



# Localización robótica interior

Integrantes:

Nicolás Furquez, Oscar Saldías

Tutor:

Gonzalo Tejera

# Índice general

<b>Resumen</b>	<b>1</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Trabajos relacionados</b>	<b>3</b>
<b>3. Marco teórico</b>	<b>5</b>
3.1. Tecnologías	5
3.1.1. Banda ultra-ancha	5
3.1.2. Otras tecnologías aplicadas a la localización interior	6
3.1.2.1. Infrarrojo	6
3.1.2.2. Ultrasonido	6
3.1.2.3. Radiofrecuencia	6
3.1.2.4. Visión por computadora	9
3.1.2.5. Luz visible	9
3.1.2.6. Señales acústicas	9
3.2. Infraestructura de sistemas de posicionamiento	9
3.3. Técnicas y algoritmos de localización	10
3.3.1. Detección por proximidad	10
3.3.2. Lateración	10
3.3.2.1. Tiempo de vuelo	12
3.3.2.2. Diferencia de tiempo de llegada	13
3.3.2.3. Tiempo de viaje de ida y vuelta	14
3.3.2.4. Fase de llegada	14
3.3.2.5. Basados en la fuerza de la señal recibida	15
3.3.3. Angulación	16
3.3.4. Navegación por estima	17
3.3.5. Reconocimiento de patrones	18
3.3.5.1. Fingerprinting	18
3.3.6. Robotic Operating System	19
3.3.6.1. Paquete <i>Transform</i>	19
3.3.6.2. Paquete <i>Rosserial</i>	19
<b>4. Trabajo realizado</b>	<b>20</b>
4.1. Características de la tecnología elegida	20
4.2. Pruebas de concepto	22
4.3. Solución planteada	24
<b>5. Experimentación</b>	<b>31</b>
5.1. Efecto del incremento de dispositivos <i>anchors</i> en la precisión del sistema	34
5.2. Errores en la estimación de distancia entre dispositivos	36
5.3. Efecto de los obstáculos en la precisión del sistema	38
5.4. Efecto de la inclinación vertical de la antena en la precisión del sistema	41

5.5. Efecto de la topología en la precisión del sistema . . . . .	43
5.6. Efecto de la propagación multicamino . . . . .	48
<b>6. Conclusiones</b>	<b>49</b>
<b>7. Trabajo a futuro</b>	<b>51</b>
7.1. En dispositivos . . . . .	51
7.2. Desarrollo fuera de los dispositivos . . . . .	52
<b>A. Comparativa de tecnologías</b>	<b>54</b>
<b>B. Fuentes de error</b>	<b>56</b>
B.1. Relojes . . . . .	56
B.2. Ausencia de línea de visión . . . . .	56
B.3. Propagación multicamino . . . . .	56
B.4. Mala geometría . . . . .	57
<b>C. Inconvenientes encontrados</b>	<b>58</b>
<b>D. Proceso de calibración del parámetro Antenna Delay</b>	<b>62</b>
<b>E. Efecto de los obstáculos en la precisión del sistema (aplicando filtros)</b>	<b>64</b>
<b>Glossary</b>	<b>66</b>
<b>Acrónimos</b>	<b>71</b>
<b>F. Agradecimientos</b>	<b>78</b>

# Índice de figuras

3.1.	Diferencias entre una antena omnidireccionada y una sectorizada. El área de color gris corresponde al área de cobertura de cada tipo de antena. . . . .	10
3.2.	Lateración circular 2D. . . . .	11
3.3.	Lateración hiperbólica 2D. . . . .	12
3.4.	Tiempo de Llegada. Los emisores (RN) envían su marca de tiempo y el emisor (UE) calcula la distancia en función de la diferencia de tiempo. Todos los nodos deben tener sincronizados sus relojes. . . . .	13
3.5.	Localización basada en Tiempo de vuelo. Usando cuatro nodos emisores (RN) es posible calcular la posición del nodo receptor (UE) en un plano. . . . .	13
3.6.	Arreglo de antenas y cómo llega la señal en cada una de ellas. . . . .	15
3.7.	Angulación. . . . .	16
3.8.	Error potencial en la angulación. . . . .	17
3.9.	IMU de 9 grados de libertad. . . . .	18
4.1.	Diagrama de arquitectura utilizada por Localino. . . . .	22
4.2.	Proceso de relevamiento de puntos en el laboratorio de robótica. Se utilizó una estación total para este fin. . . . .	23
4.3.	Diagrama de arquitectura planteada. . . . .	25
4.4.	Flujo de comunicación entre <i>tag</i> , <i>anchor</i> y agente robótico; debajo, el pseudocódigo de los métodos. Se puede ver cómo se computa la distancia en el método <i>computeRangeAsymmetric</i> . 27	
4.5.	Representación gráfica de la técnica <i>Two-way Ranging</i> entre los dispositivos <i>A</i> y <i>B</i> . Es posible ver los tiempos que se tienen en cuenta en la ecuación 4.1. <i>RMARKER</i> deriva de <i>Ranging Marker</i> y es parte del estándar definido en IEEE802.15.4-2011. Representa el comienzo de la transmisión o recepción del mensaje y es utilizado para obtener las marcas de tiempo. . . . .	28
4.6.	Flujo del proceso de estimación de la posición del nodo estimador de posición. Este se dispara al llegar un mensaje de estimación de distancias desde el <i>tag</i> . . . . .	30
5.1.	Diagrama de posiciones de los puntos en el laboratorio utilizado en los experimentos. La grilla muestra una resolución de 1 m de lado. . . . .	32
5.2.	Varios lugares en los que se colocaron los dispositivos para los distintos experimentos. En algunas de las imágenes se resalta el dispositivo (en color) dentro de la imagen general (en blanco y negro). . . . .	33
5.3.	Datos graficados de la tabla 5.2, que muestran cómo se mueven los valores para error promedio, mediana del error, error mínimo y error máximo a medida que se aumenta la cantidad de <i>anchors</i> . Las líneas curvas son líneas de tendencia de cada serie; se puede ver que los mínimos del promedio y la mediana se dan entre 4 y 6 <i>anchors</i> . . . . .	35
5.4.	Histograma que muestra el error en el cálculo de distancias por parte de los dispositivos. . . . .	37
5.5.	Histograma que muestra la raíz cuadrada de la media; esta representa la variación del error en la muestra. . . . .	37
5.6.	Imagen de RViz de los puntos utilizados en el experimento. El punto 8 corresponde a la posición del <i>tag</i> ; en el resto de los puntos se colocaron los <i>anchors</i> . . . . .	38
5.7.	Gráfica de error según tabla de datos 5.4. . . . .	39

5.8.	Gráfica de desviación estándar según tabla de datos 5.4. . . . .	40
5.9.	Gráfica que muestra la variación de los valores de error en las medidas a medida que se cambia el ángulo del <i>tag</i> . . . . .	42
5.10.	Posiciones de los dispositivos en el escenario generado con la herramienta RViz. . . . .	44
5.11.	Posiciones de los dispositivos en el escenario generado con la herramienta RViz. . . . .	44
5.12.	Posiciones de los dispositivos en el escenario generado con la herramienta RViz. . . . .	45
5.13.	Posiciones de los dispositivos en el escenario generado con la herramienta RViz. . . . .	45
5.14.	Comparación de los valores de error promedio, mediana del error, error mínimo y máximo por escenario. Dado que el error es muy grande para varios escenarios, como se puede apreciar en la figura (a), se ajusta la escala para ver todos los valores de todos los casos, como se muestra en la figura (b). . . . .	47
5.15.	Comparación de los dos escenarios utilizados para el caso del efecto de la propagación multicamino. . . . .	48
C.1.	Ensamblado de componentes. . . . .	58
E.1.	Gráfica de error según escenario, aplicando los filtros en la estimación del punto. . . . .	65
E.2.	Gráfica de error según escenario, aplicando el promedio de los últimos cuatro valores en la estimación del punto. . . . .	65

## Resumen

El proyecto consiste en desarrollar un sistema de posicionamiento interior, para ser integrado en agentes robóticos. Esto nace de la limitación en la utilización del Sistema de Posicionamiento Global en ambientes interiores.

El trabajo comenzó con el relevamiento y análisis de tecnologías y técnicas aplicadas hoy en día en la academia y en la industria. Luego de este trabajo de investigación, se resolvió utilizar la tecnología de radiofrecuencia Banda Ultra Ancha, por no ser susceptible a los principales problemas que se presentan en la mayoría de las tecnologías usadas para localización, entre ellos, sensibilidad al cambio en el ambiente por movimiento de objetos o de personas, variación en la humedad o temperatura, entre otras.

Luego de determinar la tecnología base, se hizo una búsqueda de plataformas o soluciones ya existentes, priorizando los proyectos de código abierto tanto en aplicaciones como en especificaciones y diseño de componentes físicos y considerando la posibilidad de ensamblar parte de los elementos electrónicos. El resultado derivó en la utilización del dispositivo de Localino (v2), que consiste en una placa que se integra con la antena DW1000 de Decawave, es de uso en investigación académica y compatible con un proyecto de código abierto. Esto permitiría adaptar el código fuente, minimizando los tiempos, el costo de ensamblado de las placas de circuitos impresos y principalmente acotando imprevistos e incertidumbres. Estas fueron las principales ventajas respecto a otras opciones evaluadas, dado el alcance del proyecto y el poco conocimiento en el equipo en cuanto al diseño de esquemas de placas de circuitos integrados.

La solución desarrollada implicó el rediseño de la implementación ofrecida por Localino, que se basa en que los nodos de posición fija y conocida reporten sus distancias estimadas al nodo objetivo a un programa que se ejecuta en un computador. Dicha solución requiere que el agente robótico pueda estimar su propia posición, por lo que los nodos de posición fija y conocida deben reportar a este agente, que es a su vez el objetivo, sus distancias estimadas a él.

En una fase intermedia, se experimentó con los dispositivos para establecer la precisión del sistema en función del error que presenta la antena en la estimación de la distancia y analizar cómo esto influye en el cálculo de la posición. Para tal fin, se establecieron puntos de posición conocida en un ambiente controlado en el laboratorio de robótica.

Posteriormente, se desarrolló el módulo a ser integrado en la plataforma de trabajo del Sistema Operativo Robótico.

Finalmente, se evaluó la precisión y el rendimiento de todo el sistema integrado.

Uno de los objetivos opcionales que se manejaron en el proyecto fue la posibilidad de utilizar Docker como plataforma de virtualización para poder simular la ejecución en el agente robótico, lo que dio muy buenos resultados.

**Palabras clave:** localización interior, robótica, radiofrecuencia, banda ultra-ancha, tiempo de vuelo de ida y vuelta

# Capítulo 1

## Introducción

El siguiente proyecto de grado se enmarca en la siguiente realidad: *El vehículo que se desea obtener es una carretilla autónoma que carga cajones de vegetales hidropónicos en invernaderos que funcionan como viveros. Los invernaderos cuentan con mesas de producción en las que el personal cosecha y luego deposita el producto en cajones. Estos cajones se apilarán en el vehículo autónomo, el que deberá transportarlos por caminos rectos entre las mesas de producción, hasta un centro de acopio en el extremo del invernadero, donde el personal descargará los cajones, devolviendo los cajones vacíos a las mesas. Dentro del ambiente se contará con algún dispositivo que permita al agente autónomo localizarse y poder así armar su ruta de transporte.*

La motivación del proyecto es poder investigar las distintas tecnologías utilizadas en localización interior (p. ej.: *Wireless Access Points - WAPs, Beacons*, entre otras) aplicada a la robótica, para luego, de acuerdo a algún criterio de conformidad, seleccionar alguna y generar un prototipo robótico autónomo similar a la realidad planteada. A su vez, es necesario obtener métricas para conocer qué precisión tiene la solución implementada y poder compararla con otros sistemas. Para ello es necesario analizar qué trabajos hay en el ámbito académico e industrial y tener una aproximación al área de la localización. Un resumen de esta investigación de trabajos previos se encuentra en el capítulo 2.

Un sistema de localización interior es cualquier sistema que puede dar la localización precisa de un elemento dentro de una estructura cerrada, como pueden ser edificios, almacenes, hospitales, aeropuertos, entre otros (Farid et al., 2013). Según Küpper (2005), la localización es el proceso de obtener la posición espacial del objetivo. Un sistema de localización tiene como base la medición de un número de parámetros observables, esto incluye: velocidad, ángulo y diferencias de rango en las medidas. Estos parámetros, por lo general, miden la relación espacial de ciertos puntos fijos con el elemento a localizar. En estos sistemas es posible usar varias tecnologías, las que serán revisadas en la sección 3.1 haciendo hincapié en aquellas utilizadas para los ambientes interiores.

En la sección 3.2 se podrá ver una categorización de los sistemas de posicionamiento según el actor que realiza el cálculo de la posición. Luego de tener presente qué tecnologías existen hoy en día y qué es posible realizar con ellas, es necesario entender qué técnicas y algoritmos son utilizados para determinar una posición. En esta investigación se encontró que muchos de estos algoritmos aprovechan ciertas características de las tecnologías (sec. 3.3) e incluso es posible combinar varias de ellas para mejorar la precisión del sistema.

Habiendo explorado todas las posibles tecnologías a utilizar y teniendo una idea clara de los requerimientos necesarios para implementar un sistema de localización interior, es posible decidirse por una. La tecnología seleccionada para la realización de este trabajo es banda ultra-ancha (de acá en más se

utilizará UWB, por sus siglas en inglés de Ultra-wideband), por ser la que da mejores resultados y por las capacidades que tiene. Luego, con esta tecnología se armó un prototipo de sistema de localización, cuyo proceso se puede ver en la sección de trabajo realizado (cap. 4).

El sistema de localización interior se implementará como un sistema de *localización basada en la terminal* (sec. 3.2), donde se identifican dos actores principales: uno es el agente que requiere conocer su posición, llamado receptor, etiqueta o *tag* (en este documento se utilizará *tag*), y otro llamado antena, baliza o *anchor* (se utilizará este último término como referencia en el resto del documento), de posición fija y conocida. El *tag* interactúa con tres o más *anchors* para poder determinar su posición, utilizando *lateración circular* (sec. 3.3.2).

Al momento de implementar un sistema y de tener seleccionadas las tecnologías y técnicas a usar, es necesario entender qué posibles fuentes de error pueden afectarlo. Dentro del anexo B se describen las fuentes de error más comunes en sistemas de localización.

Cuando se tiene el sistema implementado, es interesante ver cómo se comporta bajo ciertas condiciones y experimentar con él. Por esto, en el capítulo 5 se describen varios experimentos donde se ponen a prueba el sistema y la tecnología, en especial si esta se comporta como el fabricante dice que lo hace en cuanto a capacidades y resultados.

Con los resultados de los experimentos, es posible generar una opinión en cuanto a la tecnología aplicada y el prototipo creado. En el capítulo 6 se puede ver este análisis con más profundidad, pero, en pocas palabras, los resultados del uso de la tecnología fueron satisfactorios y esta cumple con los puntos deseables para la localización interior.

