía de modo que sea posible agrupar los objetos en cajas mayores. Así, se buscará en primer lugar cuál de estas *bounding boxes* mayores será impactada, y luego iterar sobre sus componentes, eliminando la necesidad de comprobar todos los objetos de la escena y ahorrando coste computacional.

En este caso solo se encontraran estas colisiones entre rayos y obstáculos, pero no se registrará su paso por el espacio libre. Para ello aparece la idea de usar otros obstáculos como receptores.

El concepto de receptor en este caso no diferirá del del obstáculo en su modelización: será un elemento del mismo tipo, pero no generará rebotes. En el caso de que un rayo impacte con él, se registrará su potencia al colisionar y se dejará evolucionar al rayo incidente sin modificarlo.

Colisiones con mallas

La modelización de forma habitual de modelaje en tres dimensiones cuando la geometría no es regular consiste en formar un mallado de triángulos. Siendo una estretegia habitual y ampliamente usada en entornos de diseño, surgen implementaciones que aprovechan este tipo de modelos.

En estos casos la búsqueda de colisiones se produce buscando el punto de intersección entre la recta del rayo y el plano en el que alguno de los triángulos está contenido, comprobando más tarde sus límites. También es habitual el uso de coordenadas baricéntricas, que no dependen de la orientación del triángulo en el espacio. [3]

Esta estrategia ha sido la utilizada en esta simulación. En lugar de triángulos, en este caso se han usado rectángulos, aunque la base es la misma. En secciones posteriores se abodará la geometría necesaria para estos casos.

Al igual que en el caso del *bounding box*, será necesario colocar receptores en aquellos lugares de interés de modo que sea posible registrar los parámetros necesarios en ciertos puntos del mapa.

2.2. Geometría del trazado de rayos

2.2.1. Rayos

Los rayos a evaluar son el elemento básico a caracterizar. Su composición es muy simple: constan de un punto de origen y una dirección. En su transcurso se encontrarán con las paredes del mapa, contra las que rebotarán para seguir su recorrido en la zona de interés.

Para poder modelizar este comportamiento se considera el rayo como una recta. En este caso el origen O será un punto de esta recta, con su dirección siendo el vector director $\mathbf{d}$ de la misma de modo que sigue la ecuación

$$O + t\mathbf{d} \tag{2.1}$$

con $t \in \mathbb{R}$.

2.2.2. Paredes

Las paredes serán planos definidos con cuatro puntos como extremos, a partir de los cuáles se calcula su vector normal. Así, los puntos de intersección de las rectas con alguno de estos planos serán los puntos donde los rayos golpearán las paredes, que servirán de origen para los rayos trasmitidos y reflejados que se generen.

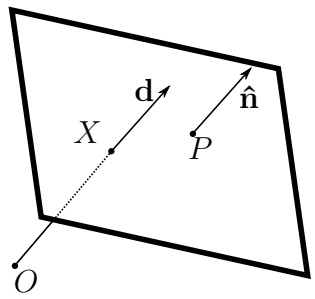


Figura 2.1: Esquema de la intersección entre rayo y plano.

Para determinar cuál es este punto de impacto partimos de la consideración de que el vector normal del plano –denominado aquí $\hat{\mathbf{n}}$– será perpendicular a cualquier vector contenido en dicho plano, en este caso el definido como diferencia entre el punto de intersección X y el punto donde está definida la normal[1] P de tal forma que su producto escalar es nulo

$$(X - P) \cdot \hat{\mathbf{n}} = 0 \tag{2.2}$$

La Figura 2.1 contiene un esquema de dicha disposición, donde se puede comprobar la perpendicularidad de este vector.

[1]Esta consideración proviene del caso de una superficie general; en este caso la normal será la misma independientemente de dónde se defina, pudiendo elegir cualquier punto del plano.

X es un punto de la recta, por lo que debe cumplir, para un cierto t_i

$$X = O + t_i\mathbf{d} \tag{2.3}$$

que es posible incluir en (2.2) de forma que

$$((O + t_i\mathbf{d}) - P) \cdot \hat{\mathbf{n}} = 0 \tag{2.4}$$

que se puede manipular para obtener t_i

$$\begin{aligned} ((O + t_i\mathbf{d}) - P) \cdot \hat{\mathbf{n}} &= 0 \\ (O - P) \cdot \hat{\mathbf{n}} + t_i\mathbf{d} \cdot \hat{\mathbf{n}} &= 0 \\ t_i\mathbf{d} \cdot \hat{\mathbf{n}} &= -(O - P) \cdot \hat{\mathbf{n}} \\ t_i &= -\frac{(O - P) \cdot \hat{\mathbf{n}}}{\mathbf{d} \cdot \hat{\mathbf{n}}} \\ t_i &= \frac{(P - O) \cdot \hat{\mathbf{n}}}{\mathbf{d} \cdot \hat{\mathbf{n}}} \end{aligned} \tag{2.5}$$

Con este valor es posible ahora usar la Ec. (2.3) para obtener las coordenadas del punto de impacto, pero no en cualquier caso.

Es necesario tener varias consideraciones a la hora de determinar t_i. La primera de ellas es que es posible obtener un valor negativo: al modelizar el rayo como una recta se abre la posibilidad de encontrar un punto de intersección en la dirección contraria al vector director, por lo que consideraremos que no hay intersección si $t_i < 0$.

Otra de las posibilidades es que la recta sea paralela al plano, de tal forma que no exista un punto de intersección. Si esto ocurre, el producto escalar del denominador de la Ec. (2.5) tomará un valor nulo, por lo que es necesario evitar la operación.

Es más, es posible optimizar este caso en un grado algo mayor al tener en cuenta que ángulos bajos de la normal del vector y la dirección de la recta también indicarán que la intersección se producirá a una distancia muy lejana, por lo que a efectos prácticos no se producirá –habrá otra pared más cerca–. Así, en el caso de que $\mathbf{d} \cdot \hat{\mathbf{n}} < 10^{-4}$ se interpretará que no hay un punto de intersección con la pared a evaluar.

La última de las consideraciones tiene que ver con los errores de redondeo. A la hora de calcular t_i o las coordenadas del punto de impacto es posible que el punto de intersección obtenido no se encuentre en el plano de incidencia, lo que provoca futuros errores con los rayos reflejados y transmitidos. Para evitarlo, solo se considerará la intersección con los planos si $t_i > 10^{-3}$.

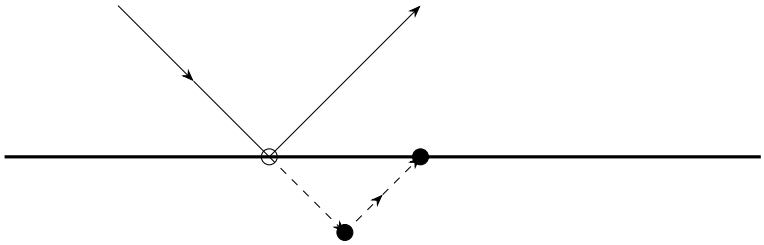


Figura 2.2: Puntos de intersección en presencia de errores de redondeo: al estar fuera del plano, la evaluación del rayo reflejado encuentra otro punto de intersección en la misma pared. Los círculos huecos y líneas sólidas indican los puntos y rayos correctos; los círculos rellenos y líneas punteadas representan los puntos y rayos calculados erróneamente.

Esta condición puede parecer confusa pero su razón se puede ver en la Figura 2.2. Debido a errores de redondeo, los rayos reflejados y transmitidos no tienen su origen en el plano de incidencia, por lo que al evaluarlos, encontraremos que la pared con la que impactaría sería, de nuevo, el mismo plano.

Con la condición introducida, no habrá intersecciones estos planos tan cercanos, por lo que el rayo será libre de obviarlo y buscar un rebote en alguna otra pared, como debería haber hecho sin los errores de redondeo.

Aunque en este ejemplo solo se han representado los rayos reflejados, en el caso de que el punto de intersección se encuentre –de nuevo, siguiendo la simetría del ejemplo– delante del plano tendríamos la misma situación al evaluar el rayo transmitido.

Estos nuevos puntos no solo son erróneos –no se deberían producir–, sino que además pueden llegar a producir valores totalmente distorsionados de la potencia de la señal como se explicará en secciones sucesivas.

Una vez determinado el punto de impacto, es necesaria una última comprobación. Al igual que al hablar de las rectas se ponía su manifiesto su extensión infinita, es posible tener la misma consideración con los planos, es decir, será posible encontrar puntos de intersección en cualquier punto del espacio.

Para imponer la presencia de los puntos de intersección entre los límites de la pared es necesario recurrir a sus esquinas –los puntos sobre los que se definen–. Tras determinar el punto de intersección, se comprueba, para cada dimensión, que su coordenada se encuentra entre los valores máximos y mínimos de las coordenadas de la pared en dicha dimensión.

Será necesario añadir un cierto épsilon para, de nuevo, evitar los errores de redondeo, y así evitar que el plano no «sea invisible» al rayo. Por tanto, la comprobación a realizar será, que para cada dimensión i, las coordenadas estén contenidas en $[\text{mín}_i(c_{ij}) - \varepsilon, \text{máx}_i(c_{ij}) + \varepsilon]$ siendo c_{ij} la coordenada i de la esquina j.

Geometría tras los impactos

Una vez obtenido el punto de impacto del rayo en la pared, queda evaluar los rayos obtenidos a partir de él.

En cada uno de estos impactos se generá un rayo reflejado y un rayo transmitido. El rayo reflejado seguirá una reflexión especular, de modo que su dirección seguirá

$$\mathbf{d}_s = 2(\hat{\mathbf{n}} \cdot \mathbf{d}_i)\hat{\mathbf{n}} - \mathbf{d}_i \tag{2.6}$$

donde $\mathbf{d}_i$ indica la dirección del rayo incidente y $\mathbf{d}_s$ la del rayo reflejado, ambos vectores normalizados.

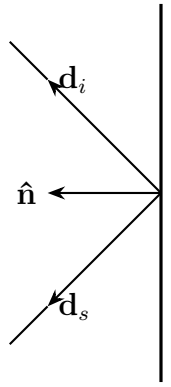


Figura 2.3: Reflexión especular.

La Ec. (2.6) supone la disposición de las direcciones tal y como se indica en la Figura 2.3, de tal forma que ambas se indican partiendo desde la pared. Desde la perspectiva del rayo incidente, esta dirección será la inversa a su dirección de incidencia.

Una vez determinado el rayo reflejado queda evaluar el rayo transmitido. Para ello se recurre a la ley de Snell para la refracción, pero teniendo en cuenta que estamos considerando las paredes como planos bidimiensionales, los rayos transmitidos compartirán la dirección del rayo impactado.

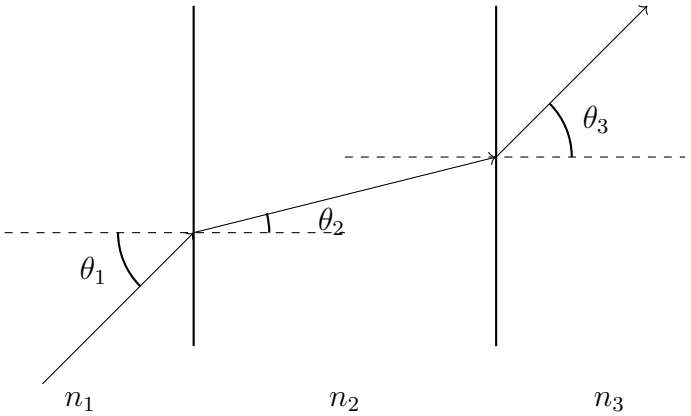


Figura 2.4: Rayos refractados al atravesar un material considerando un cambio de medio.

Al igual que la dirección también tendrán como origen el punto de impacto, debido a la bidimensionalidad del plano. La Figura 2.4 indica el caso de una pared con un cierto grosor.

Los ángulos de entrada y salida serán los mismos sin más que comprobar la ley de Snell.[3] Para el primer borde

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2) \tag{2.7}$$

y para el segundo, si el último medio fuese distinto

$$n_2 \sin(\theta_2) = n_3 \sin(\theta_3) \tag{2.8}$$

Teniendo en cuenta que $n_1 = n_3$, está claro que $\sin(\theta_1) = \sin(\theta_3)$, por lo que ambos ángulos tomarán el mismo valor y sus direcciones serán las mismas.