Como en todo proceso de investigación, se encontraron mejoras a realizar que se pueden ver en el capítulo 7, necesarias para poder continuar utilizando el proyecto a futuro y que escapan al alcance de este.

## Capítulo 2

# Trabajos relacionados

En este capítulo se abordan los trabajos más relevantes de forma más general, que fueron evaluados en la investigación previa y que establecen los fundamentos del presente proyecto. Para abordar más en profundidad el estado de la ciencia y la tecnología en el área de la localización, en especial la interior, remitirse al documento del Estado del Arte del proyecto (Furquez y Saldías, 2019).

Con respecto a la localización interior, se han encontrado diversos trabajos tanto en el área robótica como en el campo de localización personal y es en este último donde se logró encontrar la mayor cantidad de trabajos que utilizan una gran variedad de tecnologías. La búsqueda de trabajos relacionados se centró en un principio en las tecnologías más generales para la localización interior, para luego, teniendo una visión general de las tecnologías disponibles, profundizar en aquellas con aplicación específica o más adecuada a la robótica.

En lo que concierne a la robótica, se pueden aplicar mecanismos muy variados con respecto a la localización interior. En Munishwar et al. (2009), los autores proponen un robot capaz de localizarse por medio de etiquetas *RFID* (sec. 3.1.2.3), las que son instaladas en el piso del ambiente en posiciones conocidas. Luego, el robot puede estimar su posición por medio de lecturas consecutivas de las distintas etiquetas. Este tipo de localización es similar a la *Navegación por estima* (sec. 3.3.4) y puede generar acumulación de errores si la densidad de etiquetas es muy pequeña. Aplicando la misma tecnología (*RFID*), en DiGiampaolo y Martinelli (2018) lograron utilizar etiquetas tanto para ubicarse en un almacén como para la localización de productos en una estantería. Este caso utiliza análisis de señales de la respuesta para evitar algunos problemas que surgen del uso de tecnologías de radiofrecuencia (anexo B). Estos autores también refieren a la utilización de la fase de la señal para crear otro sistema robótico de localización con *RFID*, pero con las etiquetas en el techo (DiGiampaolo y Martinelli, 2014).

Otra de las tecnologías empleadas en la robótica consiste en la utilización de elementos preexistentes en el ambiente, como por ejemplo puntos de acceso o señales *WiFi*, que se encuentran normalmente para uso general. En Gu et al. (2016) utilizan la técnica de *Fingerprinting* (sec. 3.3.5.1), la que, en pocas palabras, consiste en construir un mapa con mediciones de las señales *WiFi* en ciertos puntos conocidos del ambiente y luego interpolar para todo el ambiente. Este mecanismo tiene como desventaja que es propenso a varios tipos de errores; el principal es que si existen cambios en el ambiente (muebles, personas, etc.), los valores no van a ser los mismos que los medidos previamente, por lo que es necesario realizar mediciones del ambiente y actualizar el mapa periódicamente.

Con respecto al uso de *Fingerprinting*, no es solo utilizado con señales *WiFi*; en Moghtadaiee et al. (2011) se puede ver la aplicación de esta técnica con señales de radio *FM*. El *Fingerprinting* también es utilizado con otra de las tecnologías más comunes encontradas: *Bluetooth*, en especial la versión BLE (sec.

3.1.2.3). Por ejemplo, en Pu y You (2018) se utiliza esta técnica, aplicando una ecuación relativamente simple para determinar la distancia a partir del Indicador de Fuerza de la Señal Recibida (*Basados en la fuerza de la señal recibida*, sec. 3.3.2.5), además de aplicar una aproximación por clasificación por el vecino más próximo para mejorar el rendimiento del sistema de localización.

También se encontró un trabajo realizado por estudiantes de Ingeniería de UdelaR, que utiliza señales electromagnéticas (Acevedo et al., 2018). En este caso, el objetivo es localizar ganado dentro de un área abierta y bastante extensa (del orden de hectáreas) y poder así estudiar su comportamiento. A los animales se les coloca un pequeño dispositivo, que cada cierto tiempo emite una señal que es captada por antenas georreferenciadas. Estas antenas informan el tiempo de llegada de la señal recibida a un servidor central que realiza los cálculos de localización. En este caso se utiliza *LoRa*, que es una tecnología emergente en el área de *IoT* (*Internet of Things*). La señal es capaz de viajar grandes distancias casi sin verse afectada por interferencias; por esta razón, *LoRa* también es usada en la comunicación espacial y militar. Dicha tecnología tiene alta latencia, lo que hace inviable su utilización en aplicaciones de tiempo real como la localización robótica interior. Otra desventaja es que en espacios más reducidos, como puede ser dentro de una edificación, la precisión puede verse afectada.

# Capítulo 3

## Marco teórico

A continuación se presenta el marco teórico que fue utilizado para desarrollar la solución al problema de la localización interior. Primero son abordadas las tecnologías utilizadas en los trabajos mencionados en el capítulo anterior, para relevar sus puntos fuertes y sus debilidades en la temática. Posteriormente, se revisan las técnicas más comunes en relación con la localización en general y cómo se aplican. También se analizan cuáles son mejores, dependiendo de la tecnología a elegir.

Al momento de implementar un sistema de localización y de tener seleccionadas las tecnologías y técnicas a usar, es necesario entender qué posibles fuentes de error pueden afectarlo. Por ejemplo, dentro de un invernadero es muy probable que la humedad sea elevada o por lo menos variable, en comparación con otros ambientes, y ciertas tecnologías de radiofrecuencia se vean afectadas por la humedad variando la señal, lo que podría afectar la precisión y por lo tanto debe ser tenido en cuenta para intentar minimizar el efecto en el sistema. Otro tipo de fuentes de error que afecta a los sistemas que utilizan el tiempo de propagación para estimar la distancia a un objetivo es el problema de la sincronización de los relojes. Dentro del anexo B se describen esta y otras fuentes de error en sistemas de localización.

Es posible encontrar más información en el documento de Estado del Arte del proyecto (Furquez y Saldías, 2019).

### 3.1. Tecnologías

Dentro de la localización interior se usan muchas tecnologías; la utilizada en este proyecto fue UWB, la que se describe en la siguiente sección. Luego se expondrán otras tecnologías que son utilizadas en la industria y la academia para resolver la localización interior.

#### 3.1.1. Banda ultra-ancha

Esta tecnología usualmente se conoce con el nombre de *Ultra-wideband (UWB)*. En esta se transmiten pulsos extremadamente cortos, cada uno en un período de tiempo menor a un nanosegundo, en el rango de frecuencias que van de 3,1 a 10,6 GHz, sobre un ancho de banda mayor a los 500 MHz. Utiliza un ciclo de trabajo muy bajo, lo que resulta en un reducido consumo de energía. Como consecuencia, la resolución del tiempo es muy alta, lo que la hace ideal para uso en sistemas de posicionamiento.

En un principio esta tecnología fue utilizada para sistemas de comunicación de corto alcance, tal como periféricos de computadores y otras aplicaciones de uso interior. La *UWB* ha sido especialmente atractiva para la localización interior debido a que también es inmune a las interferencias de otras señales, por su marcada diferencia en el tipo de señal y su espectro de radio. Mientras que la mayoría de las tecnologías de

radiofrecuencia presentan grandes problemas en presencia de obstáculos, la señal de *UWB* (en particular la de baja frecuencia) puede penetrar una gran variedad de materiales, incluyendo muros, aunque los metales y líquidos pueden interferir con su señal.

A lo anterior, se añade que al ser tan corta la duración de los pulsos, es menos susceptible a los efectos causados por la interferencia de *Propagación multicamino* (sec. B.3), lo que permite identificar la señal del camino principal en la presencia de señales provenientes de múltiples direcciones. Además, esta misma característica hace improbable la existencia de interferencias constructivas y destructivas o el desplazamiento de fase de la onda. Todo esto se traduce en una estimación más precisa al usar técnicas basadas en la medición del tiempo de propagación de la señal (secciones 3.3.2.1, 3.3.2.2 y 3.3.2.3).

Algunos trabajos aseguran que esta tecnología puede alcanzar una precisión de entre dos y diez centímetros, dependiendo del fabricante de la antena (Yavari y Nickerson, 2014).

Sin embargo, a pesar de haber sido propuesta inicialmente para el uso en redes de área personal (*Personal Area Network - PAN*), el lento progreso en el desarrollo de estándares de *UWB* ha limitado su uso a productos de consumo y dispositivos portátiles de usuario (Zafari et al., 2017).

### **3.1.2. Otras tecnologías aplicadas a la localización interior**

#### **3.1.2.1. Infrarrojo**

Los sistemas basados en luz infrarroja (*IR*) por lo general son usados dentro de una habitación o en espacios sin obstáculos. Esto se debe a que el *IR* tiene corto alcance, no atraviesa paredes ni puertas. Depende fuertemente de lo que se denomina *línea de visión (Line of Sight - LoS)* directa entre el emisor y el receptor. Su aplicación más común es la detección de usuarios en ciertas ubicaciones utilizando módulos emisor/receptor, los que se activan por el movimiento.

Una de las principales ventajas radica en su portabilidad: es liviano, pequeño y fácil de acoplar a una plataforma móvil. Como desventajas tiene la limitación del alcance de señal, lo que le impide ser preciso a la hora de determinar la posición, además de verse afectado por interferencias con la luz fluorescente y solar.

#### **3.1.2.2. Ultrasonido**

Esta tecnología utiliza señales de sonido por encima de los 20 KHz, fuera del espectro audible por los humanos. Es utilizado aplicando la técnica de *Tiempo de vuelo* (sec. 3.3.2.1). La principal desventaja que posee es que no puede atravesar paredes y es sensible a las condiciones ambientales como la humedad y la temperatura, aunque esto puede ser compensado usando sensores de humedad y temperatura, según Zafari et al. (2017). Estos dispositivos deben tener una alta tasa de actualización de la señal, lo que implica un impacto en la duración su batería. Otro factor que impacta en el consumo general es la alta capacidad de cómputo requerida para procesar las señales acústicas. En Gualda et al. (2019) utilizan esta tecnología en un sistema de balizas para determinar un recorrido de un robot móvil.

#### **3.1.2.3. Radiofrecuencia**

La *Radiofrecuencia (RF)*, también conocida como espectro de *Radiofrecuencia*, provee las bases para la mayoría de los métodos de posicionamiento utilizados. Esto se logra explotando ciertas características físicas de las señales de radio, en especial la velocidad en que estas viajan o la pérdida y atenuación de la señal que experimentan al viajar del emisor al receptor.

La principal ventaja de utilizar *Radiofrecuencia* para la localización es que las ondas, dependiendo de la frecuencia, pueden atravesar paredes, muebles, el cuerpo humano, entre otros materiales, por lo que estos no son un obstáculo. Además, muchas de las tecnologías tienen un amplio rango de cobertura y requieren de poco equipamiento. La *Radiofrecuencia* se puede clasificar en dos tipos:

- Banda corta: donde se encuentran las tecnologías *RFID*, *Bluetooth*, *WLAN*.
- Banda ancha: por ejemplo, *UWB*.

A continuación se listan las principales tecnologías de radiofrecuencia que son utilizadas para localización interior:

#### ■ **RFID**

La tecnología *Radio Frequency Identification (RFID)* es usada en un gran abanico de aplicaciones, desde seguridad hasta la cadena de ensamblaje en la industria automotriz. Consiste en un lector que puede interpretar etiquetas mediante señales electromagnéticas. Es una tecnología que no es afectada por el problema de *Ausencia de línea de visión* (anexo B.2) para comunicarse, dado que usa radiofrecuencia. Se define un protocolo común entre los lectores y las etiquetas para la comunicación. Las etiquetas se pueden categorizar en dos tipos:

- **Activas:** Estas trabajan por lo general en la banda de *UHF* y frecuencias microondas. Las etiquetas codifican su identificador en la señal y pueden ser leídas por los lectores a cientos de metros. Son utilizadas para localización y seguimiento de objetos, tienen buen rango de alcance, su costo es relativamente asequible y pueden ser embebidas en sistemas pequeños.
- **Pasivas:** Estas no emiten señal alguna de forma activa, sino que la señal es emitida al ser estimuladas por un lector, por lo que no necesitan de fuentes de alimentación para operar. Tienen, por lo tanto, un rango pequeño de alcance, son más pequeñas, livianas y más económicas que las activas. Pueden operar en frecuencias bajas, altas, *UHF* y microondas. Para que su aplicación sea factible en localización, sería necesario un gran número de etiquetas, como se puede ver en Zafari et al. (2017).

Según Ahson y Ilyas (2011), para el posicionamiento con *RFID* es necesario tener en cuenta un par de aspectos en relación con la infraestructura del sistema: el costo de la administración y el despliegue de las etiquetas en el ambiente. Esto último debe ser minimizado, en especial si se cuenta con una gran superficie a cubrir. A su vez, las etiquetas deben ser pasivas y de largo alcance, esto es, que cada etiqueta disponga de una antena diseñada para largo alcance. Los autores también indican que es difícil encontrar sistemas *RFID* con estas características de forma estándar, por lo que utilizan etiquetas activas para aumentar el alcance, lo que implica un aumento en el costo del sistema y en la complejidad del equipamiento requerido.

#### ■ **Bluetooth**

*Bluetooth* es una tecnología de radiofrecuencia de corto alcance utilizada en redes *WPAN (Wireless Personal Area Network)* para la comunicación entre dispositivos personales. El estándar más nuevo disponible es el llamado *BLE*, por *Bluetooth Low Energy*, también denominado *Smart Bluetooth*, especialmente diseñado para tener muy bajo consumo de energía.

A pesar de ser definida como de corto alcance, la especificación *BLE* tiene un alcance de entre 25 y 100 metros. Es una tecnología que tiene como aspecto positivo que es barata, segura y de bajo consumo. Las versiones anteriores a *BLE* tienen como desventaja en su aplicación a la localización la gran latencia al conectarse y desconectarse. Esto las hace inadecuadas para su utilización en sistemas con requerimiento de posicionamiento en tiempo real. Los dispositivos *BLE* pueden ser configurables para que emitan cada cincuenta milisegundos, lo que disminuye el tiempo de detección y conexión y los mejora en comparación con los de sus predecesoras.

Es común su utilización en dispositivos de proximidad, pero en algunos estudios se ha encontrado que al aplicarlo en la localización tiene gran precisión, en el orden de pocos centímetros. Las técnicas que más se utilizan en conjunto con esta tecnología son: *Basados en la fuerza de la señal recibida*

(sec. 3.3.2.5), *Angulación* (sec. 3.3.3) y *Tiempo de vuelo* (sec. 3.3.2.1).

Según Zafari et al. (2017), actualmente existen dos soluciones comerciales que utilizan *BLE* como sistema de proximidad o de localización: uno es *iBeacon* de Apple Inc. y el otro *Eddystone* de Google Inc.

#### ■ Basadas en WLAN

El estándar IEEE 802.11 (IEEE, 2016b), más conocido como *WiFi* o *WLAN* (*Wireless Local Area Network*), es una tecnología de radiofrecuencia que opera en la banda *ISM* (*Industrial, Scientific and Medical*) y es principalmente usada para proveer conectividad entre los distintos dispositivos, en el ambiente público, privado y comercial. *WiFi* por lo general tiene un alcance no mayor a 100 metros, pero se ha extendido el rango a cerca de un kilómetro gracias al estándar IEEE 802.11ah (IEEE, 2016a) y ha sido principalmente optimizada para aplicaciones de *IoT*.

Hoy en día, la mayoría de los dispositivos de uso cotidiano, como *smartphones*, *laptops* e incluso dispositivos para robótica o domótica, incluyen esta capacidad de fábrica, lo que la hace una tecnología ideal en el área de la localización interior.

Al tener una cantidad suficiente de puntos de acceso en el ambiente donde se utilice el sistema de localización, estos pueden ser usados como puntos de referencia de tal manera que el sistema de localización puede ser instalado sin ninguna infraestructura extra, a diferencia de *RFID* o *Bluetooth*.

De todas formas, la principal razón de la existencia de la tecnología *WiFi* es la comunicación, por lo que el estándar no incorpora características específicas para su uso en la localización. Por lo tanto, su utilización en la localización debe ser implementada encima de la capa de aplicación. Asimismo, al no estar controlada la banda *ISM*, es prácticamente inevitable la incidencia de interferencias provenientes de otros dispositivos, por lo que este tipo de problemas afectan a las técnicas que utilizan esta tecnología, influyendo en la exactitud en el proceso de localización (ver anexo B por más detalles).

Las técnicas *Basados en la fuerza de la señal recibida*, *Tiempo de vuelo* y *Angulación* (ver sección 3.3), o incluso una combinación de ellas, son las ideales para ser usadas en conjunto con esta tecnología.

#### ■ Estándar IEEE 802.11ah

Este estándar en particular es una de las tecnologías emergentes para su utilización en *IoT* y está relacionado con el estándar *WiFi*. También es llamado *WiFi HaLow*. Principalmente, fue diseñado para tener un rango de alcance mayor que el del IEEE 802.11 y se basa en una capa física *MIMO-OFDM*, que es una norma de radiofrecuencia que permite transmitir múltiples señales simultáneamente por el mismo canal, en la que cada señal tiene un rango único de frecuencia y es modulada por los datos (más información sobre esta norma en Vargas y López (2007)).

*WiFi HaLow* puede operar bajo varios modos de transmisión en el espectro del subgigahertz, usando un ancho de canal de 1, 2, 4, 8 o 16 MHz. Puede transmitir de varias maneras, desde los 150 Kbps pensado para dispositivos pequeños a batería y que deban tener gran cobertura, como ser sensores de humedad y temperatura, hasta los 346 Kbps, que puede ser usado por dispositivos en los que el consumo de energía no sea una limitante.

Esta tecnología cuenta con una baja tasa de pérdida por propagación, tanto en espacios abiertos como con obstáculos en el ambiente, gracias a su baja frecuencia de operación.

Otro punto a destacar es que no es necesario tener equipamiento muy específico y costoso: solo es necesario un dispositivo de ruteo y dispositivos que sean compatibles con la norma.

#### 3.1.2.4. Visión por computadora

Al usar visión como tecnología de posicionamiento, se hace referencia al procesamiento de las imágenes que se sensan a través de una o más cámaras y se estima el posicionamiento aplicando alguna técnica, como por ejemplo odometría visual (sec. 3.3.4). Otra manera de utilizar el componente visual, es en la localización por medio de marcas en el ambiente, en posiciones específicas conocidas, donde cada marca oficia de baliza. Al ser detectada por el componente de visión es posible deducir la ubicación de quien utilice el sistema.

#### 3.1.2.5. Luz visible

El utilizar la luz visible como mecanismo de comunicación refiere a una tecnología emergente denominada *VLC* (*Visual Light Communication*). Esta transfiere información a velocidades de entre los 400 y 800 THz, modulada y emitida por *LEDs*. La utilización de esta tecnología para localizarse consiste en tener un sensor de luz y detectar la posición y la dirección de los *LEDs* emisores. Básicamente, su funcionamiento es similar al de las tecnologías *BLE*: en *VLC* estos diodos continuamente emiten señales y el receptor las capta e identifica qué nodos pueden localizarse. La mejor técnica utilizable para esta tecnología es *Angulación* (sec. 3.3.3).

Puede verse en Ye-Sheng et al. (2014), donde se utiliza la cámara de un teléfono inteligente que es capaz de decodificar, a partir de la captura de una sola imagen, la identificación y ubicación que es transmitida por *LEDs* a través de pulsos ópticos imperceptibles al ojo humano, y determinar su ubicación y orientación relativa a los *LEDs*. Estos *LEDs* fueron modificados para transmitir la información mediante parpadeos rápidos. La captura y procesamiento de imágenes consecutivas permitiría la actualización continua de su posición.

#### 3.1.2.6. Señales acústicas

Para diferenciar esta tecnología del *Ultrasonido* (sec. 3.1.2.2), en esta se hace referencia a las señales que están por debajo de los 20 Hz (el espectro audible); es por esto que basta con tener un micrófono como sensor y hacer un procesamiento de señales acústicas. En este caso, cada nodo con posición conocida emite su señal y es procesada, lo que permite localizar al receptor. Por lo general, se aplica la técnica *Tiempo de vuelo* (3.3.2.1), en la que estas balizas emiten alguna marca de tiempo decodificable. Según los autores de Zafari et al. (2017), este mecanismo posee alta tasa de exactitud, a pesar de tener un sensor de pobres prestaciones como es el caso del micrófono de los teléfonos inteligentes. Como desventaja, necesita una infraestructura específica, lo que implica un aumento del costo. Además, puede generar polución sonora, por lo que los dispositivos emisores deben emitir lo más bajo posible. Otro problema que debe enfrentar esta tecnología es el alto poder de procesamiento en el receptor que requiere la señal, que puede ser muy costoso en poder de cálculo y por lo tanto redundar en un alto consumo energético.

## 3.2. Infraestructura de sistemas de posicionamiento

Según Küpper (2005), es posible categorizar la infraestructura de los sistemas de posicionamiento de muchas maneras. La categorización que se expone a continuación es en función de qué actor es responsable de la realización de las mediciones y los cálculos de posición. Dicha clasificación se divide en tres tipos:

**Basadas en la terminal:** En este caso, todo lo que implica el posicionamiento, es decir, la medición, cálculo y mapeo, es realizado por el sistema móvil o agente del que se requiere conocer su posición. Un ejemplo de esto es el *Global Positioning System (GPS)*: el receptor *GPS* mide las señales emitidas por los satélites específicos para este fin, calcula su ubicación y se la da al dispositivo que tenga el receptor.

**Basadas en la red:** En este caso, las mediciones y los cálculos son realizados por la infraestructura de la red del sistema de localización. Por lo general, estos sistemas poseen un servidor central, que se encarga de hacer los cálculos y dejar la información disponible para su uso. Un ejemplo de esto es el sistema *LA-2000* de *Trapeze Networks*, que rastrea la posición de todos los nodos *WiFi* que están dentro de un recinto, permitiendo conocer su ubicación en todo momento.

**Terminal asistida:** En esta categoría entran todos aquellos sistemas en donde la terminal realiza las mediciones necesarias. Luego la información es transmitida a la red, que hace los cálculos, para finalmente pasar el resultado a la terminal.

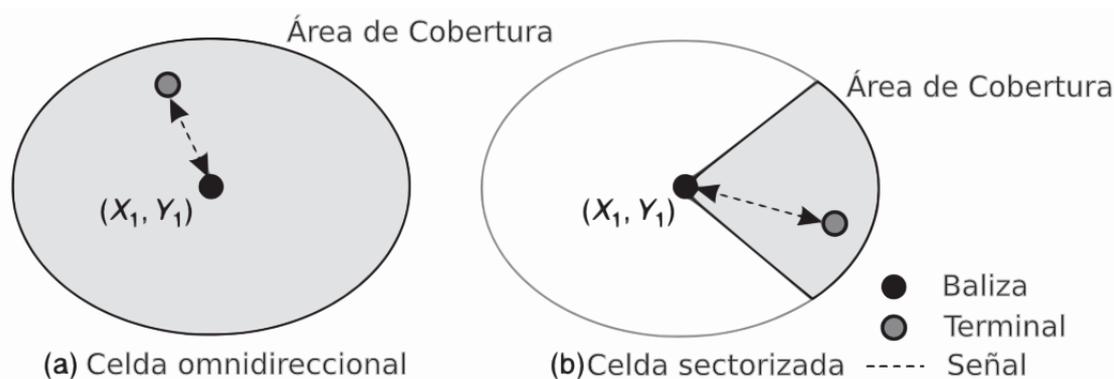
### 3.3. Técnicas y algoritmos de localización

Actualmente, para la localización en general existen varias técnicas que, aplicadas a las tecnologías mencionadas anteriormente, permiten estimar la posición. A lo largo de este capítulo se desarrollarán los principales algoritmos planteados por Küpper (2005).

#### 3.3.1. Detección por proximidad

Es el mecanismo más simple y más ampliamente usado. Se basa en los límites de cobertura de sistemas de Radiofrecuencia, Infrarrojo o Ultrasonido. Las coordenadas a determinar por el receptor se estiman a raíz de la posición conocida de los nodos emisores (una antena de red celular, por ejemplo) y la señal de cobertura que tiene el emisor. Si la antena es omnidireccional, la ubicación del receptor es la zona en un radio de la máxima cobertura de emisión. Si es una antena sectorizada, es posible acortar la ubicación a un sector del área de cobertura.

En la figura 3.1 puede verse la diferencia en cuanto al área de cobertura de ambos tipos de antenas. Este mecanismo es llamado *Celda de Origen* (*Cell of Origin - CoO*). En el caso de detectar más de un nodo, se toma como posición el nodo cuya señal tenga la mayor intensidad. La precisión de *CoO* es relativa a la densidad de los nodos emisores y a su posición.



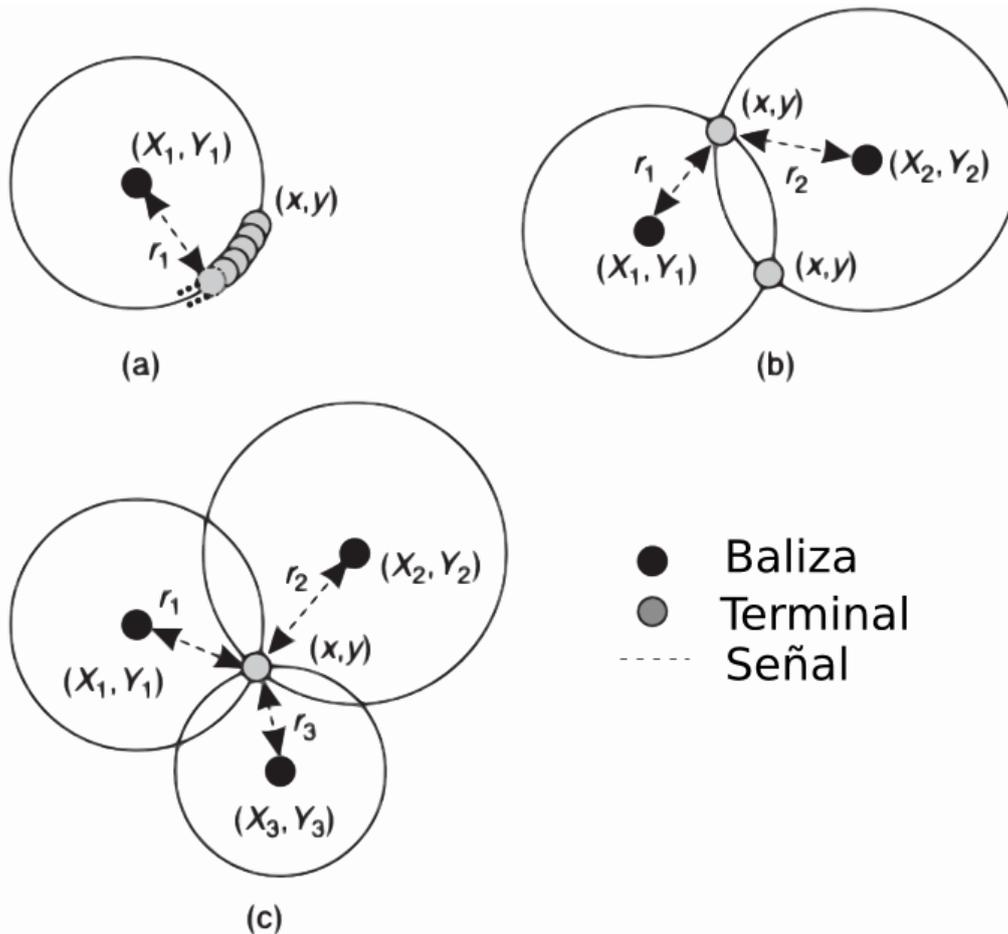
Fuente: Location-Based Services: Fundamentals and Operation - Küpper (2005)

**Figura 3.1:** Diferencias entre una antena omnidireccional y una sectorizada. El área de color gris corresponde al área de cobertura de cada tipo de antena.

#### 3.3.2. Lateración

En este caso, se debe conocer la distancia o la diferencia de distancias entre el receptor y al menos tres nodos emisores de posición fija y conocida, también llamados nodos de referencia. Estos datos son usados en una serie de  $N$  ecuaciones no lineales para calcular la posición del nodo receptor, donde  $N$  es la cantidad de los nodos emisores. Para  $N=3$  el método se llama Trilateración. Según si se tiene la distancia o diferencias de distancias, se utilizan dos tipos de lateración:

- Lateración circular:** El conocer la distancia entre el nodo receptor y el emisor (o baliza) limita la posición del receptor en un círculo alrededor del emisor, donde el radio es la distancia (figura 3.2-a). Si se agrega un nodo emisor más, es posible determinar dos posibles puntos del receptor; estas posiciones son la intersección de los círculos de ambos nodos (figura 3.2-b). Al agregar un tercer nodo emisor es posible determinar un punto en el plano y así encontrar al receptor (figura 3.2-c).