En cuando al punto de origen del nuevo rayo, ya en la Figura 2.4 se puede observar que vendrá determinado por el ángulo θ_2 y el tamaño de la pared. En el caso en el que este tamaño sea cero, el desplazamiento también lo será, por lo que el origen del nuevo rayo será el punto de impacto.

Determinación del ángulo de impacto

Como se explica en siguientes secciones, será necesario evaluar el ángulo de impacto del rayo en la pared para obtener la potencia de los rayos generados.

Este valor se obtiene sin problema teniendo en cuenta que su coseno será el producto escalar de la normal y el vector de la dirección incidente –más concretamente de su inverso, como se puede observar de nuevo en la Figura 2.3–, pero es necesario hacer una puntualización.

El cálculo de la normal del plano se define a partir de las tres primeras esquinas introducidas para cada pared. Por ello, dependiendo del orden en el que se escriban será posible que la normal tome un sentido o su opuesto.

En las ecuaciones (2.5) y (2.6) ese cambio de signo es irrelevante, pero no en este caso. Si la normal está definida en la dirección opuesta a la de llegada del rayo, este ángulo será el complementario, como se puede observar en la Figura 2.5.

(a) Normal en la dirección de llegada del rayo.

(b) Normal en la dirección opuesta a la llegada del rayo.

Figura 2.5: Dependiendo de la definición de la normal, el ángulo de incidencia puede variar.

Para corregir este comportamiento bastará con tomar el valor absoluto del producto escalar mencionado anteriormente, de modo que su arco coseno esté limitado entre 0 y $\pi/2$ y se obtenga el ángulo correcto.

2.2.3. Receptores

Una vez definidos los rayos y el entorno solo falta la modelización de los receptores de señal.

La interpretación anterior de rayos y paredes hace que solo se tengan en cuenta los puntos de partida e impacto de las rectas, pero no su camino entre ellos. Es este camino el objetivo del problema, pues son los puntos en los que la señal de WiFi es útil.

Para solventarlo se colocan a lo largo del entorno generado una serie de esferas que harán las veces de antenas receptoras. Estas esferas registrarán la potencia de los rayos que impacten contra ellas, de modo que sea posible obtener en los puntos en los que se han colocado la potencia total de señal que una antena colocada en el mismo lugar.

Así, se necesario buscar la intersección de las rectas de las que se compone cada rayo con cada una de estas esferas. Teniendo en cuenta que la ecuación de una esfera de radio r centrada en C es

$$\|X - C\|^2 = r^2 \tag{2.9}$$

y recuperando la Ec. (2.3) es posible combinarlas de modo que tengan una intersección para algún t_i

$$\|O + t_i\mathbf{d} - C\|^2 = r^2 \tag{2.10}$$

Tras manipularla, llegamos a poder obtener t_i a partir de una ecuación cuadrática de modo que

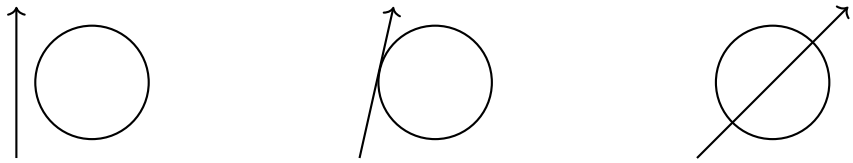
$$t_i = \frac{-2\hat{\mathbf{n}} \cdot (O - C) \pm \sqrt{(2\hat{\mathbf{n}} \cdot (O - C))^2 - 4\|\hat{\mathbf{n}}\|^2(\|O - C\|^2 - r^2)}}{2\|\hat{\mathbf{n}}\|^2} \equiv \hat{\mathbf{n}} \cdot (O - C) \pm \sqrt{\Delta} \quad (2.11)$$

con

$$\Delta = (2\hat{\mathbf{n}} \cdot (O - C))^2 - 4\|\hat{\mathbf{n}}\|^2(\|O - C\|^2 - r^2) \quad (2.12)$$

El discriminante Δ de esta solución indicará las características del punto de intersección.

Si toma un valor negativo la recta y la esfera no se encontrarán; si es nulo, ambas soluciones serán idénticas, es decir, la recta toca a la esfera de forma tangencial. Por último, valores mayores que cero indicarán que el rayo penetra en la esfera, con lo que hay dos puntos de intersección: uno de entrada y otro de salida. Estos comportamientos se reflejan en la Figura 2.6 con esquemas para cada caso.



(a) $\Delta < 0$ (b) $\Delta = 0$ (c) $\Delta > 0$

Figura 2.6: Esquema de las tres posibilidades de determinante y su significado geométrico.

Es únicamente este último caso el que se busca en la simulación, ya que un contacto tangencial no tiene relevancia física para el problema: el volumen de las esferas no corresponde de forma correcta con el de las antenas, por lo que en esos casos los rayos no impactarían de forma directa con la propia antena en un entorno real. Por ello, solo se considerará la existencia de un impacto en el caso de que el discriminante de la Ec. (2.11) sea estrictamente mayor que cero.

Es posible, por las mismas razones que se explicaban en el caso de los planos, que las soluciones tomen valores negativos. De nuevo, no se considerarán estos casos como impactos de los rayos al no encontrarse en su dirección de propagación.

Se obtienen dos soluciones para t_i. Si ambas son positivas, está claro que la solución en el caso de la suma en la Ec. (2.11) será siempre mayor que la de la resta, por lo que es en este último caso en el que la distancia de impacto es menor.

Se tomará ese valor –el de la resta– como solución final para determinar el punto de intersección de rayo y esfera, recurriendo de nuevo a la Ec. (2.3) para obtener sus coordenadas.

Determinación del ángulo de impacto

Para simular de forma correcta el comportamiento de las antenas será necesario determinar el ángulo de impacto del rayo en los receptores, de la misma forma que se hacía en las paredes –aunque ahora no se busquen rayos reflejados–.

Para ello este caso es más sencillo. Bastará con pasar a coordenadas esféricas la dirección del ángulo incidente, a las que se podrían añadir ciertos *offsets* en cada uno de los ejes si los receptores se colocan con una orientación distinta a la de los ejes cartesianos.

2.3. Electromagnetismo en los impactos

Con las bases de la geometría del problema establecidas, resta abordar la física del problema: la interacción electromagnética de los rayos con el medio.

Como punto de partida es necesario abordar el comportamiento de estos rayos en su camino anterior a cualquier impacto contra alguno de los obstáculos, para lo que es habitual el uso de la ecuación de transmisión de Friis, definida como[4]

$$P_t = P_i \left(\frac{\lambda}{4\pi r} \right)^2 \tag{2.13}$$

donde P_t indica la potencia de una señal de longitud de onda λ a una distancia r, siendo emitida en su origen por una potencia P_i.

Esta ecuación es una aproximación que no siempre es aplicable. Su uso se limita a la modelización de la señal emitida y captada por antenas, en las que se omite cualquier efecto de reflexión entre ellas.

Para poder hacer esta consideración, se establece el límite

$$r > \frac{2D^2}{\lambda} \tag{2.14}$$

como rango válidos, con D siendo el tamaño característico de las antenas –en el caso de que sean distintos, se considera el mayor de ellos–.

Teniendo en cuenta que en este caso se usarán ondas de WiFi de 2,4GHz, su longitud de onda será $\lambda \approx 0{,}12$m. Las antenas a usar tienen un tamaño de entre 10 y 20 cm, así que la Ec. (2.13) será valida a distancias superiores a unos 70cm.

Comportamiento tras los impactos

Tras su recorrido libre en el aire, a su llegada al punto de impacto contra alguna de las paredes u obstáculos el rayo seguirá las leyes de Fresnel para la frontera entre dos medios.

En este caso son solo de interés las expresiones de los coeficientes de reflexión y transmisión de la potencia, que describen la proporción de energía de los rayos reflejados y transmitidos respectivamente.

Estas ecuaciones tendrán expresiones distintas dependiendo de si tratamos ondas s-polarizadas –con el campo eléctrico perpendicular al plano de incidencia– o p-polarizadas –con el campo eléctrico paralelo al plano de incidencia–.[5]

$$\begin{aligned} R_{\mathrm{s}} &= \left| \frac{Z_2 \cos(\theta_i) - Z_1 \cos(\theta_t)}{Z_2 \cos(\theta_i) + Z_1 \cos(\theta_t)} \right|^2 \\ R_{\mathrm{p}} &= \left| \frac{Z_2 \cos(\theta_t) - Z_1 \cos(\theta_i)}{Z_2 \cos(\theta_t) + Z_1 \cos(\theta_i)} \right|^2 \end{aligned} \tag{2.15}$$

donde Z_i hace referencia a la impedancia de la onda en cada medio, dada por

$$Z = \sqrt{\frac{j\omega\mu}{\sigma + j\omega\varepsilon}} \tag{2.16}$$

con μ indicando la permeabilidad magnética del medio, ε la permivitidad dieléctrica y ω la frecuencia angular de la onda en cuestión.

Asumiendo que ambos medios son no magnéticos –es decir, $\mu = \mu_0$, como va a ser el caso en las simulaciones a realizar– las expresiones de la Ec. (2.17) pasan a depender de los índices de refracción de modo que

$$\begin{aligned} R_{\mathrm{s}} &= \left| \frac{n_1 \cos(\theta_i) - n_2 \cos(\theta_t)}{n_1 \cos(\theta_i) + n_2 \cos(\theta_t)} \right|^2 \\ R_{\mathrm{p}} &= \left| \frac{n_1 \cos(\theta_t) - n_2 \cos(\theta_i)}{n_1 \cos(\theta_t) + n_2 \cos(\theta_i)} \right|^2 \end{aligned} \tag{2.17}$$

Es posible eliminar el término dependiente del ángulo transmitido usando la ley de Snell

de la refracción, de modo que las Ecs.(2.17) pasan a ser

$$\begin{aligned} R_{\mathrm{s}} &= \left| \frac{n_1 \cos(\theta_i) - n_2 \sqrt{1 - \left[\frac{n_1}{n_2} \sin(\theta_i)\right]^2}}{n_1 \cos(\theta_i) + n_2 \sqrt{1 - \left[\frac{n_1}{n_2} \sin(\theta_i)\right]^2}} \right|^2 \\ R_{\mathrm{p}} &= \left| \frac{n_1 \sqrt{1 - \left[\frac{n_1}{n_2} \sin(\theta_i)\right]^2} - n_2 \cos(\theta_i)}{n_1 \sqrt{1 - \left[\frac{n_1}{n_2} \sin(\theta_i)\right]^2} + n_2 \cos(\theta_i)} \right|^2 \end{aligned} \tag{2.18}$$

Asumiendo que el medio 1 es el aire, con índice de refracción igual a 1, se obtiene la expresión final usada en la simulación

$$\begin{aligned} R_{\mathrm{s}} &= \left| \frac{\cos(\theta_i) - \sqrt{\varepsilon_r - \sin^2(\theta_i)}}{\cos(\theta_i) + \sqrt{\varepsilon_r - \sin^2(\theta_i)}} \right|^2 \\ R_{\mathrm{p}} &= \left| \frac{\varepsilon_r \cos(\theta_i) - \sqrt{\varepsilon_r - \sin^2(\theta_i)}}{\varepsilon_r \cos(\theta_i) + \sqrt{\varepsilon_r - \sin^2(\theta_i)}} \right|^2 \end{aligned} \tag{2.19}$$

donde se ha utilizado la relación $n = \sqrt{\varepsilon}$, dejando las expresiones dependiendo solo de la permitividad dieléctrica del medio y el ángulo de incidencia.

Para cada uno de los dos casos, la potencia transmitida será

$$T_i = 1 - R_i \tag{2.20}$$

por consevación de la energía.

Queda por aclarar la polarización de la ondas emitidas. La aproximación tomada es asumir ondas no polarizadas, o lo que es lo mismo, con polarización circular, de modo que las energías de los campos eléctrico y magnético son las mismas y es posible tomar un coeficiente de reflexión de

$$R = \frac{1}{2} \left[R_{\mathrm{s}} + R_{\mathrm{p}} \right] \tag{2.21}$$

Difracción y *scattering*

Además de los rayos recibidos de forma directa y por el reflejo en los objetos existen otros dos fenómenos de generación de radiación en los impactos.

En primero de ellos es la difracción. En el contacto de los rayos con los objetos se crearán conos de emisión de otros rayos, pero su modelización es muy complicada.