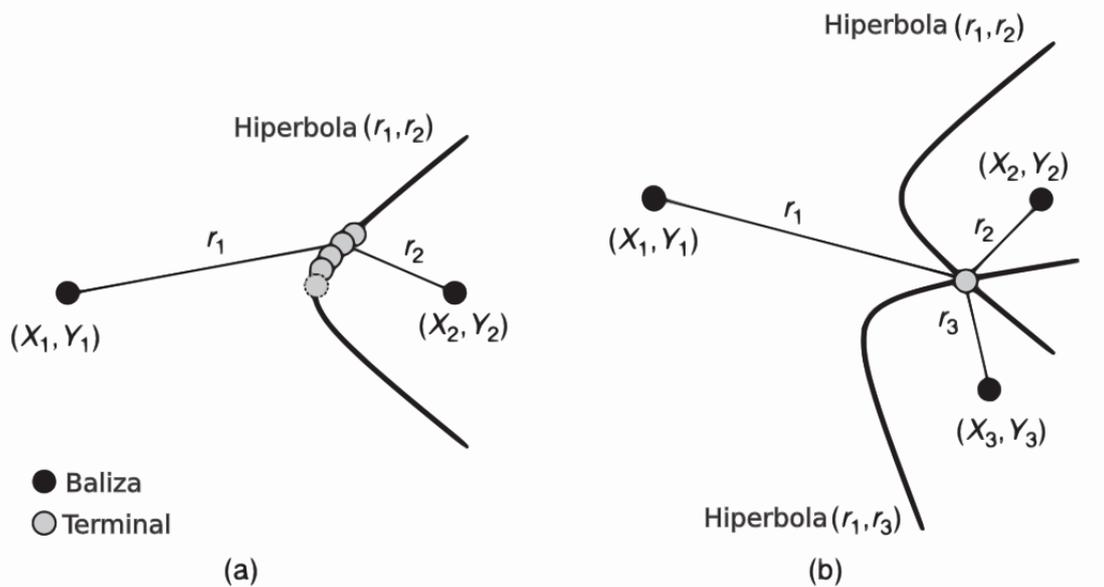


Fuente: Location-Based Services: Fundamentals and Operation - Küpper (2005)

**Figura 3.2:** Lateración circular 2D.

La aplicación de la lateración circular es posible verla combinada con medidas de tiempo; en este caso se dice que se usa el método de *Tiempo de vuelo* (sec. 3.3.2.1). El *GPS* es un claro ejemplo de la utilización de este mecanismo.

- Lateración hiperbólica:** En este caso, el receptor tiene diferencias de distancias respecto a un nodo emisor, a diferencia de la lateración circular que tiene un valor absoluto. Para el caso con un nodo emisor, el receptor se posiciona ya no en una circunferencia, sino en una hipérbola cuyo foco es la posición del nodo emisor. En la figura 3.3, al igual que con la lateración circular, se pueden ver las posibles posiciones del receptor a medida que se van agregando nodos emisores al sistema, llegando así a tres y permitiendo localizar el nodo receptor.



Fuente: Location-Based Services: Fundamentals and Operation - Küpper (2005)

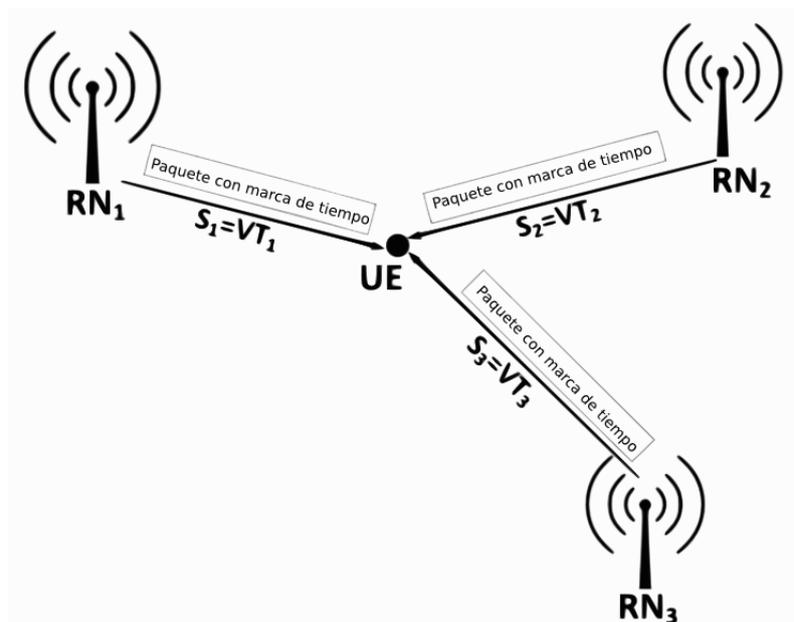
**Figura 3.3:** Lateración hiperbólica 2D.

Este tipo de lateración es aplicada en *Diferencia de tiempo de llegada* (sec. 3.3.2.2) y es la técnica que en la actualidad utilizan las redes celulares para determinar las posiciones de las terminales.

### 3.3.2.1. Tiempo de vuelo

Esta técnica se la puede encontrar con el nombre en inglés *Time of Flight (ToF)*. También se conoce como tiempo de llegada, en inglés *Time of Arrival (ToA)*. Depende fuertemente de una sincronización muy precisa del tiempo en los nodos del sistema. En este caso, cada nodo emisor envía un mensaje con la marca de tiempo de partida codificada. Luego, a partir de la diferencia entre el tiempo de llegada en el nodo receptor y la marca de tiempo en el mensaje y teniendo en cuenta la velocidad de propagación de la señal, se realiza la estimación de la distancia entre dichos nodos. Esto se puede ver en la figura 3.4. Es posible utilizar esta técnica en conjunto con la *Lateración circular* (sec. 3.3.2) para determinar la posición del receptor.

Para aplicar *ToA* es preciso tener un conocimiento del comienzo de transmisión de la señal; es por esto que los nodos deben estar sincronizados entre sí. Este método es uno de los más aplicados dentro de la localización interior, pero tiene como desventaja que su precisión puede ser afectada por el problema de *Propagación multicamino* (sec. B.3), debido a que la señal debe recorrer distancias más largas o atravesar obstáculos. La principal desventaja del método es la necesidad de una sincronización estricta entre los relojes de los nodos.

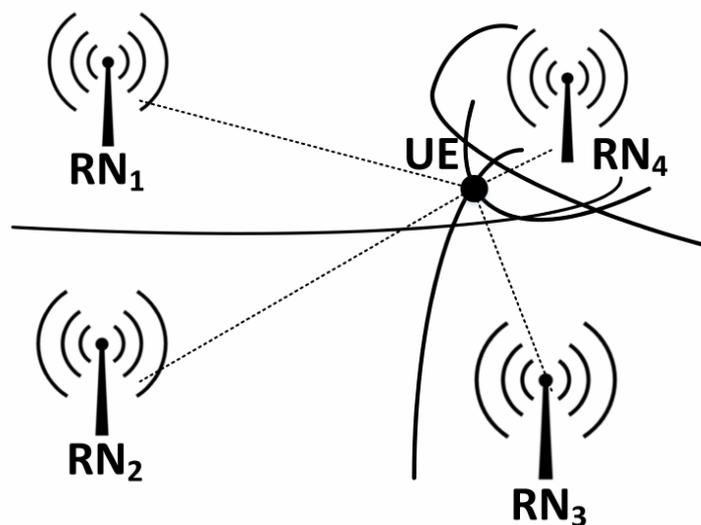


Fuente: A Survey of Indoor Localization Systems and Technologies - Zafari et al. (2017)

**Figura 3.4:** Tiempo de Llegada. Los emisores (RN) envían su marca de tiempo y el emisor (UE) calcula la distancia en función de la diferencia de tiempo. Todos los nodos deben tener sincronizados sus relojes.

### 3.3.2.2. Diferencia de tiempo de llegada

Diferencia de tiempo de llegada es conocida también como *Time Difference of Arrival (TDoA)*. En *TDoA*, se utiliza la diferencia del tiempo de propagación de las señales que viajan desde los distintos nodos emisores al receptor. La principal ventaja que tiene con respecto a *Tiempo de vuelo* (sec. 3.3.2.1) es que esta técnica no requiere que el reloj del nodo receptor esté sincronizado con el de los emisores, pero sí deben estar sincronizados entre los nodos emisores. Luego, es aplicada la *Lateración hiperbólica* (sec. 3.3.2), que permite determinar la posición del nodo receptor.



Fuente: A Survey of Indoor Localization Systems and Technologies - Zafari et al. (2017)

**Figura 3.5:** Localización basada en Tiempo de vuelo. Usando cuatro nodos emisores (RN) es posible calcular la posición del nodo receptor (UE) en un plano.

### 3.3.2.3. Tiempo de viaje de ida y vuelta

Técnica nombrada en inglés como *Round-Trip Time of Flight (RTof)*. Es conocida también como *Two-way Ranging (TWR)* o *Two-way Time of Arrival (TW-ToA)*; se basa en la medición del tiempo de propagación de la señal que viaja desde el emisor al receptor más el tiempo de propagación de la respuesta recibida por el emisor. El nodo emisor es el encargado de registrar el tiempo de partida del mensaje y el de llegada de la respuesta. Además, se apoya en la asunción de que las demoras en ambas transmisiones son iguales. Por lo tanto, un ciclo de medición consta de dos transmisiones en lugar de una, a diferencia de los esquemas *Tiempo de Vuelo* y *Diferencia de Tiempo de Llegada* descritos en las secciones anteriores. La principal ventaja de este enfoque es que no requiere ningún tipo de sincronización de relojes entre los nodos del sistema. Como contrapartida, incorpora una demora adicional que debe ser tenida en cuenta en caso de que el emisor o el receptor estén en movimiento entre que se emite el mensaje y vuelve la respuesta. Esto podría generar estimaciones incorrectas y por lo tanto afectar la precisión. También existe el problema del desconocimiento de la demora y el tiempo de procesamiento del mensaje en el receptor, antes de la emisión de la respuesta. Este inconveniente puede ser ignorado si dicha demora es significativamente menor respecto al tiempo de transmisión, como en sistemas de mediano y largo alcance, o es posible incorporar mecanismos en el protocolo que ayuden a minimizarlo.

El método explicado, también conocido como *Single-sided Two-way Ranging (SS-TWR)*, es la versión más simple y como se describió anteriormente, su funcionamiento consiste en la medición del *Round-Trip Time (RTT)*. Una extensión es *Double-sided Two-way Ranging (DS-TWR)*, en la que se realizan dos mediciones del *RTT*: una en el emisor y la otra en el receptor. Luego, estas son combinadas en una fórmula para dar el tiempo de vuelo con mayor precisión, debido a que este protocolo está diseñado para mitigar el error por desplazamiento de reloj que puede existir en los dispositivos, según se explica en Yavari y Nickerson (2014). Esta variante originalmente utiliza cuatro mensajes para la medición de los *RTTs*, pero puede optimizarse utilizando la respuesta del primer *RTT* como iniciador de medición del segundo *RTT* y así reducir el protocolo a tres mensajes (según se explica en Decawave, 2017, p.16).

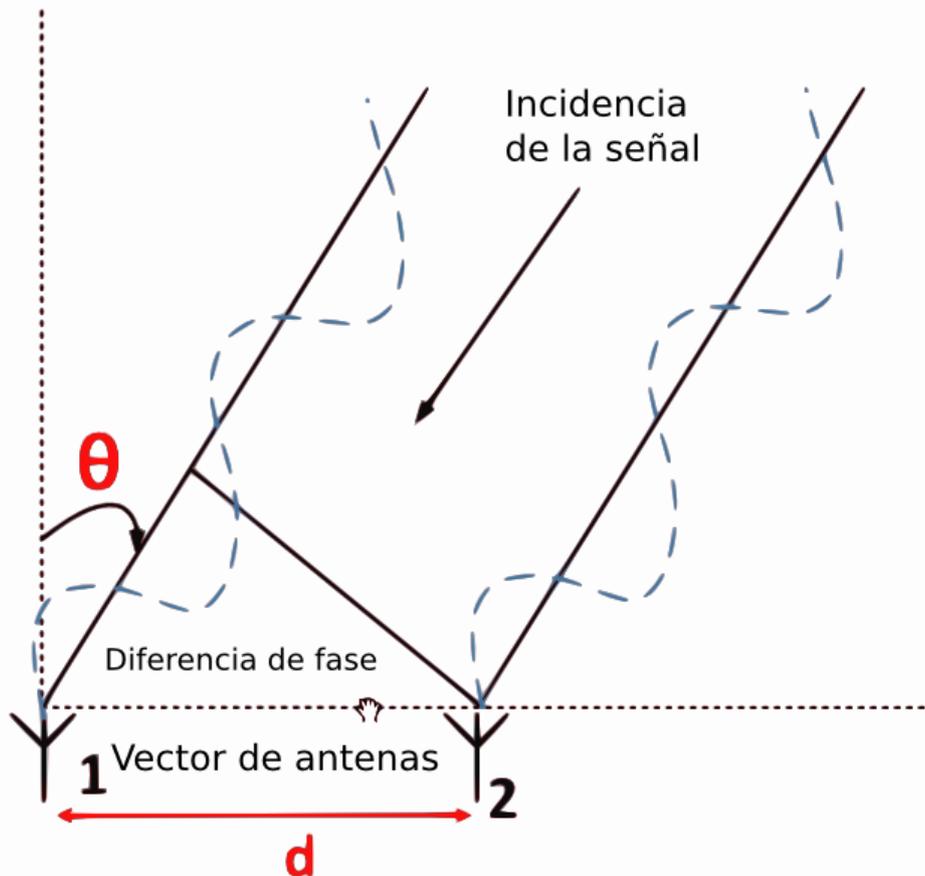
El esquema DS-TWR también tiene dos variantes: *Asymmetric Double-sided Two-way Ranging (ADS-TWR)* y *Symmetric Double-sided Two-way Ranging (SDS-TWR)*. *ADS-TWR* es llamado asimétrico porque no requiere que los tiempos de respuesta de cada dispositivo sean iguales, con la desventaja de que el cálculo de la distancia es más complejo y por lo tanto requiere mayor poder computacional. Por otro lado, en el esquema simétrico, *SDS-TWR*, el cálculo de la distancia es más simple, ya que solo requiere de operaciones de adición, sustracción y división entre un número constante y por lo tanto podría ser ejecutado fácilmente en un microcontrolador de bajo poder computacional. Su desventaja radica en la complejidad de lograr que los tiempos de respuesta sean iguales en el emisor y en el receptor. Además, debido a lo anterior, el ciclo de estimación podría demorar considerablemente más que en el esquema asimétrico.

### 3.3.2.4. Fase de llegada

También se la conoce con el nombre en inglés *Phase of Arrival (PoA)*. Este enfoque se fundamenta en el uso de la fase o diferencia de fase de la señal portadora para estimar la distancia. El supuesto para determinar la fase de la señal es que las señales transmitidas desde las balizas o desde el dispositivo son de forma sinusoidal pura y tienen la misma frecuencia y desplazamiento de fase cero. Hay un número de técnicas disponibles para determinar el rango o distancia entre el transmisor y el receptor usando *PoA*. Una de ellas es asumir que existe una demora finita de tránsito entre ellos, que puede ser expresada como una fracción de la longitud de onda de la señal. Como se puede ver en la figura 3.6, las señales arriban con diferencia de fase en cada antena del arreglo y este puede ser usado para calcular la posición del emisor.

Si la diferencia de fase entre dos señales transmitidas desde diferentes puntos de referencia puede ser usada para estimar la distancia, entonces los algoritmos basados en *Tiempo de vuelo* (sec. 3.3.2.2) pueden ser utilizados para calcular la localización.

Por otro lado, es posible usar *PoA* en conjunto con *IFSR* (sec. 3.3.2.5), *Tiempo de vuelo* (sec. 3.3.2.1) y *Diferencia de tiempo de llegada* (sec. 3.3.2.2) para incrementar la precisión y mejorar el rendimiento general del sistema. El gran problema que presenta el enfoque basado en esta técnica es que necesita estar en la línea de visión (*LoS*) (sec B.2) de los nodos de referencia para alcanzar un alto nivel de precisión, lo que podría ser difícil de lograr en ambientes interiores.



Fuente: A Survey of Indoor Localization Systems and Technologies - Zafari et al. (2017)

**Figura 3.6:** Arreglo de antenas y cómo llega la señal en cada una de ellas.

### 3.3.2.5. Basados en la fuerza de la señal recibida

También se la conoce con el nombre en inglés *Received Signal Strength (RSS)*. Este método es uno de los más simples y ampliamente utilizados en la localización interior y requiere ser implementado con tecnologías que emitan señales electromagnéticas (sec. 3.1.2.3). La fuerza de la señal es medida desde el receptor (generalmente medida en dBm o mW) y puede ser utilizada para estimar la distancia entre el transmisor y el receptor. La distancia absoluta puede ser estimada aplicando varios modelos de propagación de señal diferentes, dado que se conoce la potencia de transmisión o la potencia en un punto de referencia. Esta técnica suele utilizarse en conjunto con lateración circular para estimar una posición.

Si bien el enfoque basado en RSS es simple y rentable, sufre de una precisión deficiente, en especial en condiciones de NLoS, debido a interferencias de otras señales y a la atenuación adicional que resulta del efecto de la propagación multicamino (sec. B.3)

*Received Signal Strength Indicator (RSSI)* (que a menudo se confunde con RSS) es el indicador de RSS, una medida relativa de RSS que tiene unidades arbitrarias y la define principalmente cada proveedor de

chips. Depende fuertemente de la interferencia del ambiente, por lo que puede variar, por ejemplo, si se enciende maquinaria cerca. Por lo tanto, no es un indicador lineal (Farid et al., 2013).

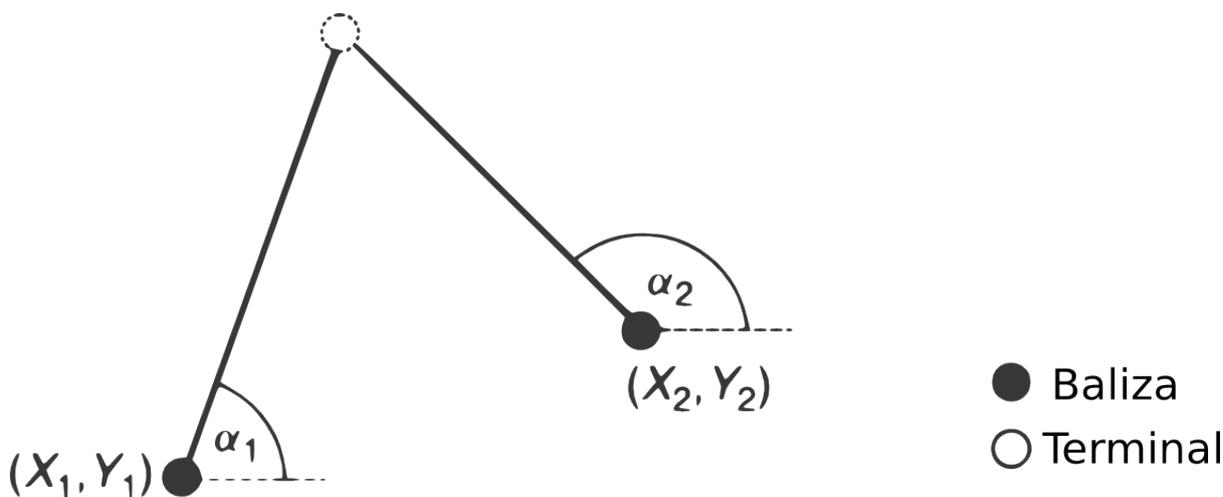
En ambos casos, por lo general, se realiza un premapeo de valores de la zona donde se pretende aplicar el sistema, para luego interpolar los valores recibidos en el receptor. En otros casos, simplemente se ajustan parámetros del mecanismo que dependen fuertemente del ambiente en donde se utiliza el sistema.

### 3.3.3. Angulación

Es conocida también como ángulo de llegada, en inglés *Angle of Arrival (AoA)*, o dirección de llegada, en inglés *Direction of Arrival (DoA)*. Esta técnica consiste en determinar la posición del nodo que se pretende localizar, calculando los ángulos de incidencia de las señales de cierta cantidad de balizas.

Para determinar este ángulo es necesario que los nodos tengan un arreglo alineado de antenas o antenas direccionales, ya sea en la terminal o en los nodos emisores, según si es la terminal o la red la que realiza el cálculo de posición (ver sec. 3.2).

El principio que hay detrás de esta técnica consiste en medir el ángulo de la señal que le llega a la terminal. Este ángulo determina una línea entre la terminal y el nodo que emite esta señal. Tomando a su vez otro nodo que emita otra señal, se tienen dos rectas que se intersectan en la terminal que quiere saber su posición. Desde el punto de vista teórico, solo eso es necesario para obtener la posición de la terminal en el plano (ver figura 3.7).

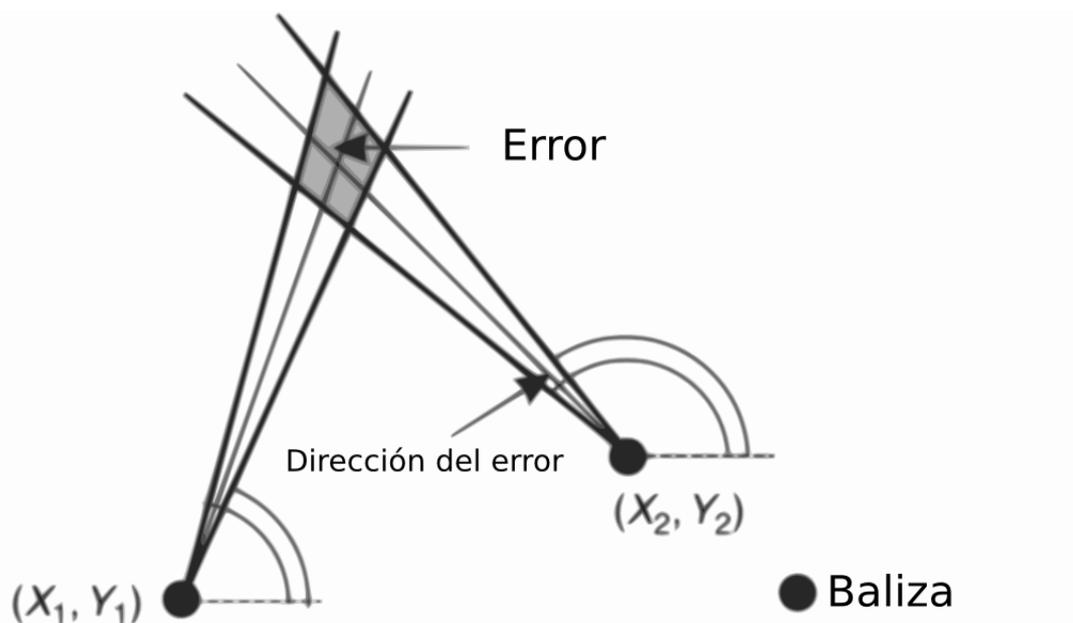


Fuente: Location-Based Services: Fundamentals and Operation - Küpper (2005)

**Figura 3.7:** Angulación.

De todas maneras, esta técnica tiene como problema que padece de mala resolución debido a la cantidad de antenas necesarias para determinar los ángulos, por lo que el ángulo medido puede ser una aproximación del valor real. Dicha aproximación puede ser más precisa si la terminal está más cercana a los nodos emisores que si está más lejana (ver figura 3.8).

Además de estos problemas, al igual que en *Lateración*, se tiene como problema principal el efecto de *Propagación multicamino* (anexo B.3) si no hay visión directa entre el nodo y la terminal (anexo B.2), por lo que el reflejo en objetos puede empeorar los problemas antes mencionados. Es por esto que Küpper (en Küpper, 2005) afirma que es recomendable hacer la *Angulación* con tres o más nodos para mejorar la precisión y compensar estos errores.



Fuente: Location-Based Services: Fundamentals and Operation - Küpper (2005)

**Figura 3.8:** Error potencial en la angulación.

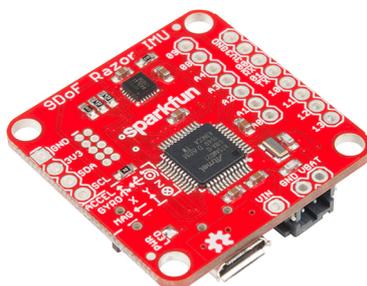
### 3.3.4. Navegación por estima

Esta técnica, también conocida como navegación inercial o en inglés como *Deduced Reckoning* (o su abreviación *Dead Reckoning*), es una de las más usadas en lo relacionado con la localización y ha sido utilizada por navegantes desde la antigüedad. Hoy en día, gracias a la mejora tecnológica, ha ganado terreno en lo que es navegación de aviones, barcos y automóviles.

La técnica consiste en determinar (o extrapolar) la posición actual en función de la última posición conocida, asumiendo que es conocida la dirección y la velocidad del movimiento o la distancia que se recorrió desde la última estimación. Para esto puede usarse una *unidad de medición inercial* (*Inertial Measurement Unit - IMU*) como la que se muestra en la figura 3.9, que mide la dirección y magnitud del desplazamiento que se hace, dado que tiene un acelerómetro, un giroscopio y un magnetómetro.

Una implementación particular de esta técnica es la *odometría*. Si el móvil tiene ruedas, es posible medir su rotación y estimar cuánto y en qué dirección estas se movieron. Otra técnica muy aplicada es la de capturar imágenes, procesarlas y estimar el movimiento del móvil: este método se llama *odometría visual*. En todos los casos, la técnica tiene un grado de error no despreciable, por lo que es necesario ajustar periódicamente para que sea confiable; por ejemplo, al tener la certeza de un punto con posición conocida, ajustar el error.

Otra consideración a tener en cuenta es acerca de la posición inicial, la dirección y la velocidad sin errores. La posición inicial o los puntos conocidos deben ser determinados por otras técnicas, como por ejemplo *GPS*, lectura de etiquetas *RFID*, entre otras, por lo que la técnica no debería ser utilizada sola; es recomendable tener un sistema que le permita obtener estos puntos conocidos y corregir el error acumulado. Con respecto a la velocidad y posición, es posible utilizar lo comentado anteriormente: *IMUs*, movimiento de ruedas, *odometría visual*, entre otras técnicas.



Fuente: <http://www.sparkfun.com>

**Figura 3.9:** IMU de 9 grados de libertad.

### 3.3.5. Reconocimiento de patrones

En el reconocimiento de patrones, también conocido como búsqueda de patrones o en inglés *Pattern Matching*, la estrategia es observar el ambiente donde se encuentra el dispositivo a localizar y estimar su posición a partir de dichas observaciones o mediciones. Se puede clasificar en patrones *ópticos* y *no ópticos*.

- En el reconocimiento de patrones *ópticos*, también llamado análisis de escena, imágenes del entorno son tomadas por una o más cámaras y son comparadas con otras previamente adquiridas. A su vez, esta clasificación es subdividida en análisis *estático* o *dinámico* (o *diferencial*) de la escena. En el análisis estático, las imágenes adquiridas son comparadas con una serie de imágenes preprocesadas del ambiente o de ciertos objetos en la escena, tomadas desde distintos ángulos y posiciones. Por otro lado, en el análisis dinámico, la localización es estimada a raíz de la diferencia entre las sucesivas capturas de imágenes del ambiente. Este análisis viene del área del proceso de imágenes, pero es muy usado en la robótica y la realidad aumentada.
- Para el caso de reconocimiento de patrones *no ópticos*, se utilizan otros valores físicos medibles del ambiente para determinar la posición, como por ejemplo la fuerza de señal (sec. 3.3.2.5) en las señales de radiofrecuencia del ambiente, que la terminal detecta en ciertas posiciones fijas del entorno. Este caso particular de *búsqueda de patrones* también es conocido como *Fingerprinting* (sec. 3.3.5.1).

El reconocimiento de patrones, por lo general, es utilizado con el esquema de terminal asistida (sec. 3.2), donde la terminal toma las mediciones de las señales del ambiente y luego un servidor es el encargado de calcular la posición, la que es comunicada a la terminal. Este enfoque no requiere modificaciones en la terminal o si las hay, son mínimas. Un ejemplo de este enfoque es el uso de la tecnología *GSM*, donde la terminal podría enviar continuamente los valores de las celdas vecinas a la red y esta devolverle la posición, la que luego puede ser utilizada para tomar decisiones (Küpper, 2005).

#### 3.3.5.1. Fingerprinting

Esta técnica fue propuesta por Kaemarungsi y Krishnamurthy (2004) y consta de dos fases. Una es denominada fase *offline*, en donde se determina una grilla en la zona de interés y luego, en cada punto de la grilla, se toman los valores de la fuerza de señal (*Basados en la fuerza de la señal recibida*, sec. 3.3.2.5) de todos los nodos emisores que son detectables en ese punto. A este vector se lo denomina *vector huella* (o *vector Fingerprint*). Luego, en la fase *online* o en tiempo real, la terminal va recabando información de las señales que percibe a medida que se mueve, las compara contra el *vector huella* generado en la fase anterior y estima la posición. La manera más común de determinar la posición es computar la *distancia*

*euclídea* (Kaemarungsi y Krishnamurthy, 2004). Según Zafari et al. (2017), existen implementaciones que utilizan redes neuronales, métodos probabilísticos, el  $k$ -ésimo vecino más cercano o algoritmos de aprendizaje supervisado.

### 3.3.6. Robotic Operating System

*Robotic Operating System*<sup>1</sup> o *ROS* es un *framework* flexible, de código abierto, que contiene un conjunto de herramientas, bibliotecas y convenciones y que busca simplificar la creación de comportamientos complejos y robustos para robots, a través de una gran variedad de plataformas robóticas. Permite abstraer el hardware usado en estas plataformas, así como el control de dispositivos de bajo nivel; tiene implementadas funcionalidades de uso común e interconecta los servicios a través del intercambio de mensajes. Su estructura está basada en grafos y el procesamiento se realiza en los nodos de estos grafos, donde es posible recibir, enviar y multiplexar mensajes de sensores, actuadores, control, estados y planificación, entre otros (Trinidad, 2020). Los nodos se comunican por medio de colas de mensajes llamadas tópicos, utilizando el modelo *publish-subscribe*, que permite una rápida y fácil construcción de sistemas computacionales distribuidos. Por esto último es que a ROS también se lo considera un *middleware* robótico.