Los coeficientes de difracción no tienen una expresión tan sencilla y existen métodos desarrollados desde hace décadas para su modelización como la teoría geométrica de la difracción –GTD por su nombre en inglés– y posteriormente la teoría uniforme de la difracción –UTD–.[1]

En ambos casos las potencias de los rayos difractados será muy baja. En el caso de que los rayos originales tengan una potencia muy alta estos productos de la difracción tendrán un efecto menor. En la propagación de interiores, donde las potencias son muy bajas, su efecto será residual.[5]

En esta simulación no se considerará este efecto. El alto coste computacional y la dificultad en la modelización no compensa el escaso efecto que tendrá comparado con el resto de rayos. En general, esta ausencia está también presente en otros modelos de trazado de rayos.

Por otro lado también existe el fenómeno de *scattering*, fruto de ciertas irregularidades y bordes en las superficies de los obstáculos y paredes.[1]

Los rayos producidos por este fenómeno tienen, como en el caso anterior, muy poca potencia. Su parametrización es importante en entornos urbanos por la alta cantidad de edificios, pero en el caso de interiores no es tan relevante.

Por ello, por los mismos argumentos utilizados para los rayos difractados se aplican en este caso: el coste computacional no compensa el escaso efecto que producirían en los resultados, por lo que no se considerarán en la simulación.

2.4. Antenas

Una vez establecidas las bases para la evaluación de los rayos en el área de interés es necesario abordar su origen y su recepción.

La emisión y recepción de estos rayos se hará mediante antenas que tendrán cierta direccionalidad fruto de su simetría que será necesario considerar. Aquí aparece el concepto de direccionalidad, donde se parametriza la intensidad de la emisión o recepción dependiendo del ángulo de emisión o incidencia del ángulo.[4]

Esta directividad está definida como

$$D = \frac{4\pi U}{P_{\text{rad}}} \tag{2.22}$$

donde U indica la potencia por unidad de ángulo sólido y P_{rad} la potencia total emitida.

En el caso de la antena receptora se obtiene la misma expresión, teniendo en el denominador la potencia recibida. Este valor de recepción es a veces llamado «ganancia», aunque su origen y concepto es el mismo que la direccionalidad.

En el caso de que una antena emita isotrópicamente la direccionalidad será la misma para todos los ángulos, de modo que es posible definir la direccionalidad como la razón entre la potencia de emisión entre la potencia de emisión si tuviese una emisión isotrópica.

La obtención de la función de forma general se obtiene de forma numérica. En el caso de antenas con un diseño con alguna simetría existen expresiones analíticas, pero no podrán ser aplicadas en entornos reales como el que se trata en este trabajo.

Así, se recurrió a las utilidades proporcionadas en el *Antenna Toolbox* de MATLAB, que permite definir antenas con un diseño arbitrario compuesta de una amplia serie de materiales y con ellos obtener su direccionalidad.[6]

Las antenas utilizadas son antenas cilíndricas, como es habitual en el caso de comunicaciones WiFi. En la Figura 2.7 se incluyen dos imágnes de las antenas utilizadas.

(a) Imágenes de la antena del router utilizado. (b) Imágenes de la antena receptora utilizada

Figura 2.7: Imágenes de las antenas utilizadas.

En este caso se eligió una antena cilíndrica con 1mm de radio y 10cm de altura, con la alimentación justo a la mitad para componer una antena dipolar típica de cualquier router o receptor de WiFi. Este modelo fue utilizado para las dos antenas utilizadas.

Esta antena forma un patrón con forma de toroide como se puede observar en la Figura 2.8. Así, se obtiene una direccionalidad algo mayor en ángulos cercanos al plano perpendicular al cilindro y prácticamente nula en el eje del propio cilindro.

La herramienta de MATLAB permite especificar los ángulos para los que se quiere generar valores de directividad al ser un cálculo numérico. En este caso se generó con saltos de un ángulo sexagesimal, ya que permite un buen equilibrio entre una simulación que se completa

en un tiempo razonable y puntos lo suficientemente juntos. Para esta antena el patrón es muy suave, no es necesaria una precisión mayor.

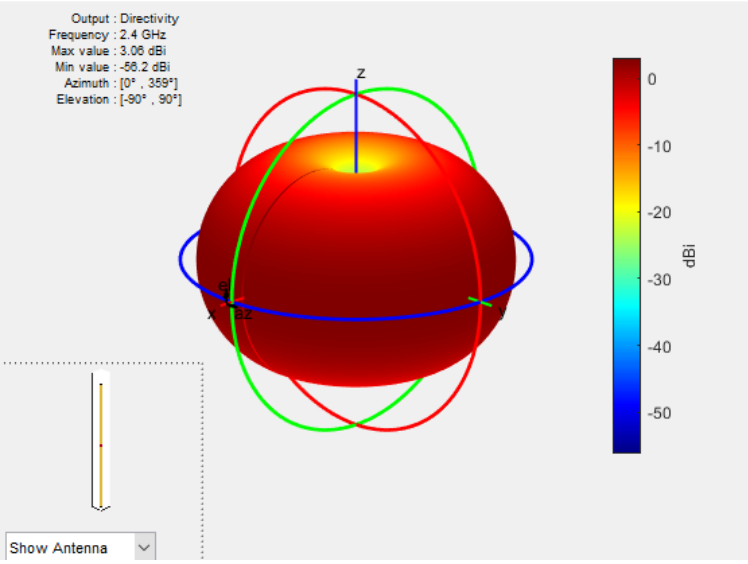


Figura 2.8: Captura de la herramienta de MATLAB que permite visualizar el patrón de emisión numérico de una antena.

Otra de las propiedades que proporciona esta herramienta es la de determinar la polarización de la antena a partir del campo magnético y eléctrico en varios puntos de la misma, de la misma forma que lo hacía para el patrón de emisión.[6]

En la Figura 2.9 se puede observar el patrón de la antena simulada. Se puede observar que en prácticamente en todas las zonas la polarización es circular –es decir, no está polarizada– obteniendo intensidades similares en ambos campos.

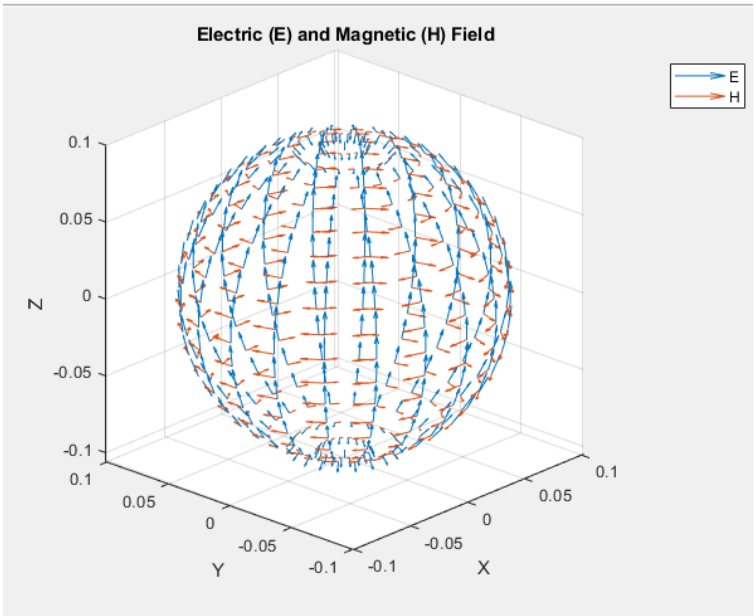


Figura 2.9: Captura de la herramienta de MATLAB que permite visualizar la polarización de una antena.

Esta polarización sí que cambia en zonas cercanas a la vertical de la antena, de la misma forma que ocurría con el patrón de emisión.

Esta característica es la que da pie a considerar rayos no polarizados en la sección anterior. La antena solo emitirá ondas no polarizadas en las zonas con un patrón menor, es decir, donde la potencia de los rayos generados es menor.

Considerando el gran decaimiento de la potencia que sufren estos rayos con la distancia, el resto de rayos polarizados, con menor potencia en el origen, no tendrán un gran efecto respecto a los no polarizados. Además, siendo generados prácticamente en la vertical de la antena, es poco probable que lleguen a los receptores, colocados mayoritamente a diferentes posiciones pero en el mismo plano.

3. Simulación en serie y en paralelo

A la hora de lanzar todos los rayos de la simulación y su evaluación se presentan dos vías: la ejecución en serie en una CPU usual y la ejecución en paralelo en una GPU.

Aunque ambas opciones compartan gran parte de los cálculos, las diferentes estructuras entre los dispositivos hacen que se presenten diferencias a la hora de implementar la simulación.

3.1. Elementos comunes

Partiendo de las bases descritas en la sección anterior la simulación consistirá, de forma general, en lanzar rayos y acumular los resultados de los receptores.

Aprovechando las ventajas de la programación orientada a objetos, se considerarán los siguientes clases:

- **Rayo** Esta clase contendrá el punto de partida y el vector de dirección del rayo, así como la potencia en el origen.
- **Pared** En esta clase se incluirán los cuatro puntos de los extremos de la pared, a partir de los que se calculará su normal, también almacenada. Además, contendrá el valor de permeabilidad dieléctrica de su material. Implementará métodos para determinar si un rayo la impacta, y obtener tanto el punto de impacto como el rayo reflejado.
- **Receptor** Esta clase incluye el origen de la esfera y su radio. Al igual que la clase de las paredes implementará métodos para determinar si un rayo la impacta, y obtener la potencia teniendo en cuenta el patrón de emisión de la antena que modeliza.

La simulación comienza con la obtención del entorno virtual. A partir de medidas en el entorno real, se escriben en un archivo las esquinas de todas las paredes de entorno, así como los límites de los objetos.

Con estas esquinas se construye un vector con los objetos de paredes, creados a partir de estas esquinas. Este vector constiuirá el mapa de la simulación.

Una vez se ha creado el mapa se crean los receptores, con la posición y su distancia. Aunque la implementación permite cualquier radio, en este caso se limitó a 2,5cm.

El siguiente paso será leer el patrón de radiación de las antenas. Ya que para esta simulación el de el emisor y el receptor es el mismo, solo se hará en una ocasión, y se utilizarán también estos datos para la generación de rayos.

Este patrón, como se explicaba en la Sección 2.4, está generado directamente por MATLAB, y su formato es el de una matriz en el que cada uno de los ejes indica un ángulo. Su lectura y uso sigue el mismo formato.

Una vez cargados todos los datos, se inicia el bucle principal, deberá recorrer el ángulo azimutal y el ángulo de elevación avanzando con un cierto valor prefijado en ambos. En las subsecciones posteriores se profundizará en la evaluación de cada rayo.

Una vez se han lanzado todos los rayos se acumulan los resultados –de nuevo, en las siguientes subsecciones se discutirá en detalle la estructura de memoria para estos resultados– y se escriben en un archivo para su análisis posterior.

El Algoritmo 1 recoge en pseudocódigo todo el proceso.

Algoritmo 1 Bucle Principal

```
Leer parámetros.
Cargar mapa.
Colocar receptores.
for azimut ∈ [0, 360) do   ▷ Se evaluan los ángulos sumando un cierto paso fijado en los parámetros.
    for elev ∈ [90, −90] do
        eval_ray(azimut, elev, pot, data)
    end for
end for
                                ▷ Data indica la matriz donde se acumulan los resultados.
for i ← 0, rows(data) do
    for all j ← 1, cols(data) do
        data[i][0] ← data[i][0] + data[i][j]
    end for
end for
write_results(file, data)
```

Evolución de cada rayo

El lanzamiento y evolución de cada rayo tiene varias fases a discutir.

En primer lugar, una vez se ha determinado la dirección de lanzamiento se debe calcular

la potencia a emitir. Habiendo importado la direccionalidad de la antena de emisión, se realiza una interpolación bidimiensional[2] para estimar el valor correspodiente.

Una vez se ha lanzado, cada rayo buscará en primer lugar la pared contra la que impactará. Esta pared será la que proporcione una distancia de impacto menor.

Tras ello, se buscarán los receptores contra los que impactará el rayo. En este caso se recorrerán todos los receptores colocados, y solo se considerarán como impactados aquellos para los que la distancia de impacto sea menor que la distancia de impacto contra la pared.

Esta consideración se produce para evitar el impacto contra receptores que se encuentren detrás de una pared u objeto.