#### 3.3.6.1. Paquete *Transform*

El paquete *Transform*<sup>2</sup> consiste en una biblioteca que permite realizar el seguimiento de múltiples marcos de referencia en todo momento. La relación entre estos marcos de referencia es organizada en una estructura de árbol almacenada por tiempo y permite transformar puntos, vectores, etcétera, entre dos marcos de referencia en cualquier instante en el tiempo. El poder de este paquete es la posibilidad de conocer en todo momento, de una manera sencilla, la posición y orientación de los distintos marcos de referencia que se utilizan.

#### 3.3.6.2. Paquete *Rosserial*

Es un protocolo para encapsular mensajes serializados en el estándar de ROS y multiplexar múltiples tópicos y servicios a través de un puerto serial o una conexión de red. Al utilizar *rosserial* se simplifica el trabajo de conectar un dispositivo con un nodo de *ROS*, ya que no es necesario implementar protocolos o sistema de mensajería para la comunicación por serial y luego un nodo que envíe la salida por un tópico.

En el proyecto se utilizó la biblioteca *rosserial\_arduino*, que permitió una fácil y rápida integración con la plataforma Arduino.

---

<sup>1</sup><http://wiki.ros.org>

<sup>2</sup><http://wiki.ros.org/tf2>

# Capítulo 4

## Trabajo realizado

Al finalizar la etapa de investigación que se muestra en el capítulo anterior y en el documento de Estado del Arte del proyecto, se tiene un panorama general, imprescindible para poder seleccionar la o las tecnologías que permitan implementar un sistema de localización interior y que mejor se adapten a la realidad planteada. Con respecto a las técnicas a utilizar, estas dependerán de las limitaciones de la tecnología. Con esto en mente, en este capítulo se presenta la tecnología elegida y el por qué de su elección respecto a las otras. Luego, se aborda el proceso de realización de las pruebas de concepto con los dispositivos adquiridos y en el cierre del capítulo se describe el funcionamiento de la solución desarrollada.

### 4.1. Características de la tecnología elegida

Se selecciona para trabajar la tecnología banda ultra-ancha (Ultra-wideband - UWB) (sec 3.1.1). Esta fue elegida en función de las características destacadas para su aplicación en la localización, tomando como base de comparación los principales problemas que presentan las otras tecnologías, como ser:

- Robustez ante el problema de NLoS (anexo B.2) que presentan otras tecnologías.
- El movimiento de los objetos o personas en el ambiente, posterior a su despliegue, no afecta su funcionamiento.
- Resuelve el problema de propagación multicamino (anexo. B.3).
- La solución que se plantea implementar no requiere sincronización de relojes.

Otra característica que se sumó a la hora de elegir UWB fue la posibilidad de encontrar trabajos externos relacionados y la existencia de dispositivos en el mercado o la posibilidad de crearlos de una manera económica y rápida. Si bien la tecnología requiere de equipamiento más costoso que otras (*WiFi*, por ejemplo), existen opciones accesibles dependiendo del fabricante.

Se encontraron proyectos *open source* que presentaban los esquemas de placas de circuitos impresos para poder fabricar los dispositivos, así como empresas que comercializaban la solución completa, esto es, los dispositivos físicos más la aplicación que realiza la localización, todo esto licenciado. Algunos de los fabricantes de estos dispositivos daban la posibilidad de modificar el código que se ejecuta dentro del dispositivo, lo que permitiría poder experimentar con distintos algoritmos y configuraciones.

Dentro de todas las opciones encontradas se optó por la disponibilizada por Localino<sup>1</sup>, por varios motivos. El motivo principal fue que daba el código fuente, con lo que se tenía la posibilidad de hacer modificaciones dentro del *firmware* que se iba a utilizar, ya que el fabricante contaba con una versión del dispositivo que comercializaba a universidades para su uso en investigación. Otro motivo importante es que el costo por dispositivo no era prohibitivo en comparación con las otras tecnologías. Otra característica relevante es que funcionaba con una biblioteca *open source* llamada *arduino-dw1000*<sup>2</sup>, compatible con Arduino, que implementa una API que resuelve la comunicación con el módulo UWB de Decawave<sup>3</sup> y que implementa también un protocolo de red y algoritmos para la estimación de la distancia entre los dispositivos *anchors* y el *tag*.

Los dispositivos de Localino, por la forma en que fueron creados para trabajar, vienen con un módulo *WiFi* integrado, que brinda una capacidad extra para el envío de información desde los dispositivos, muy importante en el proceso de implementación y pruebas. De acuerdo a los objetivos primarios del presente trabajo y previendo posibles desperfectos en los dispositivos, se entendió prudente adquirir diez unidades de placas *Localino v2*.

Características físicas del dispositivo Localino:

- Módulo UWB DWM1000 de la marca Decawave.
- Módulo ESP8266 como módulo *WiFi*.
- Microprocesador STM32, específicamente el STM32F103 de la marca ST<sup>4</sup>. Compatible con Arduino.
- Dimensiones 29 mm x 52.7 mm.
- Posibilidad de expansión por medio de pines de entrada/salida, como por ejemplo I2C, CAN, UART, SPI, etc.
- Se alimenta con 5 V desde un conector mini USB.

En la figura 4.1 se puede ver un diagrama de arquitectura de la solución original, que describe el funcionamiento implementado por Localino. Por un lado está el *tag* (dispositivo a ser localizado), que se comunica con los *anchors* (dispositivos fijos y de posición conocida) por medio de UWB utilizando la especificación IEEE 802.15.4-2011. Estos últimos reportan cada uno su distancia estimada al *tag*, a través de *WiFi*, a un programa corriendo en un computador que se encarga de realizar el cálculo de la posición del *tag*.

---

<sup>1</sup><https://www.localino.net/en/>

<sup>2</sup><https://github.com/thotro/arduino-dw1000>

<sup>3</sup><https://www.decawave.com>

<sup>4</sup><https://www.st.com>

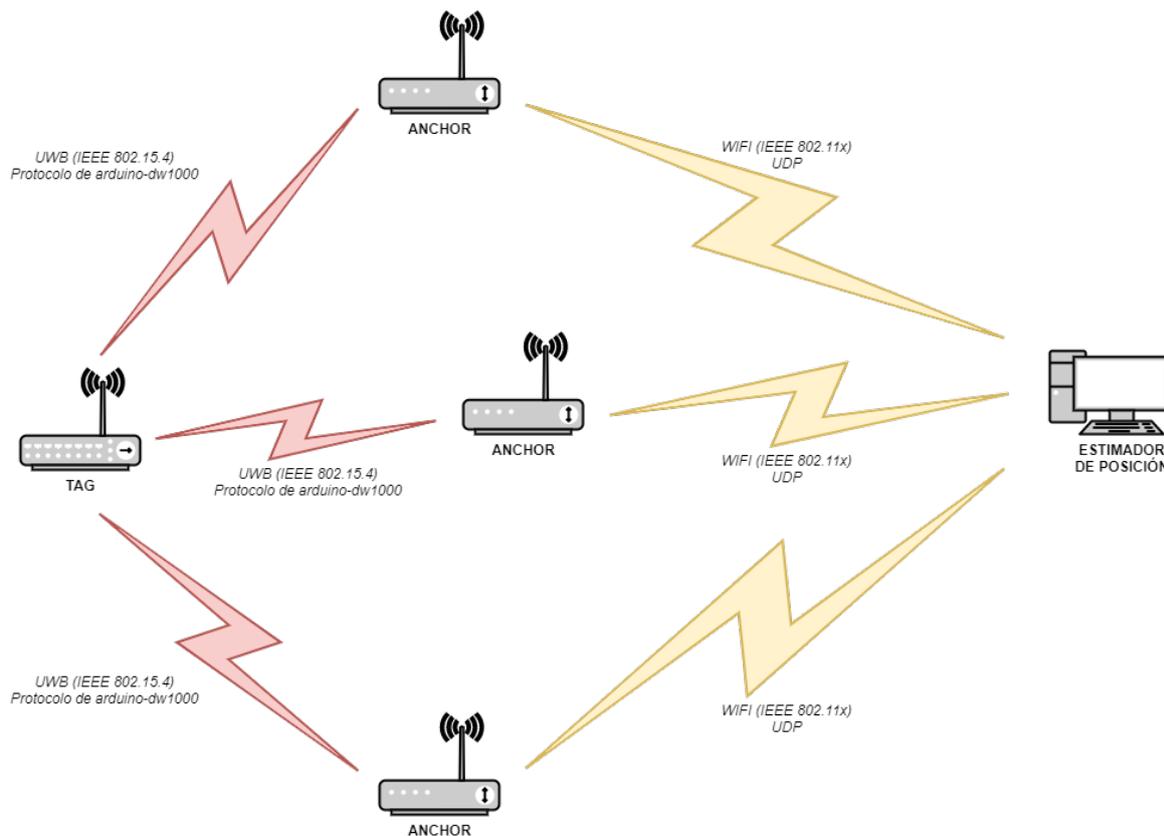


Figura 4.1: Diagrama de arquitectura utilizada por Localino.

## 4.2. Pruebas de concepto

Como parte de la solución proporcionada por el fabricante de los dispositivos, se encontró un programa que muestra en forma gráfica la posición del *tag*, en función de la información enviada por los *anchors* y de sus posiciones fijas y conocidas. Dispositivos y programa, en su conjunto, fueron utilizados para realizar la prueba de concepto y la validación del funcionamiento básico de los dispositivos, tal como fueron concebidos por el fabricante.

En un primer intento de utilizar el material de Localino, se encontraron varios problemas. Estos se exponen en el anexo C, dado que exceden el alcance de esta sección, pero es importante tratarlos, ya que provocaron demoras importantes en la realización de estas pruebas conceptuales.

Luego de resolver los problemas mencionados y otros que ocurrieron durante las pruebas del sistema original, se logró tener un funcionamiento parcial de la localización utilizando la aplicación provista. Si bien el resultado fue satisfactorio porque permitió ver el funcionamiento de todos los componentes entre sí, dicho resultado no fue riguroso, ya que los puntos donde estaban los *anchors* se colocaron de forma aproximada utilizando instrumentos de poca precisión.

La siguiente etapa, de fase de experimentación con el sistema, motivó la necesidad de contar con la posición más precisa posible de los puntos, para que los resultados tuvieran validez suficiente. Así, se pidió ayuda al Instituto de Agrimensura para posicionar veinte puntos en todo el laboratorio de robótica, además de tres puntos que sirven para el proceso de calibración. Se puede ver el proceso de medición con una estación total en las imágenes 4.2a y 4.2b.



(a) Toma de punto para calibración.



(b)

**Figura 4.2:** Proceso de relevamiento de puntos en el laboratorio de robótica. Se utilizó una estación total para este fin.

Con los puntos bien determinados fue posible realizar el proceso de calibración del parámetro *antenna delay* (ver anexo D). Este proceso surge de una serie de recomendaciones planteadas por el fabricante del módulo *DW1000*, entre las que menciona que esta calibración puede ser necesaria en función de los parámetros internos que se le configuren al módulo y también depende fuertemente del hardware en el que sea integrado dicho módulo. Por esto es que se investigó cuál es el proceso de calibración que recomienda el fabricante y se realizó un algoritmo que permita hallar de una forma automática el valor más adecuado de *antenna delay*. Para esto también fue necesario contar con dos puntos separados a una distancia específica (es parte del proceso), por lo que estos puntos también fueron parte de los relevados en la instancia que se nombró anteriormente.

El proceso de calibración consta de tener dos dispositivos separados a una distancia dada (tabulada por el fabricante, ver anexo D) y hacer una serie de estimaciones de distancia, modificando el valor del parámetro *antenna delay* hasta obtener estimaciones lo más cercanas posible a la distancia real entre ambos dispositivos. En el anexo D se puede ver en más detalle el proceso y el algoritmo usado para la calibración de este parámetro.

Dado que la solución debe tener un nodo integrado en ROS, lo siguiente fue realizar pruebas de integración de la salida resultado del *tag* por serial con un nodo ROS que permita obtener esta información

y procesarla. Para esta comunicación se utilizó *Rosserial*<sup>5</sup>, que permite generar un publicador hacia un tópico desde Arduino y levantarlo como un nodo más en ROS. Esto posibilita simplificar y evitar implementar un procesador de la salida y luego publicarla en un tópico. De esta manera queda integrado todo el sistema y es posible tener todo el flujo: desde que se ejecuta el protocolo de distancia entre los dispositivos hasta que, usando nodos de ROS, se recibe la distancia por serial, se procesa de algún modo simple y se escribe en pantalla.

### 4.3. Solución planteada

La solución planteada para resolver el problema de localización interior consiste en un sistema donde se tienen varios tipos de dispositivos: unos llamados *anchors*, cuya posición es fija y conocida, y otro llamado *tag*, ubicado en el agente que pretende localizarse en el ambiente. Estos son capaces de comunicarse utilizando la tecnología de UWB (sec. 3.1.1), a partir de un protocolo específico de la solución. El *tag* recibe la distancia desde donde se encuentra a los *anchors* que logra detectar en ese instante, por medio de una adaptación de ToA (sec. 3.3.2.1) llamada *Two-way Ranging*. La distancia y el identificador se transmiten al agente y este procesa esta información junto con la que ya posee, como ser la posición de los *anchors* y las lecturas previas, para calcular su posición aproximada en tiempo real. Esta información es transmitida a través del puerto serial del dispositivo con un protocolo propio para este fin. El procesamiento de la información es realizado por un nodo que calcula la posición utilizando la técnica de **lateración circular** (sec. 3.3.2).

En la figura 4.3 se puede ver un diagrama de arquitectura de todo el sistema. En este, se muestra la comunicación entre los dispositivos *anchors* y el *tag*. Esta es por medio del protocolo implementado en la biblioteca *arduino-dw1000* y se transmite a través de UWB. Este protocolo es el encargado del cálculo de distancias que se verá más adelante. Además, la figura muestra la comunicación entre el *tag* y el estimador de posición. Dicha comunicación se realiza por *Serial* utilizando la biblioteca *Rosserial*.