Una vez se han recorrido ambos elementos, se habrá completado la interacción del rayo con el medio. Se determinará su punto de impacto: si lo hay, será el punto donde se generarán el rayo transmitido y el rayo reflejado. Se evaluará solo el rayo transmitido, por lo que el reflejado tendrá que ser almacenado.

Este almacenamiento se hará en forma de pila, de tal forma que los rayos añadidos más recientemente serán los primeros en ser evaluados. Esta elección es totalmente arbitraria: todos los rayos son independientes entre sí.

Para optimizar el uso de memoria de la pila se reservará toda la memoria que se podría usar antes de iniciar el bucle y así no necesitar movimientos posteriores. El tamaño necesario vendrá determinado por el número de rebotes que se quieran registrar.

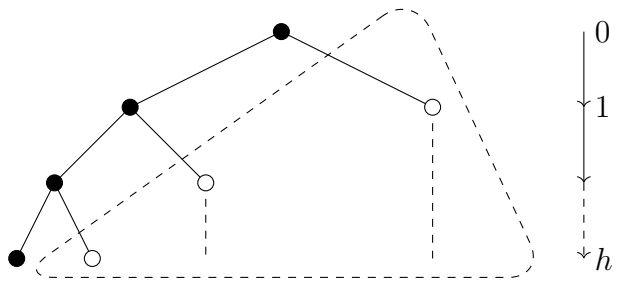


Figura 3.1: Árbol generado por los impactos de los rayos hasta llegar a h impactos. Los puntos negros indican los rayos reflejados, evaluados inmediatamente; los blancos, los transmitidos. En la línea punteada, el número máximo de rayos a almacenar.

En la Figura 3.1 se indica el árbol generado por un rayo, donde en cada impacto nacen dos rayos. Considerando que siempre se evalúa el rayo reflejado –en esta representación, el

[2]En este caso esta interpolación ha sido bilineal. Aunque es posible obtener unos valores más precisos con métodos de mayor orden, partiendo de puntos separados solo un grado en ambos ángulos esta elección proporciona valores aceptables sin tener que abordar las condiciones de contorno en los extremos.

camino de la izquierda– podría parecer que se deberá reservar memoria para todos los rayos restantes, pero es posible un uso menor.

Los rayos guardados en la pila no son evaluados de forma inmediata, así que los rayos que se generan no se conocerán hasta que se llegue a su nodo. Es decir, el número de rayos en la pila no llegará nunca a ser tan alto, aunque es necesario dar un número de elementos a reservar.

Para ello se asumirá que es imprescindible guardar todos los rayos recogidos en la línea punteada. Estos serán todos los elementos del árbol ($\sum_h 2^h$) menos el número de niveles que tenga, que será $h+1$ teniendo en cuenta que tal y como se indica, se ha empezado a contar desde el cero y así

$$\#\text{ Rayos a guardar} = \sum_{h=0}^{\infty} 2^h - (h+1) \tag{3.1}$$

El Algoritmo 2 contiene, en pseudocódigo, la subroutina que evalúa cada rayo tal y como se ha descrito en la sección.

Registro en los receptores

Una vez los rayos se propagan de forma libre por el entorno se esparará a su impacto con cada uno de los receptores para registrar su efecto.

Hay que tener en cuenta que se busca sumar la potencia recibida en cada receptor por parte de todos los rayos que los cruzen, pero con un detalle: no se pueden registrar todos los rayos sin más.

Si se suma la potencia de cualquiera de los rayos que impactan contra un receptor se obtendrán potencias totales totalmente irreales, ya que, especialmente en el caso de receptores colocados a distancias cercanas al emisor, habrá una gran cantidad de rayos que, indicando un mismo frente de ondas, lleguen al impacto contra ellos.

Este comportamiento es completamente irreal, fruto de la aproximación de la antena como esferas. El frente de ondas, representado por un solo rayo, solo debería cruzarla una vez y no un número mayor como ocurriría en este caso.

La misma situación se presenta en los rebotes, con lo que se podría acabar con un exceso de potencia tanto en los rayos que llegan de forma directa como en los que llegan tras la interacción con las paredes u objetos.

Para evitar esto solo se registrará el rayo con mayor potencia. Debido al gran decaimiento

Algoritmo 2 Bucle que evalúa cada rayo

```
while Tamaño pila >0 do
    reb ← segundo valor de la pareja {rayo, rebote}.
    for all Paredes do
        hit_dist ← INFINITY
        wall_hit ← -1
        if Pared golpeada then
            if dist <hit_dist then
                hit_dist ← dist
                wall_hit ← pared
            end if
        end if
    end for
    for all Receptores do
        hit_dist ← INFINITY
        power ← 0
        if Receptor golpeado then
            if hit_power >power AND dist <hit_dist then
                hit_power ← power
            end if
        end if
    end for
    if power >CUTOFF_POWER AND reb <MAX_REBOUND then
        {rayo, rebote} ← { rayo_reflejado, reb+1}
        Añadir { rayo_transmitido, reb+1} a la pila.
    else
        {rayo, rebote} ← último valor de la pila.
    end if
end while
```

de la potencia con la distancia, esto indicará el rayo que impacte de forma más directa contra la esfera. En otro caso, la distancia adicional hará que su potencia sea menor.

Además, se tendrá la consideración de los rebotes sufridos por el rayo. Si solo se toma el criterio anterior, solo se registrarían los rayos que impactan de forma directa.

Para solventar este problema se asociará cada rayo al número de rebotes que ha sufrido. Así, una vez impacten con los receptores se considerará este valor, y cada vez que reboten contra algún objeto se actualizará.

Los resultados se almacenarán considerando una matriz con tantas filas como receptores y tantas columnas como número máximo de rebotes a registrar. Cuando el cálculo finalice, se acumularán los resultados sumando los valores por filas en la primera columna obteniendo el resultado final de cada receptor.

3.2. Implementación en serie

La implementación en serie de la simulación para ser ejecutada en una CPU sigue el procedimiento explicado en esta sección.

Existirá una zona de memoria reservada para registrar los datos y simplemente se recurre a un doble bucle *for* que irá completando los resultados evaluando cada par de ángulos avanzando con un cierto paso predefinido.

Ya que la evaluación de cada rayto se realizará de forma totalmente secuencial, no será necesario añadir ninguna consideración adicional.

3.3. Implementación en paralelo

El hecho de que los rayos lanzados sean independientes entre sí convierte en el bucle principal en el escenario perfecto para ser ejecutado de forma paralela, ya que cada iteración del bucle no se verá interrumpida por las demás.

Para esta paralelización se ha hecho uso de tarjetas gráficas, que cuentan con un gran número de núcleos de cómputo pero para las que hay que tener en cuenta su estructura de memoria.

En general, a la hora de paralelizar un algoritmo es necesario hacer duplicaciones de datos para evitar condiciones de carrera que proporcionen resultados erróneos. En los casos de muy alta paralelización como este es posible encontrarse con ciertas limitaciones.

La tarjeta gráfica usada en este caso ha sido una Nvidia GTX1070, sobre la que es posible el uso de la librería de Nvidia CUDA. Esta tarjeta cuenta con 1920 núcleos agrupados en 15 multiprocesadores.

La estrategia de paralelización consistirá en asignar un rayo a cada hilo, que contará con su matriz de datos correspondiente donde registrará los impactos para luego ser acumulada en los datos finales.

Las tarjetas gráficas agrupan sus hilos en bloques. Los hilos que se encuentren en un mismo bloque pueden acceder a una cierta cantidad de memoria compartida de un acceso más rápido que la memoria global común.

Para reducir la necesidad de memoria y aprovechar esta característica la acumulación de los datos se producirá en dos fases: el cálculo de cada rayo –asignado a cada hilo– volcará sus datos en la memoria compartida del bloque. Una vez acaban todo los hilos del bloque, se acumulan los datos en la memoria global, donde se ha reservado espacio para cada bloque.

Debido a la herencia del uso de este tipo de dispositivos para trabajos de vídeo, es posible acceder a los hilos y bloques con dos índices, de forma similar a una matriz. Esta característica será útil en este caso, en el que los rayos se lanzarán en torno a los dos ángulos de las coordenadas esféricas.

Así, se asociará la dirección x al ángulo azimutal y la dirección y a la elevación.

Queda determinar el número de hilos en cada bloque. En general, la memoria disponible para su uso compartido es una cantidad baja. En el caso de la tarjeta gráfica usada, es de 48 KB[7].

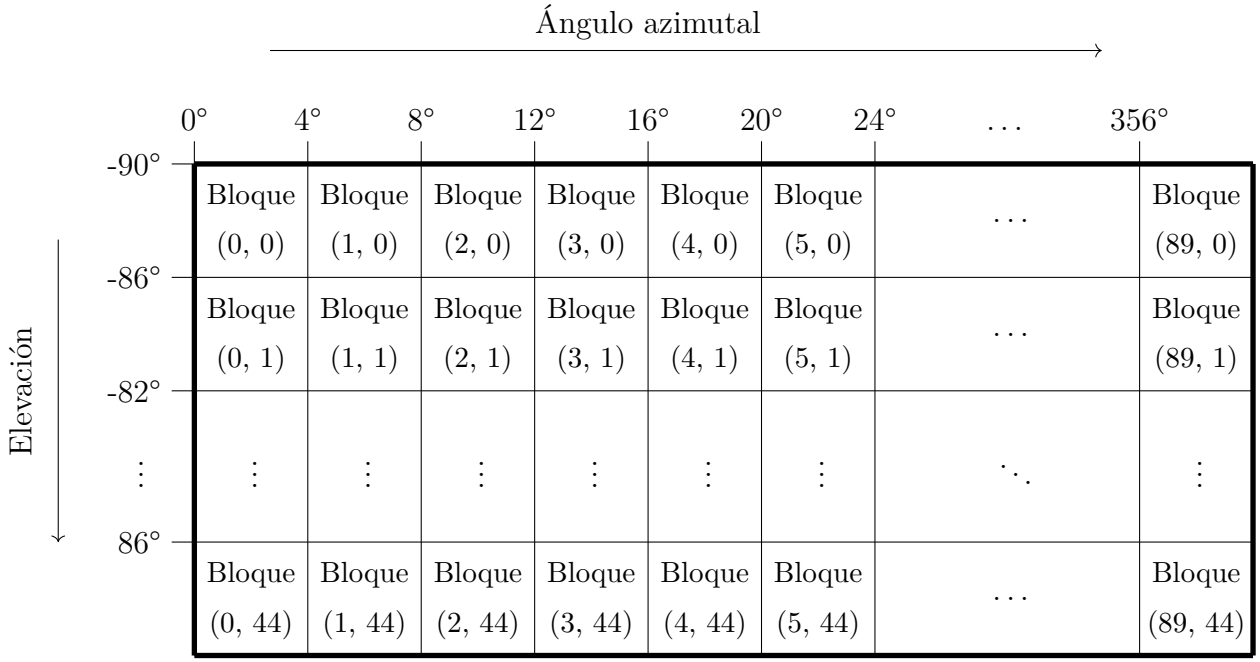


Figura 3.2: Esquema con la asignación de ángulos a cada bloque de hilos.

Esta cantidad no permite lanzar bloques con un número grande de hilos, aunque dependerá del número de receptores y los rebotes que se quieran registrar. A modo de ejemplo, al colocar unos 40 receptores y registrar los rayos hasta 5 rebotes, con valores de precisión simple –es decir, 4 bytes–, podremos lanzar bloques con 60 hilos como máximo.

Para evitar futuras incompatibilidades en caso de elegir valores mayores en alguno de estos dos parámetros, se han elegido el mínimo valor de hilos posible. La elección es la siguiente: solo se lanzarán 16 hilos por bloque, 4 en cada dirección.

Además de ser un valor bajo compatible con los límites establecidos, el número de ángulos a evaluar en ambas direcciones es divisible por 4, hecho aprovechable para la implementación.

La elección de asignación de valores angulares a los bloques será de forma fija: cada bloque solo evaluará 4 grados en cada dirección, independientemente del número de rayos que tenga asignado ejecutar.

Si los saltos en el bucle del rayo es de un grado, cada hilo solo ejecutará el rayo con los ángulos que le corresponde sin más. En el caso –habitual– que la distancia sea menor, cada hilo deberá evaluar más de un rayo.

Para facilitar el recorrido y garantizar una ejecución lo más homogénea posible[3], solo se

[3]Hay que tener en cuenta que la ejecución de los programas en tarjetas gráficas se realizan ejecutando una misma instrucción en varios hilos a la vez. Evitar expresiones condicionales que creen distintas ramas en el desarrollo del programa maximizará el rendimiento del dispositivo.

considerarán saltos en los valores angulares como potencias negativas de dos –es decir, solo se usarán saltos de valor 0,5, 0,25, 0,125,...– a fin de poder utilizar la siguiente estrategia.

Se considerará una matriz de «minibloques» dentro del bloque, todas ellas con los 16 hilos del bloque. La cantidad de estos minibloques dependerá del valor del salto entre ángulos: si es $0{,}5 \equiv 1/2$ habrá 4 minibloques, dos en cada dirección; si es $0{,}25 \equiv 1/4$ habrá 16 minibloques, etc.

Dentro de cada uno de los minibloques solo habrá 16 pares de direcciones a evaluar, uno para cada hilo. Una vez todos estos hilos finalicen su bucle, se avanzará al siguiente minibloque en dirección vertical –es decir, avanzando en el eje y– hasta acabar con todos los ángulos que se deben evaluar.

4. Entorno de pruebas

Para comprobar cómo se ajusta el modelo al comportamiento real de la emisión se tomaron medidas en el Laboratorio de Robótica 0L3 del Instituto de Computación Científica Avanzada de la Universidad de Extremadura (ICCAEx), situado en los Institutos Universitarios de Investigación de la Universidad de Extremadura en Badajoz.

En este laboratorio se dispone de una superficie amplia en la que es posible usar un robot para automatizar la toma de medidas en divesos puntos, que más tarde fueron usados en la simulación colocando los receptores en dichos puntos.

Figura 4.1: Imágenes del laboratorio donde se realizaron las pruebas.

Cabe destacar que el entorno de pruebas tenía un especto electromagnético muy poblado. Al encontrarse en un edificio con otros laboratorios, se encontraban entre 10 y 15 redes WiFi, con lo que resultó imposible encontrar algún canal libre para realizar las pruebas.

Para solventar esta contaminación se tomaron, para cada punto, 5 medidas esperando 5 segundos ente cada una de ellas, de modo que fue posible, tomando medias de las mismas, retirar los posibles efectos de inteferencia provocados por el resto de redes.

4.1. Robot

Con el fin de automatizar la toma de medidas para hacerla lo más eficiente posible se ha decidido usar un robot móvil capaz de desplazarse mediante navegación autónoma en un entorno conocido.

El elegido en este caso fue el robot TurtleBot 2, un robot con fines educativos y de investigación capaz de desplazarse y orientarse con total libertad en superficies llanas, como eran los en el que se desarrolló este trabajo. En la Figura 4.2 se puede ver el robot durante una de las tomas de medidas.

Figura 4.2: Turtlebot en una de las tomas de medidas.

El TurtleBot funciona con ROS –*Robot Operating System*, en inglés–,[8] un entorno de trabajo enfocado a los sistemas robóticos encargado del soporte para todos los dispositivos hardware del robot como motores, encoders o cámaras. Cuenta con diversas librerías para distintos casos de uso, entre las que se encuentra la navegación en un entorno controlado como la que se da en el caso de este trabajo.

Su funcionamiento se basa en una arquitectura de grafos, donde se definen *nodos*, tareas en las que se realiza el procesamiento de sensores, control, actuadores o cualquier otra función. Los nodos se comunican entre ellos con mensajes llamados *topic*, y es posible su desarrollo con los lenguajes C++ y Python.

Dentro de los paquetes disponibles, el usado en el desarrollo del trabajo es el encargado de la navegación del robot, que permite desplazar al robot a cualquier punto del entorno de trabajo y proporcionar de forma constante su posición en el mapa.

Es posible acoplar al robot un sistema de visión llamado Kinect que, mediante luz codificada, permite obtener un mapa de profundidad del entorno que lo rodea. Fue desarrollado en primera instancia para el uso en videojuegos pero su uso también se ha extendido a labo-

res de investigación al permitir el posicionamiento de objetos y paredes, de tal manera que permite al robot evitar obstáculos dinámicos en el caso de que interpongan en su camino.

La funcionalidad de navegación aprovecha estos sensores realizando una fusión de sus resultados con los datos de movimiento de los motores que impulsan al robot, de tal forma que es posible corregir cualquier error en casos donde el empuje de las ruedas no se traslade directamente en un desplazamiento del robot, como puede ocurrir al rotar sobre sí mismo.

Para conseguir el posicionamiento en el mapa el paquete de navegación utiliza un *planner* global sobre el que es posible determinar las rutas a seguir por el robot para llegar a un punto dado sorteando obstáculos y paredes. Junto a él trabaja un *planner* local, en el que gracias a los sensores incorporados se evalúan de forma continua los alrededores del robot. Así, es posible conseguir de forma continua una evaluación de los posibles obstáculos que no se encuentran en el mapa del planner global, para lo que se construye un mapa de costes con el fin de abortar el movimiento en el caso de que sea imposible alcanzar el punto objetivo [8].

Con la información actualizada del *planner* local, el *planner* global es capaz de modificar las trayectorias para que el robot pueda continuar su desplazamiento por el mapa. Es posible observar un esquema del funcionamiento de estos sistemas en la Figura 4.3.

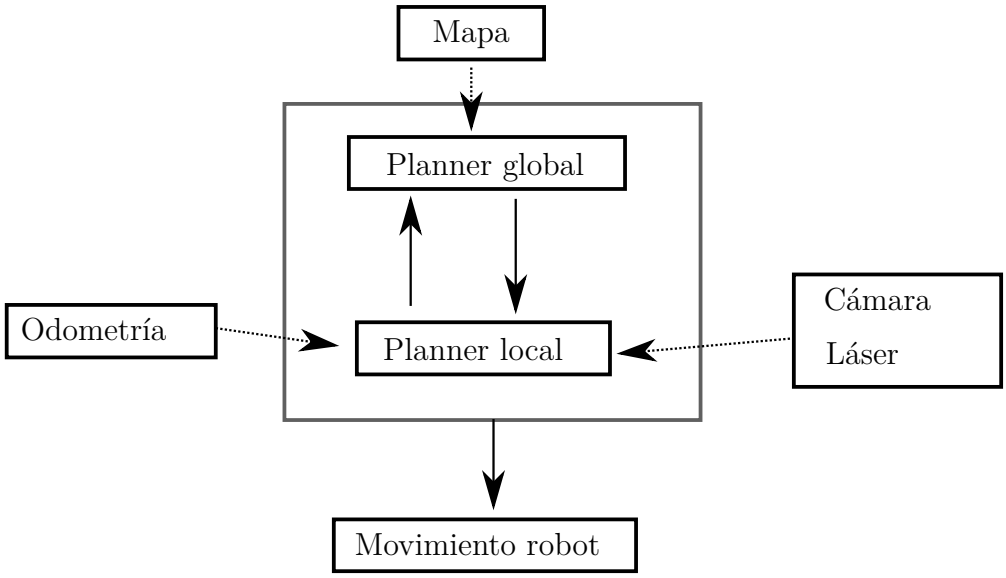


Figura 4.3: Esquema del funcionamiento del paquete de navegación de ROS.

Con esta herramienta la librería de navegación permite un mapeado autónomo tomando como referencia los datos de odometría que proporcionan los motores propulsores del robot para determinar las dimensiones del entorno.

En el caso de disponer de antemano de dichas dimensiones es posible proporcionar un mapa al sistema de navegación y evitar el paso de reconocimiento del entorno. Esto no

solo ahorra tiempo, sino que además minimiza las posibles discrepancias entre los datos de odometría y los desplazamientos reales del robot al realizar el mapeado de forma autónoma.

La opción de realizar un mapa previo fue la elegida en este caso, ya que el robot cuenta con rutinas para el reposicionamiento en el mapa en dicho caso. Así, a partir de los límites establecidos y comprobados de forma manual, las posibles discrepancias en la odometría del robot se ven continuamente compensadas y corregidas.

A partir de estos datos corregidos es posible conocer en cualquier momento la posición del robot en el mapa, expuesta a través de ROS en uno de los topic disponibles.

Para facilitar el uso de ROS existe la posibilidad de usar el simulador Stage, capaz de crear un mundo virtual a partir de un mapa en dos dimensiones en el que colocar el Turtlebot y simular su funcionamiento de forma total sin tener acceso al robot de forma física. Es posible observar su interfaz en la Figura 4.4.

Además, ROS también permite el uso de una herramienta de visualización de la posición del robot en el mapa y de todos los sensores que incorpora llamada *rviz*. Aunque es compatible con Stage, sus funcionalidades brillan al usar el robot en entornos reales, donde es posible comprobar de forma continua que su posicionamiento es correcto y que sus sensores funcionan como es debido.

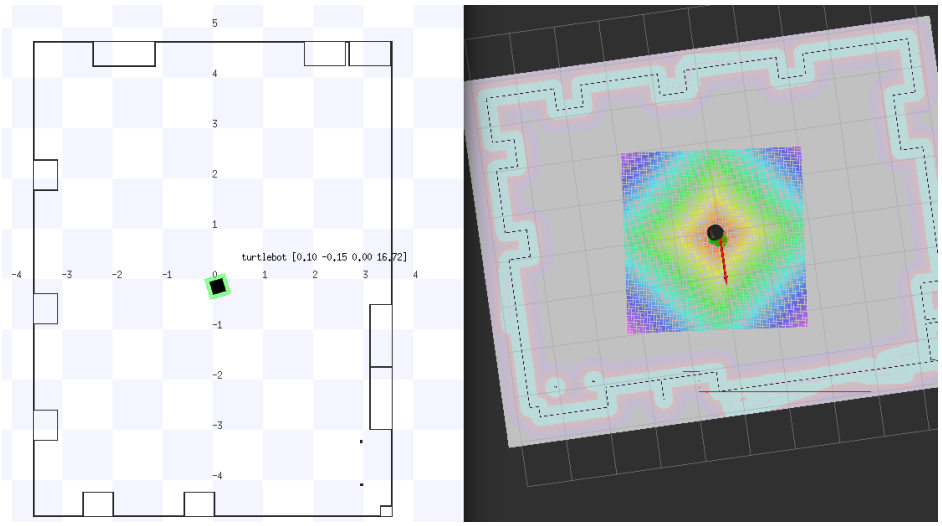


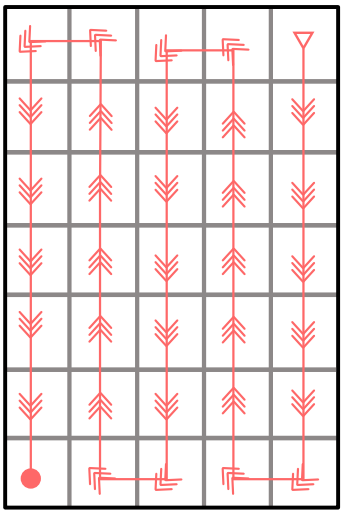
Figura 4.4: Captura del simulador Stage (a la izquierda) y la herramienta de visualización *rviz* (a la derecha).

4.1.1. Trayectoria

Dentro del laboratorio se eligió un área de 7×5m para el recorrido del robot, eligiendo los puntos en los que tomar las medidas separados 1 metro.

31	32	33	34	35
26	27	28	29	30
21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

(a) Puntos a evaluar.



(b) Trayectoria del robot.

Figura 4.5: Puntos de medida por el robot en el laboratorio y la trayectoria tomada.

La Figura 4.5 recoge la disposición de los puntos tomados y la ruta del robot para ello. En total se compone de 35 puntos: en cada uno de ellos el robot se parará y tomará las 5 medidas correspodientes.

En todos los escenarios a comentar en la sección posterior la trayectoria fue la misma. Si se coloca algún obstáculo en el camino del robot lo podrá evitar de forma totalmente autónoma, evitando tener que hacer cualquier modificación en la trayectoria establecida y programada.

4.2. Escenarios

Además del escenario del laboratorio sin ningún obstáculo, se utilizaron unas tablas de madera de dimensiones $1 \times 0{,}64 \times 0{,}02$m forradas de forma parcial con papel de aluminio, de forma que pudieran presentar un cierto efecto de apantallamiento de la radiación en su dirección.