---

<sup>5</sup><http://wiki.ros.org/rosserial>

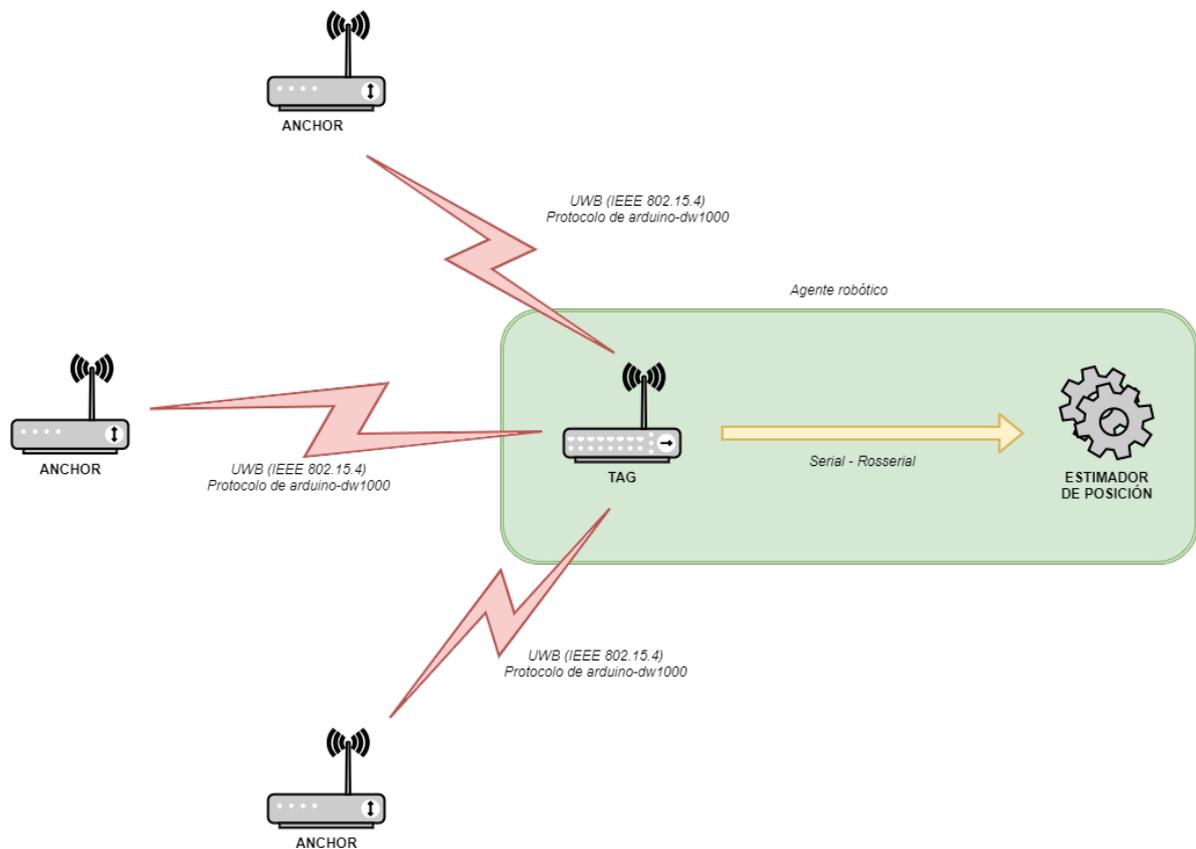


Figura 4.3: Diagrama de arquitectura planteada.

La técnica de cálculo de distancia es llamada *Two-way Ranging* y el fabricante del módulo UWB proporciona un pseudocódigo para implementarla. Esta técnica es la que se encontró implementada inicialmente en la biblioteca `arduino-dw1000`. En la imagen 4.4 se puede ver la interacción entre los distintos actores, así como el intercambio de los mensajes entre un *anchor* y el *tag* que forman parte de un ciclo de estimación. En el protocolo implementado en `arduino-dw1000` el *tag* siempre inicia la comunicación y el *anchor* se limita a responder los mensajes del *tag*. Su funcionamiento, a grandes rasgos, consiste de dos etapas: la primera corresponde al establecimiento de la conexión, que es realizada una única vez hasta que se produzca una desconexión entre dispositivos; la segunda etapa es iterativa y consiste en una serie de mensajes de intercambio entre *tag* y *anchors*, cursados en un orden específico, donde el resultado final es la estimación de distancia de cada *anchor*, que participó del ciclo de estimación, al *tag*.

En la etapa de establecimiento de conexión el *tag* envía periódicamente un mensaje específico (*BLINK*) a la dirección de difusión con su identificación única en la red. Cada *anchor* que lo recibe lo incluye en su lista de dispositivos y envía otro tipo de mensaje (*RANGING\_INIT*) con su identificación única al *tag*, que realiza el registro del remitente.

En el ciclo de estimación de distancias el *tag* comienza el proceso enviando un mensaje (*POLL*) a la dirección de difusión de la red, pero con la lista de *anchors* participantes del ciclo de estimación codificada. Esta lista corresponde a los dispositivos registrados en la red del *tag* al momento de generarse el mensaje. Además, junto a la identificación de cada *anchor* destinatario también se envía un tiempo de espera específico para cada uno, que es utilizado para coordinar el acceso al medio de transmisión en la emisión de la respuesta (*POLL\_ACK*) por parte de los *anchors*. Es así que el protocolo de acceso al medio usado en esta biblioteca implementa la técnica de tiempo ranurado. Cada *anchor* espera su

turno para enviar su respuesta y así evitar la colisión en la transmisión por el mismo canal. Luego de que el *tag* recibe la respuesta, este envía a la dirección de difusión otro mensaje con la identificación de los *anchors* que respondieron, junto a las marcas de tiempo registradas en el *tag* de los mensajes de envío y de recepción desde cada *anchor* intercambiados en la iteración actual. Estos datos son el insumo necesario para que cada *anchor* realice el cálculo de distancia al *tag*. Finalmente, los *anchors* destinatarios del último mensaje calculan la distancia y la reportan (*RANGE\_REPORT*) al *tag*. Como se puede ver, la estrategia consistió en utilizar un mensaje más en cada iteración, con el fin distribuir el costo computacional requerido por el cálculo de distancia, que en este caso se delega a cada uno de los *anchors* en lugar de estar centralizado en el *tag*.

La ecuación 4.1 calcula el tiempo de propagación. Al multiplicarlo por la velocidad en que se propagan las ondas se obtiene la distancia entre *anchor* y *tag* (fórmula 4.2).

$$T_{prop} = \frac{(T_{round1} \cdot T_{round2} - T_{reply2} \cdot T_{reply1})}{(T_{round1} + T_{round2} + T_{reply2} + T_{reply1})} \quad (4.1)$$

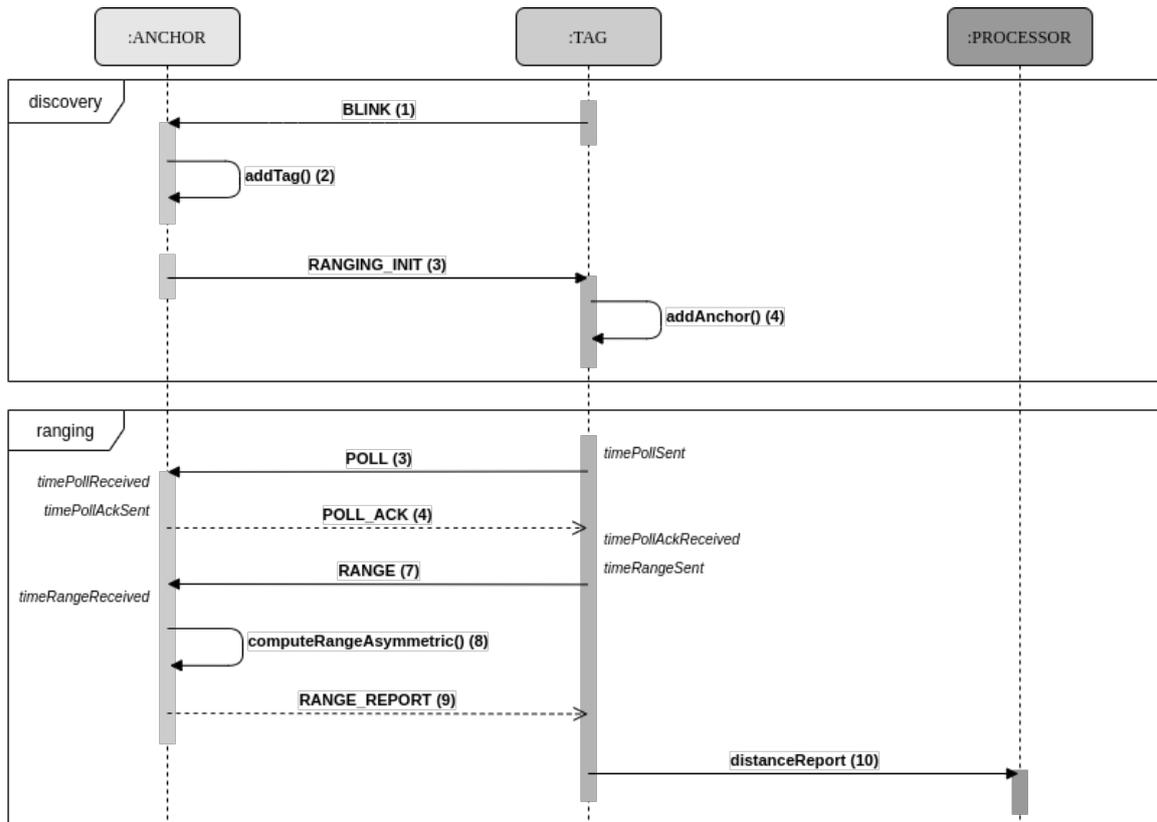
$$D = T_{prop} \cdot V_{uwb} \quad (4.2)$$

Donde:

$$\begin{aligned} T_{round1} &= TS_{POLL\_ACK}^{received} - TS_{POLL}^{sent} \\ T_{reply1} &= TS_{POLL\_ACK}^{sent} - TS_{POLL}^{received} \\ T_{round2} &= TS_{RANGE}^{received} - TS_{POLL\_ACK}^{sent} \\ T_{reply2} &= TS_{RANGE}^{sent} - TS_{POLL\_ACK}^{received} \end{aligned}$$

- $TS_M^{received}$  es la marca de tiempo tomada al momento de recibir el mensaje  $M$ .
- $TS_M^{sent}$  es la marca de tiempo tomada al momento de enviar el mensaje  $M$ .

En la figura 4.5 se pueden ver representados gráficamente los tiempos descritos anteriormente. Cabe destacar que en el cálculo de la distancia se usa como  $V_{uwb}$  la velocidad de la luz ( $C$ ) como aproximación. Dentro de la biblioteca encargada de hacer este cálculo, se utiliza una constante nombrada como *DISTANCE\_OF\_RADIO* en ese producto, la que parece ser una medida de longitud. Esto no es correcto dimensionalmente, pero no pudo ser fundamentado por el equipo por medio de la documentación de la biblioteca o por trabajos relacionados que utilizan la biblioteca.



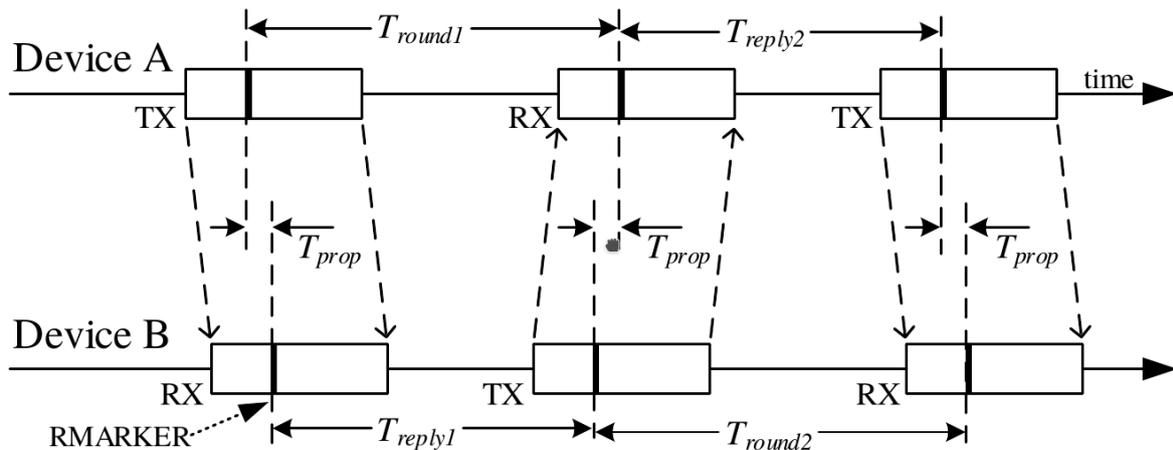
```

(1) BLINK(tagAddress, tagShortAddress)
(2) addTag(tagAddress, tagShortAddress)
(3) RANGING_INIT(tagShortAddress, anchorAddress, anchorShortAddress)
(4) addAnchor(anchorAddress, anchorShortAddress)
(5) POLL(BROADCAST_SHORT_ADDRESS, List<ReplyTDM> listReplyTDM)
ReplyTDM{
    anchorShortAddress,
    replyTime
}
(6) POLL_ACK(tagShortAddress)
(7) RANGE(BROADCAST_SHORT_ADDRESS, List<Ranging> listRanging)
Ranging{
    anchorShortAddress,
    timePollSent,
    timePollAckReceived,
    timeRangeSent
}
(8) computeRangeAsymmetric() {
    round1 = timePollAckReceived - timePollSent
    reply1 = timePollAckSent - timePollReceived
    round2 = timeRangeReceived - timePollAckSent
    reply2 = timeRangeSent - timePollAckReceived

    timestamp = (round1 * round2 - reply1 * reply2) / (round1 + round2 + reply1 + reply2);
    range = timestamp * DISTANCE_OF_RADIO
}
DISTANCE_OF_RADIO = 0.0046917639786159f
(9) RANGE_REPORT(tagShortAddress, range, rxPower)
(10) distanceReport(anchorID, distance, rxPower, errCode, errMessage)

```

Figura 4.4: Flujo de comunicación entre tag, anchor y agente robótico; debajo, el pseudocódigo de los métodos. Se puede ver cómo se computa la distancia en el método *computeRangeAsymmetric*.



Fuente: DW1000 User Manual - Decawave (2017)

**Figura 4.5:** Representación gráfica de la técnica *Two-way Ranging* entre los dispositivos *A* y *B*. Es posible ver los tiempos que se tienen en cuenta en la ecuación 4.1. *RMARKER* deriva de *Ranging Marker* y es parte del estándar definido en IEEE802.15.4-2011. Representa el comienzo de la transmisión o recepción del mensaje y es utilizado para obtener las marcas de tiempo.

En párrafos anteriores se habló de que se utilizaba una adaptación de RToF (sec. 3.3.2.3). Esta adaptación está implementada en la biblioteca `arduino-dw1000` utilizada para la comunicación entre dispositivos UWB. Cabe mencionar que se realizaron modificaciones en dicha biblioteca, dado que llevaba tiempo sin mantenimiento, y se solucionaron algunos errores que se encontraron en su protocolo.