Estas tablas, colocadas en dos configuraciones, proporcionaron otros dos escenarios con obstáculos para comprobar de manera más profunda el comportamiento del simulador.

En todos los casos el emisor estuvo colocado en la misma posición: un metro por detrás del punto 2 tal y como se indica en la Figura 4.5, a 0,55cm de altura.

Es esta altura por la que solo se forra la mitad superior de las tablas: tal y como se mostraba en la sección 2.4, el patrón de emisión coloca la mayor parte de la potencia siendo emitida en el plano de la antena. Sumando el mayor decaimiento al recorrer una distancia

mayor, las trayectorias alternativas que involucren rebotes contra el suelo o el techo tendrán un efecto mucho menor, así que se conseguirá un apantallamiento suficiente con esta configuración.

4.2.1. Escenario A

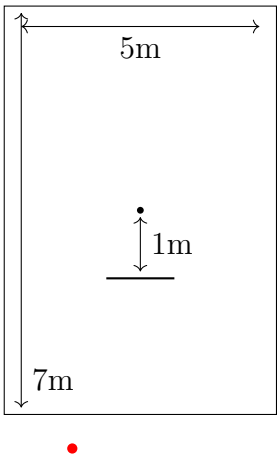
El primer escenario fue la utilización del laboratorio sin ningún obstáculo, de forma que se espera que los rayos impacten únicamente contra las paredes, puertas y demás mobiliario que se queda fuera de la trayectoria del robot.

Por ello, en este caso predominará el efecto del decaimiento con la distancia, salvo en puntos cercanos a los límites del laboratorio.

4.2.2. Escenario B

Para un segundo escenario se colocó una tabla forrada tal y como se describía anteriormente.

Se colocó centrada horizontalmente, 1 metro por debajo del centro de la trayectoria como se muestra en la Figura 4.6.



(a) Esquema de la posición de la tabla. El punto rojo indica la posición del emisor.

(b) Imagen del laboratorio en el escenario B.

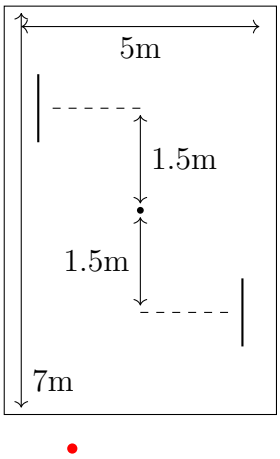
Figura 4.6: Disposición en el escenario B.

En este caso se busca que la tabla presente un efecto atenuante en toda la zona detrás de la tabla, creando una zona diagonal donde se deberían obtener valores de potencia menores que en el escenario A.

4.2.3. Escenario C

El tercer y último escenario tiene en cuenta dos tablas, esta vez colocadas con una orientación distinta.

Se colocan en este caso dos tablas a 1,5 metros del centro de distancia horizontal, a ambos lados. En cuanto a la distancia vertical, una de ellas se coloca a 1,5 metros por encima y la otra la misma distancia por debajo. –donde esta orientación hace referencia a la distribución indicada en la Figura 4.7–.



(a) Esquema de la posición de las tablas. El punto rojo indica la posición del emisor.

(b) Imagen del laboratorio en el escenario C.

Figura 4.7: Disposición en el escenario C.

A diferencia del caso anterior, en este escenario no se busca tanto el apantallamiento –aunque puede haberlo– sino el efecto de los rebotes.

Así, se espera que en la parte central pueda haber una potencia algo mayor que en el escenario A, aunque no debería haber un aumento significativo debido a la mayor distancia recorrida por los ángulos y el rebote que sufren.

5. Resultados

Una vez establecidas las bases teóricas y la disposición del entorno de pruebas, en esta sección se muestran los resultados obtenidos de la simulación y las medidas en el entorno real.

Además, también se indican los tiempos de ejecución de las versiones de ejecución en serie y en paralelo.

5.1. Resultados en las medidas y simulaciones

En esta sección se recogen los resultados obtenidos de la simulación de la propagación y de las medidas en el laboratorio en los escenarios descritos anteriormente.

Durante toda la sección se mostrarán representaciones en falso color sobre la potencia en entornos reales y en la simulación.

En todos los casos se deberá considerar al emisor en el punto (-1, -4) según el sistema de referencia mostrado. Además, todos los resultados se han obtenido realizando la simulación avanzando $^1/_8°$ en cada uno de los ejes, con 5 rebotes como máximo y una potencia mínima de -90dBm.

5.1.1. Escenario A

La Figura 5.1 incluye los resultados de las medidas por parte del robot en el laboratorio en el escenario A, sin ningún obstáculo.

De forma general se puede apreciar que el comportamiento dominante es el decaimiento de la distancia, con dos matices.

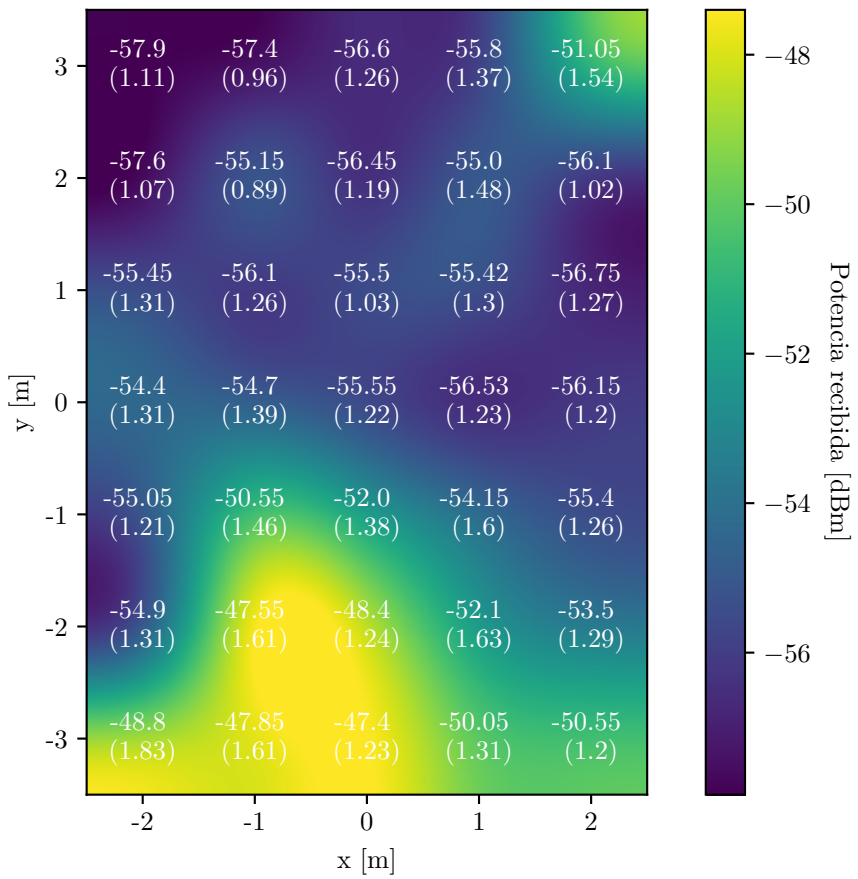


Figura 5.1: Resultado de 20 medidas en el laboratorio con el escenario A. [4] Entre paréntesis se indica la desviación estándar de los errores obtenidos en cada punto.

El primero es que, como se comentaba en la sección del fundamento teórico, la ley de Friis que modeliza este decaimiento es una aproximación que funciona mejor al alejar las antenas. En puntos cercanos entre el robot y el emisor los resultados son menos regulares, obteniendo en ocasiones incluso potencias menores aun estando a menor distancia, fruto de posibles segundos efectos entre la antenas que no se consideran en la ley de Friis.

[4]En esta representación y en todas las sucesivas figuran los valores de error obtenidos en los puntos del laboratorio donde se tomaron las medidas. A partir de ellos se ha realizado una interpolación con splines bicúbicos para obtener la distribución continua que se ha representado.

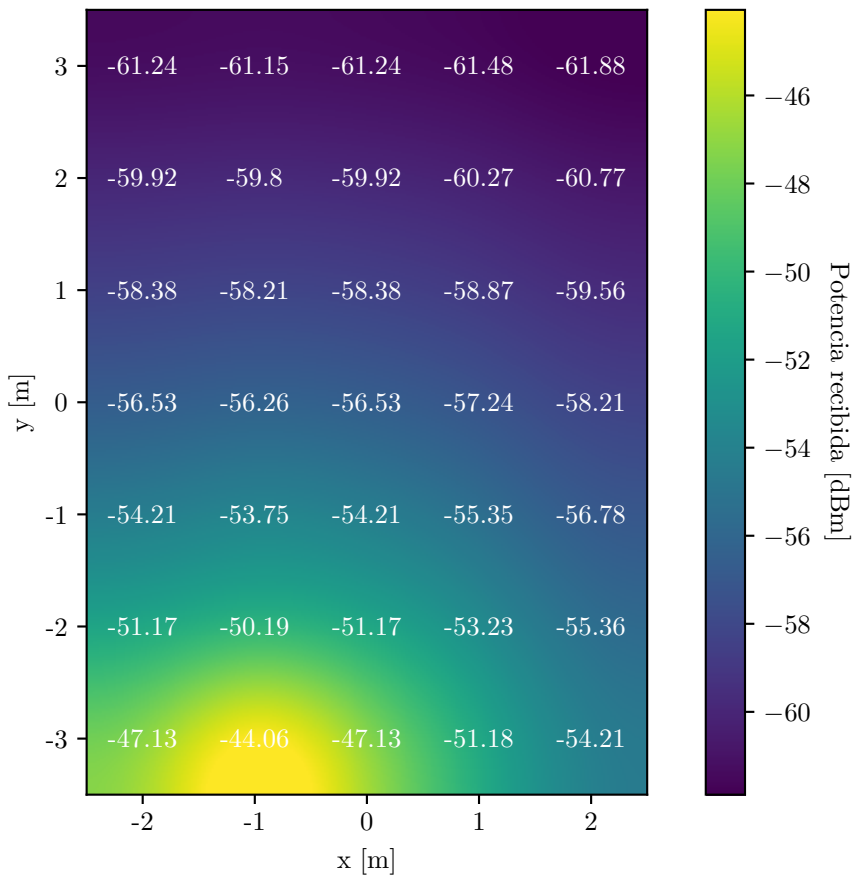


Figura 5.2: Resultado de la simulación para el escenario A.

La Figura 5.2 incluye los resultados de la simulación para el escenario A.

En este caso se observa de forma mucho más clara el comportamiento de los rayos al decaer exclusivamente con la distancia, al no tener aquí ningun tipo de contaminación ni al haberse considerado otros efectos.

Uno de las mayores discrepancias se pueden observar algunos de los puntos cercanos a la pared más colocada a la derecha –tal y como se indica en las Figuras–. En las medidas se obtienen valores más altos que en los puntos contiguos, lo que podría indicar reflexiones en la pared.

Este efecto no se aprecia en la simulación. La pared simulada no tiene una naturaleza metálica, por lo que es de esperar que sus coeficientes de reflexión no sean excesivamente altos y los rebotes no tengan un efecto significativo.

Debido a la estructura interna de la pared, es posible que contenga una estructura metálica de refuerzo que provoque estas discrepancias, aunque no fue posible comprobarlo en el laboratorio.

5.1.2. Escenario B

La Figura 5.3 incluye los resultados de las medidas por parte del robot en el laboratorio en el escenario B, en la que se colocó una tabla orientada horizontalmente –tal y como se representa en la Figura 4.6– centrada, y a un metro por debajo del origen.

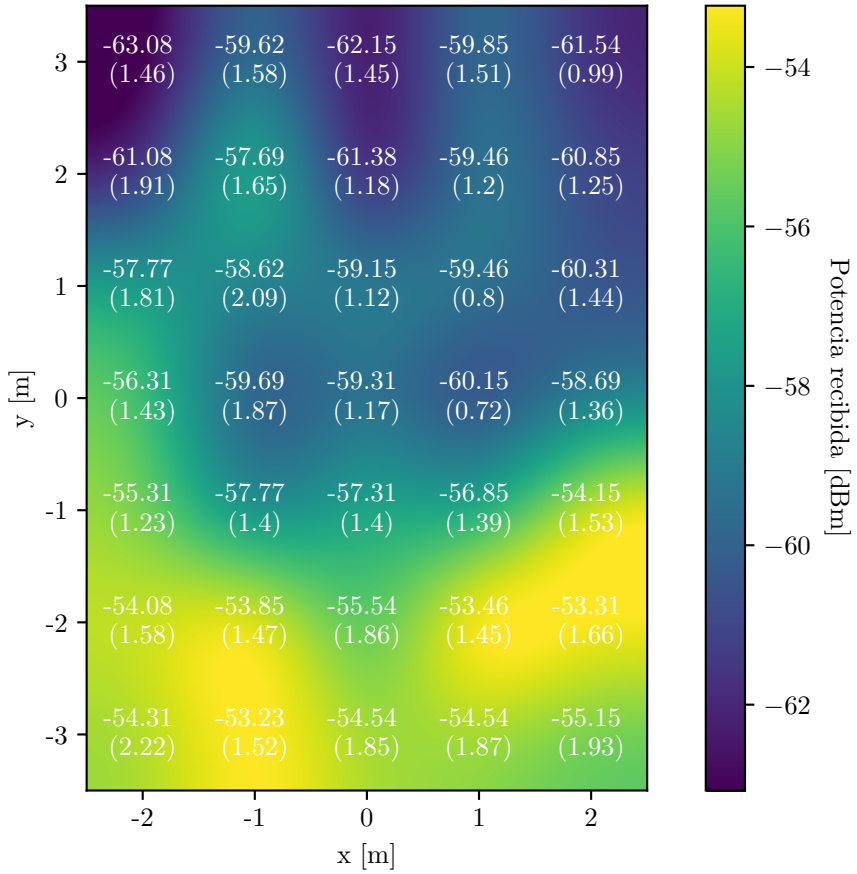


Figura 5.3: Resultado de 14 medidas en el laboratorio con el escenario B.

Es posible ver su efecto con claridad en la diagonal superior, teniendo en cuenta la posición del emisor. Aunque el apantallamiento está lejos de ser total, sí que se aprecia un decaimiento de la potencia decibida en esa zona comparado con el escenario A.

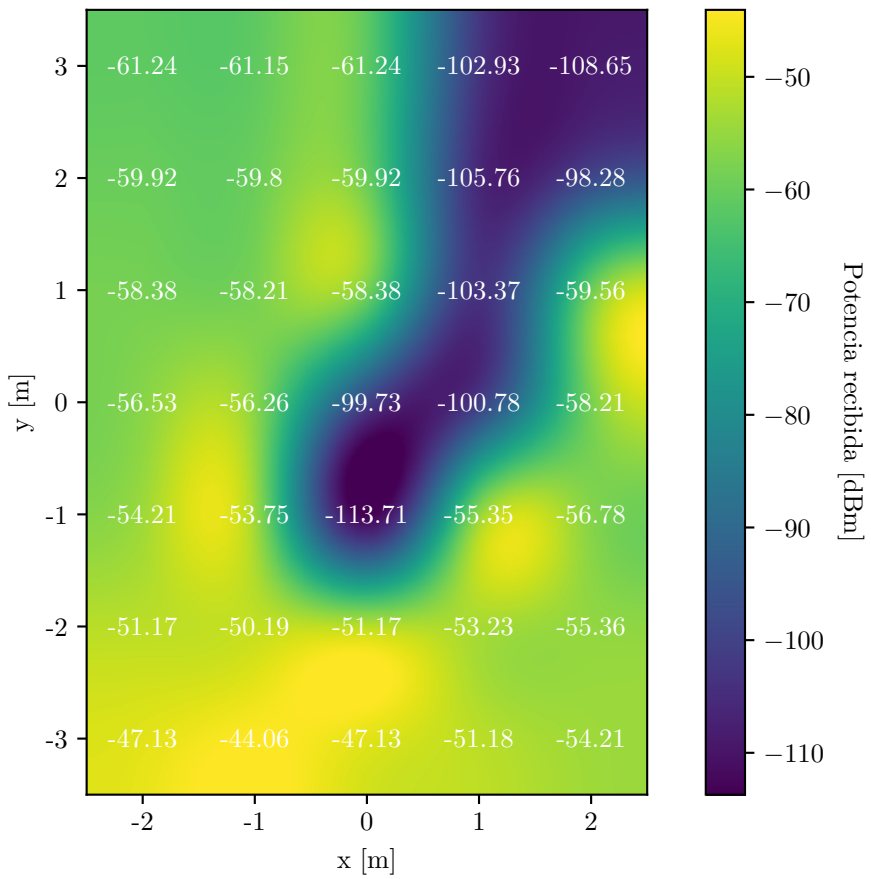


Figura 5.4: Resultado de la simulación para el escenario B.

La Figura 5.4 incluye los resultados de la simulación para el escenario B.

En este caso sí que se presentan discrepancias respecto a las medidas del laboratorio, especialmente en la zona detrás de la tabla. En el resto de puntos, como es de esperar, los resultados son similares a los del escenario A.

Esto se debe a la caracterización del papel de aluminio que forra la tabla. Para ello se recurrió al índice de refracción del aluminio, pero esta modelización falla al ser una fina capa del material en vez de un bloque sólido que se pueda ajustar mejor a un cambio de medio total.

Además, es posible que el papel no se componga exclusivamente de aluminio, lo que complica aún más determinar cuál debería ser su índice de refracción. La contaminación ya mencionada del ambiente añade una capa extra de artefactos que modifican el resultado final.

En todo caso, el simulador contempla la opción de colocar un obstáculo y predice de forma correcta la trayectoria de los rayos y su comportamiento.

5.1.3. Escenario C

La Figura 5.5 incluye los resultados de las medidas por parte del robot en el laboratorio en el escenario C, en la que se colocaron dos tablas orientadas verticalmente a 1.5 metros del origen en ambas direcciones tal y como se indica en la Figura 4.7.

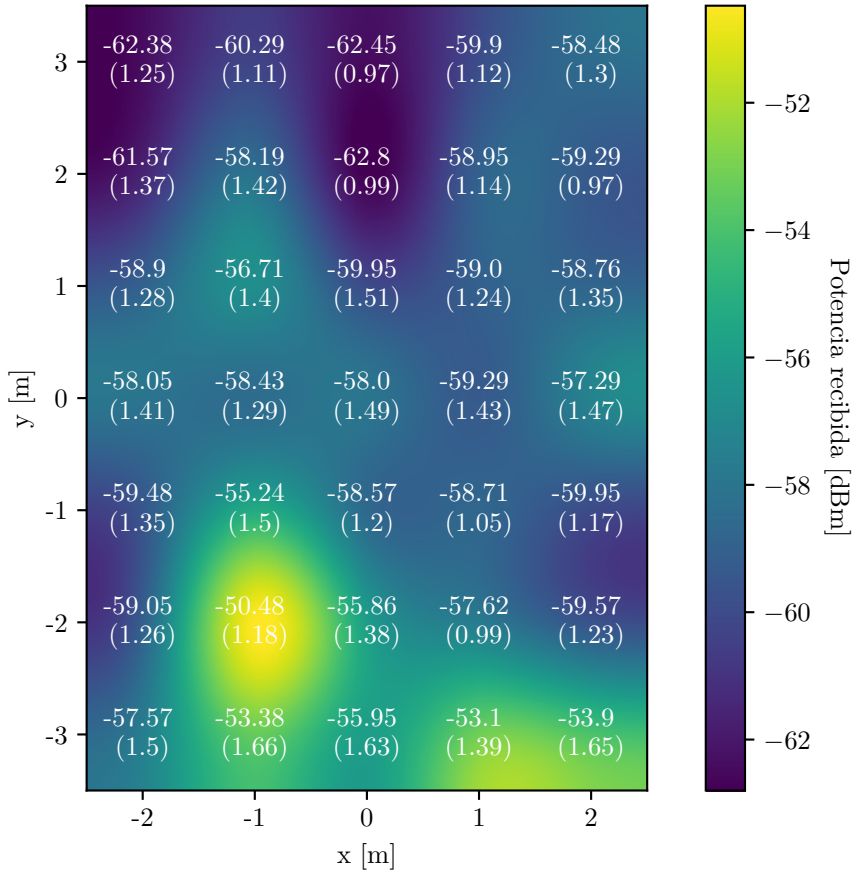


Figura 5.5: Resultado de 21 medidas en el laboratorio con el escenario C.

En este caso se esperaban dos diferencias respecto al escenario A: el primero es una menor potencia en el punto (2, -1) y alrededores, al estar detrás de una de las tablas –de la misma forma que ocurría en el escenario B.

Por otro lado, los rebotes producidos en esa misma tabla deberían producir un aumento de potencia en la banda central. Este efecto es algo menor debido a la mayor distancia que deben recorrer sumado a la potencia que pierden al rebotar.

Aun así, se puede observar como el decaimiento en esta zona es menor que en el escenario A, con lo que se puede apreciar el efecto mencionado.

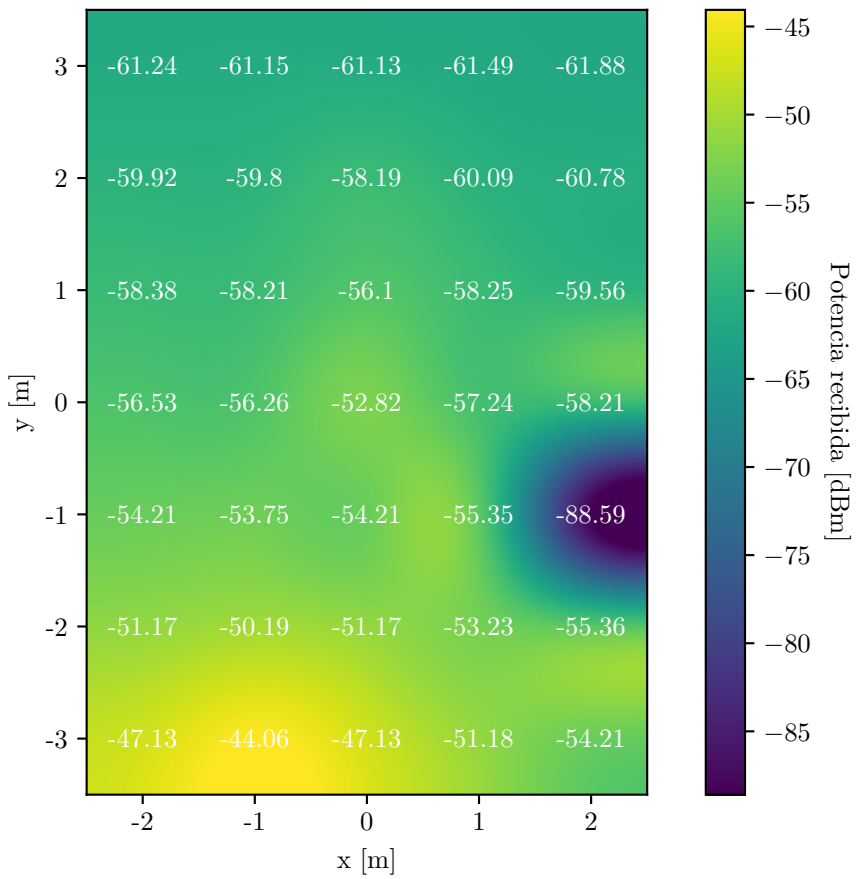


Figura 5.6: Resultado de la simulación para el escenario C.

La Figura 5.6 incluye los resultados de la simulación para el escenario C.

Aquí encontramos una situación parecida a la del escenario B: la tabla de la derecha tiene un apantallamiento mayor que el que tenía la tabla en el escenario real, por el mismo motivo al tener la misma parametrización.

En este caso se puede observar, como ocurría en el las medidas reales, unos valores más altos en la banda central –con $x = 0$ principalmente– debido a los rebotes en la tabla colocada a la derecha.

El valor tan bajo detrás de la tabla satura la escala de colores, pero se puede observar un decaimiento menor en la potencia, como era de esperar. Así, en este caso también se puede comprobar cómo el simulador predice de forma correcta la trayectoria de los rayos lanzados.

5.2. Aceleración del cómputo en paralelo

En la sección dedicada a la implementación de la simulación se distinguía entre dos modos de ejecución: en serie y en paralelo.

Para comparar el rendimiento entre las dos versiones se ha usado, en el caso de la GPU, la tarjeta gráfica Nvidia GTX1070 –como se comentaba en la sección anterior–, y para el caso del procesador, un Intel i7-4270HQ.

Var. angular (°)	CPU (ms)	GPU (ms)
0.25 (1/4)	2514.18	486.33
0.125 (1/8)	8468.546	1255.0
0.0625 (1/16)	34221.50	4715.6

Tabla 5.1: Tabla con los tiempos de ejecución. Todos los tiempos tabulados son la media de 3 ejecuciones.

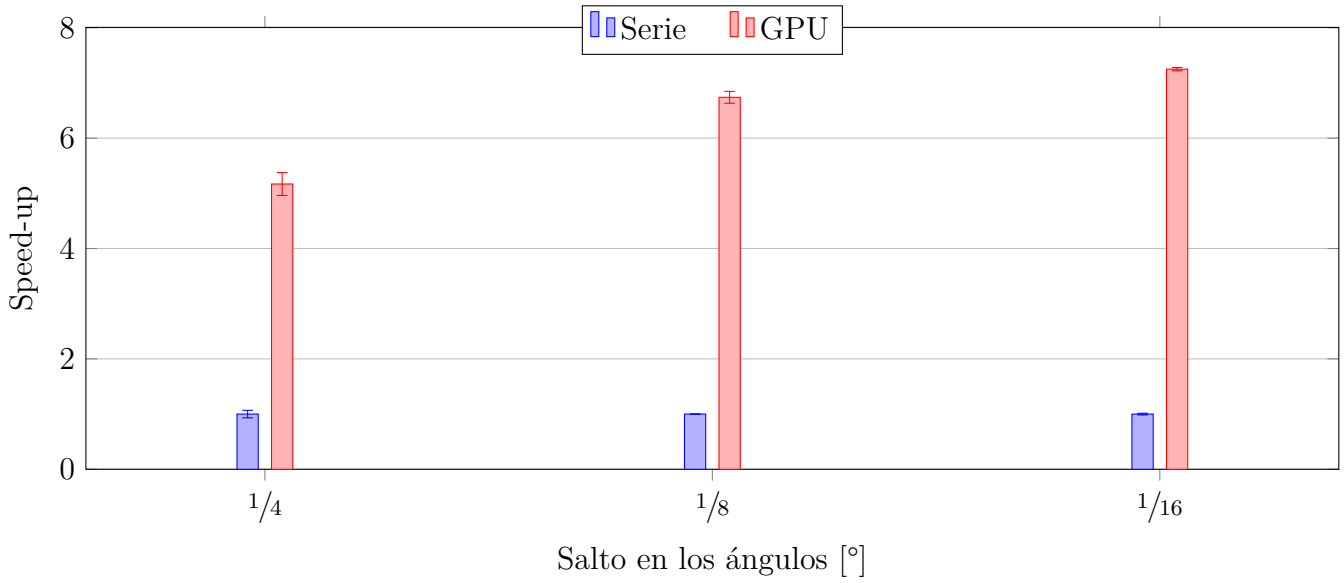


Figura 5.7: Representación del *speedup* de la ejecución en la GPU respecto a la CPU.

La Tabla 5.1 indica los tiempos de ejecución de ambas opciones, y en la Figura 5.7 se ha representado el *speedup* de la ejecución en la GPU, es decir, la razón entre los tiempos de ejecución de ambas versiones. Como se puede observar, la aceleración obtenida crece al aumentar el número de rayos lanzado, llegando a obtener tiempos de ejecución hasta 7 veces más bajos.

Hay que tener en cuenta que los tiempos de ejecución se refieren únicamente al tiempo de cómputo. En el caso de la GPU, debemos tener en cuenta que el traslado de datos desde

la memoria principal para el inicio del cálculo y el regreso de los datos una vez ha terminado llevarán un cierto tiempo.

Este tiempo es de unos 500ms aproximadamente en total. Así, para tiempos de ejecución de CPU menores no merecerá la pena realizar la ejecución en la GPU, pero en este caso, para saltos angulares de más de un cuarto grado sí que se pueden obtener grandes mejoras en el tiempo de ejecución incluso teniendo esta penalización en cuenta.

Además, hay que tener en cuenta que el mapa, los patrones de radiación de las antenas y la pila de rayos se almacenan en la memoria principal de la GPU. Es posible aprovechar las zonas de memoria más rápida para estos valores accedidos repetidamente y mejorar aún más los tiempos, aunque en el caso de la GPU usada no lo permitía al tener un espacio muy limitado.

Como se comentaba en secciones anteriores, la latencia de memoria a la hora de usar la GPU es muy importante, ya que la alta paralelizacion puede provocar cuellos de botellas en estas fases. Un correcto aprovechamiento de las herramientas disponibles podrían ayudar a obtener un rendimiento mayor.

5.2.1. Discusión de los resultados

Como se ha podido observar a lo largo de la sección, el método consigue una buena aproximación en general, aunque el encuentro con los obstáculos se presenta como el mayor problema de este tipo de simulaciones.

La caracterización de los materiales se antoja en general complicada por la gran casuística de materiales disponibles. Determinar cuál es el índice de refracción resulta muy complejo, aunque las diferencias entre materiales no magnéticos y no conductores es muy escasa.

Además, se añade la dificultad para determinar, especialmente en las paredes y el techo que delimitan el laboratorio, los posibles componentes que las atraviesan. Sean elementos de refuerzo estructural o bien parte de la red eléctrica, estos elementos tienen efectos en el comportamiento de la señal electromagnética que podrían no estar considerados en la simulación. La correcta modelización del entorno en el simulador se antoja crítica para obtener unos buenos resultados.

Por otro lado, las modelizaciones de las antenas también pueden llegar a tener un cierto efecto. Aunque su estructura esté bien delimitada y simulada, hay que tener en cuenta que el *firmware* que los fabricantes proporcionan es totalmente propietario y puede hacer que su comportamiento no sea el esperado.

Esto ocurrirá tanto para el emisor como para la antena receptora. Ahorros de energía o ciertas búsquedas para mejorar el rendimiento pueden jugar un gran papel a la hora de obtener unos buenos resultados, pero es imposible conocer estos comportamientos sin tener acceso al código o los algoritmos que los rigen.

En general, la ley del decaimiento se cumple de forma consistente salvo en puntos en los que la distancia entre las antenas sea baja. Como se explicaba anteriormente, este comportamiento es esperable siendo esas regiones zonas donde esta ley no se cumple de forma total.

La contaminación del espectro de WiFi en el laboratorio ha jugado también sin duda un gran papel en la toma de medidas. Aunque se ha logrado mitigar tomando un gran número de datos, su influencia sigue siendo alta como se puede observar especialmente en el escenario A.

Teniendo en cuenta todas estas fuentes de error, la discrepancia general no es demasiado abultada. La interacción con las tablas es el punto flaco: al ser su modelización compleja, tienen efecto tanto en los rayos transimitidos como en los reflejados, mientras que las paredes exteriores solo serán relevantes para los rebotes en el interior.

Además, la caracterización del papel de aluminio se antoja complicada: aun siendo un material conductor, el grosor es tan bajo –y además no ser sólido, sino estar únicamente recubriendo una placa de madera– que es difícil llegar a considerar si el coeficiente de refracción es correcto o es la propia estructura del material la que presenta un comportamiento errático.

En general, hay que tener en cuenta que el modelo es extremadamente simple: solo se tienen en cuenta rayos con unos pocos rebotes, sin ningún efecto más.

Ya se comentaba en la sección de fundamento teórico que no se tenían en cuenta más efectos en los bordes como la difracción o el *scattering*. Aun siendo prácticamente desprecables podrían ser uno de los efectos a añadir en caso de requerir una precisión mayor.

También, un estudio más pormenorizado de las antenas y su comportamiento, además de otro tratamiento para las distintas superficies y materiales podrían ser añadidos. Incluso no se incluyen en este caso posibles interferencias con otras redes, aunque el método de trazado de rayos hace complicado esta última opción al no tener en cuenta la trayectoria de la onda entre emisor y punto de impacto.

En general, queda claro que los efectos más significativos han sido considerados. Mejoras adicionales quedan fuera del Trabajo de Fin de Máster, aunque sería posible un estudio para evaluar si su efecto compensa la potencia de cómputo requerida para ellas.

6. Conclusiones

En este trabajo se ha desarrollado un modelo de emisión electromagnética con la aproximación de trazado de rayos para la modelización de propagación en interiores de señales de redes WiFi.

El uso de esta técnica se basa en el uso de rayos que se propagan en línea recta simulando el comportamiento del frente de ondas, de modo que se producirán rebotes en paredes y obstáculos. Esta característica la convierte en idónea para el caso de simulación en interiores como este caso.

Debido a la naturaleza independiente de los rayos se planteó la implementación en paralelo de modo que se puedan conseguir aceleraciones en los tiempos de ejecución de la simulación. Para ello se usaron GPUs, dispositivos idóneos para este tipo de tareas tan altamente paralelizables. Con ellas se pudieron conseguir tiempos de ejecución hasta 7 veces más bajos respecto a la ejecución en serie en un procesador usual.

Para determinar el comportamiento del simulador se compararon los resultados obtenidos con medidas en un entorno real tomadas con la ayuda de un robot.

Aunque en general los resultados se ajustan de forma razonable, la caracterización de los materiales del entorno resultó ser crítica a la hora de hacer que los rayos se comporten de la misma forma que lo hace la emisión en la realidad.

El entorno de pruebas estaba rodeado de otras redes, lo que contaminaba las medidas tomadas. Por ello es posible que la discrepancia entre la simulación y las medidas sea aparentemente mayor de lo que debería ser.

Con una buena parametrización de los materiales el decaimiento de la potencia con la distancia se ajusta bastante bien en la realidad, por lo que se podrían usar este tipo de simulaciones sin problemas en los entornos locales a los que están dirigidos.

La posibilidad de obtener el comportamiento de las redes WiFi en entornos locales, ampliamente usadas, abre la puerta a multitud de aplicaciones, desde la planificación de coberturas en entornos con muchos obstáculos hasta su uso para labores de posicionamiento local.

El hecho de poder automatizar la obtención de estos datos facilita estas labores enorme-

mente, eliminando la necesidad de tomar medidas en la zona de interés. Por ello, este método es más económico y rápido, además de permitir la obtención de datos desde prácticamente cualquier punto del entorno a evaluar.

Un modelo de interacción de los rayos más complejo que tenga en cuenta otros fenómenos como el la difracción, el *scattering* o incluso posibles absorciones de energía por parte de materiales podrían arrojar resultados más ajustados a la realidad, pero implicaría una mayor complejidad y un mayor coste computacional.

7. Bibliografía

[1] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[2] F. Firdaus, N. A. Ahmad, and S. Sahibuddin, "Accurate indoor-positioning model based on people effect and ray-tracing propagation," *Sensors*, vol. 19, no. 24, 2019.

[3] J. F. Hughes, A. van Dam, and Others, *Computer Graphics: Principles and Practice.* Addison-Wesley, 2014.

[4] C. A. Balanis, *Antenna Theory: Analysis and Design.* Wiley, 2016.

[5] F. Hossain, T. K. Geok, T. A. Rahman, M. N. Hindia, K. Dimyati, S. Ahmed, C. P. Tso, and N. Z. Abd Rahman, "An efficient 3-d ray tracing method: Prediction of indoor radio propagation at 28 ghz in 5g network," *Electronics*, vol. 8, no. 3, 2019.

[6] MATLAB, "Antenna Toolbox [en línea]." Dirección URL: <https://es.mathworks.com/products/antenna.html>. [Consulta: 31 de agosto de 2021].

[7] Nvidia Corporation, "CUDA C Programmig Guide: Features and Technical Specifications [en línea]." Dirección URL: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#features-and-technical-specifications>. [Consulta: 25 de junio de 2021].

[8] ROS Documentation, "move_base package [en línea]." Dirección URL: <http://wiki.ros.org/move_base>. [Consulta: 14 de julio de 2021].

Anexo

Debido a la complejidad del código dedicado a la simulación se ha optado por no incluir una copia del código en este documento.

Todo el código fue escrito en C++, aprevechando las librerías de CUDA para el caso de la ejecución en paralelo. Este código ha sido recogido en un repositorio de git –una herramienta de control de versiones– y colgado de forma pública en la URL:

https://github.com/dondanndy/RayTracerWiFiSimulator

En esta dirección es posible obtener todo el código necesario para la ejecución de la simulación, así como los patrones de radiación de las antenas y los archivos de mapa para los tres escenarios contemplados.