La comunicación entre el *tag* y el estimador de posición se realiza por serial utilizando `rosserial`, lo que permite aprovechar el uso de tópicos y resuelve la comunicación entre ambos componentes. A su vez, da la posibilidad de definir un mensaje propio que es enviado desde el *tag* al recibir la estimación de distancia del *anchor*. Posteriormente, el estimador de posición, que es un nodo en ROS, obtiene el mensaje del tópico al que envía el *tag*, lo procesa junto con la información existente de la posición de los *anchors* y los mensajes previos y obtiene así la posición del agente. El procesamiento de los datos se realiza con una biblioteca que utiliza las posiciones conocidas de los *anchors* junto con sus distancias y por medio de lateración circular estima la posición. El flujo del proceso de calcular la posición se puede ver en la figura 4.6.

La posición luego es enviada al tópico llamado `tf2` del paquete `Transform` (sec. 3.3.6.1), que posee la información de la posición relativa del componente físico, en este caso el *tag*, con respecto al marco de referencia que lo contiene. Como en este caso el marco de referencia que lo contiene es el entorno (o ambiente), se puede tomar la posición calculada como posición absoluta de dicho componente.

El mensaje usado entre el *tag* y el estimador de posición tiene los siguientes campos:

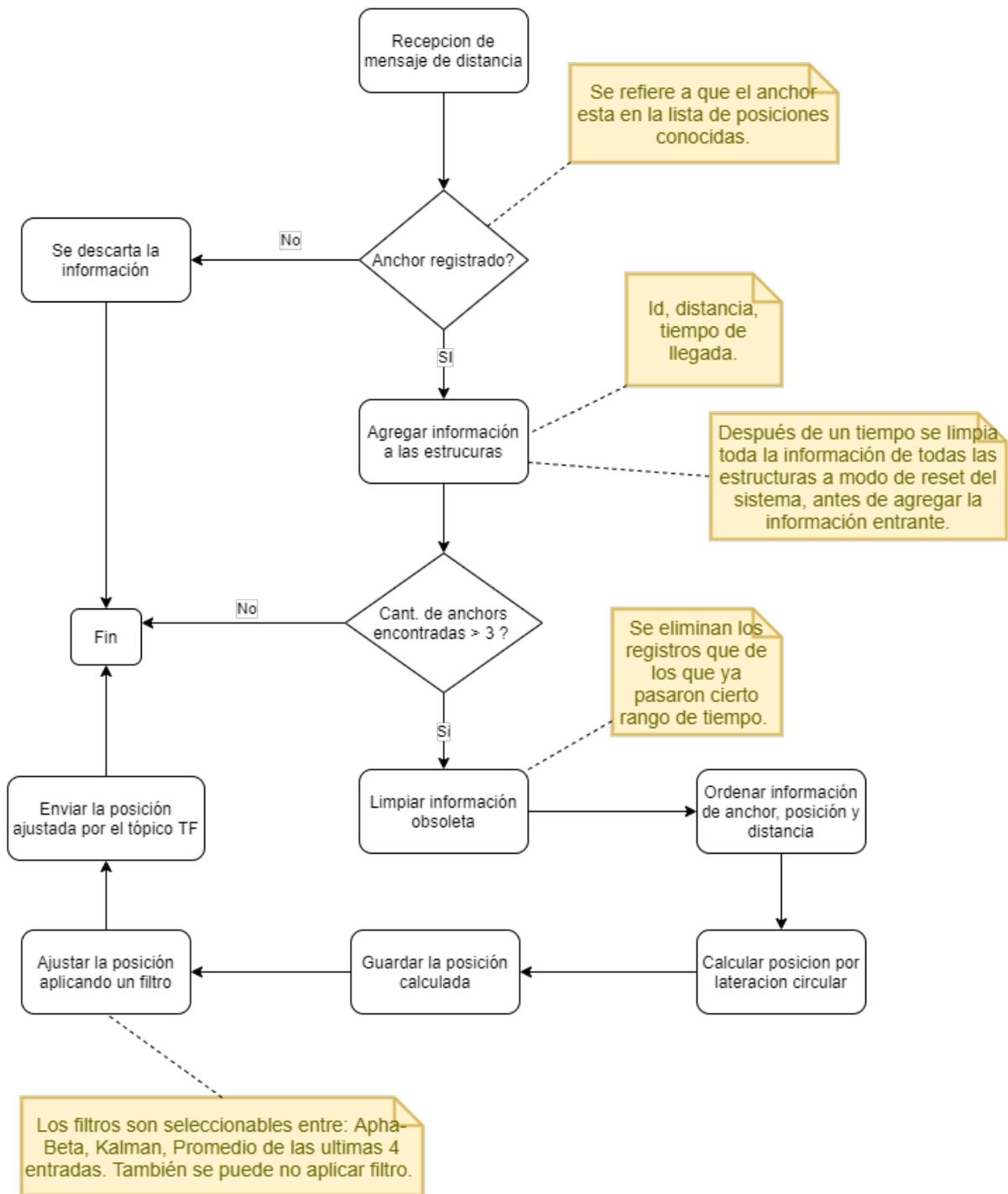
- Código de error.
- Mensaje de error si lo hay.
- Identificador del *anchor*.
- Distancia calculada.
- Ganancia de recepción.

- Marca de tiempo interna al *tag*.

Es posible ver el uso de este mensaje en el diagrama de la figura 4.4, en el método llamado *distance-Report*.

En cuanto al *tag*, su ubicación dentro del agente robótico debe ser conocida por este, ya que es necesaria para utilizar las transformaciones que implica el uso del paquete *Transform* de ROS (sec. 3.3.6.1).

A la hora de ubicar estos dispositivos en el lugar que les corresponda, es importante considerar que tanto *anchors* como *tag* deben estar en una posición lo más vertical posible, para evitar generar más errores (sec. 5.4).



**Figura 4.6:** Flujo del proceso de estimación de la posición del nodo estimador de posición. Este se dispara al llegar un mensaje de estimación de distancias desde el *tag*.

## Capítulo 5

# Experimentación

Al tener el sistema funcional, como se explica en el capítulo anterior, lo siguiente es medir la calidad de la solución. Para ello, es necesario realizar pruebas para determinar qué tan bueno es como sistema de localización y analizar si los dispositivos se comportan como afirma el fabricante. Por tal motivo, en esta sección se presentan varios experimentos realizados.

Los datos requeridos para poder realizar las pruebas se tomaron en un procedimiento previo, que consistió en grabar 2000 mensajes con la información de estimación de distancia (ver sec. 4.3) entre el *tag* y un *anchor*, enviados al estimador de posición en ROS. Ambos dispositivos fueron colocados en puntos de posiciones conocidas, según se muestra en la figura 5.1. Este procedimiento se repitió moviendo el *anchor* a distintas posiciones, de acuerdo a la planificación de las pruebas. De esta manera, se obtuvieron  $N$  conjuntos de datos, donde cada uno corresponde a la interacción del *tag* con un solo *anchor*. Luego, para cada experimento, se fueron combinando los mensajes de los distintos conjuntos de datos originados en el proceso anterior de forma sincronizada para crear así escenarios específicos de forma artificial.

El contar con los conjuntos de datos antes de realizar los experimentos permitió, por un lado, replicar estos experimentos las veces necesarias sin tener que realizarlos en el laboratorio y, por otro lado, no tener límite en la cantidad de dispositivos funcionando al mismo tiempo. En cuanto a esto último, se encontró una restricción en la cantidad de *anchors* con los que el *tag* es capaz de comunicarse en simultáneo. En la sección 7.1 se analiza en profundidad este problema y su posible solución. Cabe destacar, como desventaja, que al usar los escenarios de forma artificial las pruebas son estáticas; no es posible hacer pruebas con el *tag* en movimiento.

Para la grabación de los mensajes y su posterior reproducción se utilizó la herramienta de ROS llamada *rosvbag*<sup>1</sup>, que permite guardar en un archivo (o *bag*) los mensajes recibidos en un tópico específico y luego volver a publicarlos en el mismo tópico, conservando los tiempos como si se estuviese ejecutando en tiempo real.

Para simplificar el uso de los escenarios artificiales se utilizó un entorno de virtualización basado en contenedores, más concretamente Docker<sup>2</sup>, en conjunto con la herramienta Compose<sup>3</sup>. Además de facilitar la ejecución de dichos escenarios, esto permitiría su ejecución incluso en ambientes que no cuenten con ROS instalado o en sistemas operativos no soportados por la distribución de ROS utilizada en este proyecto.

---

<sup>1</sup><http://wiki.ros.org/rosvbag>

<sup>2</sup><https://www.docker.com>

<sup>3</sup><https://docs.docker.com/compose/>

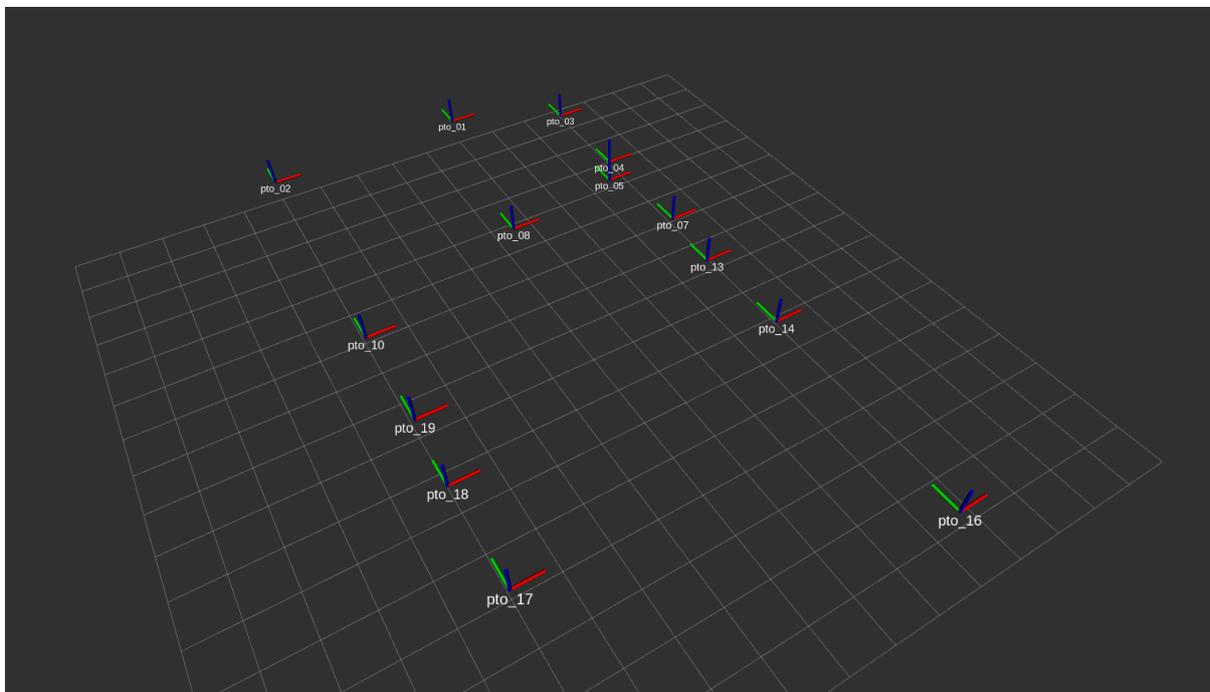
En la figura 5.2 se puede ver cómo se colocaron los *anchors* y el *tag* en los distintos puntos previamente posicionados. Estos puntos fueron seleccionados de manera tal que cubrieran todo el laboratorio y, a su vez, que varios tuvieran alguna particularidad más acorde a la realidad: por ejemplo, algunos posicionados sobre una columna de metal, otros con poca línea de visión hacia el *tag*, en el techo, entre otros.

Para facilitar la colocación de los dispositivos en dichos puntos o lo más cercano posible a ellos se diseñaron cajas para ser impresas en 3D, como se puede ver en la figura 5.2a.

En la figura 5.1 se muestran los puntos usados para experimentación en el laboratorio. Del total de 20 puntos relevados al comienzo del proyecto, se utilizaron los 13 que se listan en la tabla 5.1. Todos oficiarán como posiciones para *anchors*, a excepción del punto ocho que será usado por el *tag*.

Id. de anchors	Coord. X (m)	Coord. Y (m)	Coord. Z (m)	Observaciones
1	0.869	9.077	1.649	
2	-3.692	8.882	1.624	
3	3.291	7.809	1.691	
4	3.274	5.404	1.732	
5	3.276	5.405	1.207	
7	3.295	2.778	1.803	
8	0.091	4.091	1.725	Cerca de un caño metálico de la infraestructura eléctrica. Utilizado para el <i>tag</i> , se encuentra a 20 cm del techo.
10	-3.688	2.904	1.749	
13	3.282	1.457	1.594	Detrás de columna de hormigón, respecto a la línea de visión del <i>tag</i> .
14	3.28	-0.577	1.731	Entre dos tablonces de madera prensada de medidas no registradas.
16	3.083	-5.714	1.709	Detrás de estantería, respecto a la línea de visión del <i>tag</i> .
17	-3.697	-3.163	1.725	
18	-3.701	-0.824	1.493	
19	-3.696	0.378	1.983	Detrás de viga de hormigón, respecto a línea de visión del <i>tag</i> (fig. 5.2d).

**Tabla 5.1:** Puntos utilizados en los experimentos.



**Figura 5.1:** Diagrama de posiciones de los puntos en el laboratorio utilizado en los experimentos. La grilla muestra una resolución de 1 m de lado.



(a) Dispositivo en el marco de una ventana. La mayor parte de los puntos están en una disposición parecida, con línea de visión entre este dispositivo y el *tag*.



(b) En la imagen se puede ver resaltado el dispositivo sobre una columna de hormigón en una esquina.



(c) Para la experimentación en el laboratorio, se dejó fijo el *tag*. En esta imagen se puede ver dónde se colocó: se encuentra alejado unos centímetros de la viga del techo para que esta no genere ruido a la hora de correr los experimentos.



(d) En esta imagen se puede ver el *anchor* resaltado (en color) en una pared lateral del laboratorio.

**Figura 5.2:** Varios lugares en los que se colocaron los dispositivos para los distintos experimentos. En algunas de las imágenes se resalta el dispositivo (en color) dentro de la imagen general (en blanco y negro).

## 5.1. Efecto del incremento de dispositivos *anchors* en la precisión del sistema

Este experimento se realizó de forma artificial, de acuerdo a lo indicado al inicio del capítulo, utilizando los conjuntos de datos de distancia entre el *tag* en una posición fija y los *anchors* en el resto de los puntos del laboratorio. El objetivo del experimento es investigar cómo se comporta el sistema a medida que se agregan *anchors* en el ambiente.

Para evaluar qué tan preciso es el sistema, se puede determinar el error entre la posición actual estimada y la real. Este error se puede medir calculando la *distancia euclídea* entre estos dos puntos: a menor distancia, menor es el error y por lo tanto más preciso es el sistema.

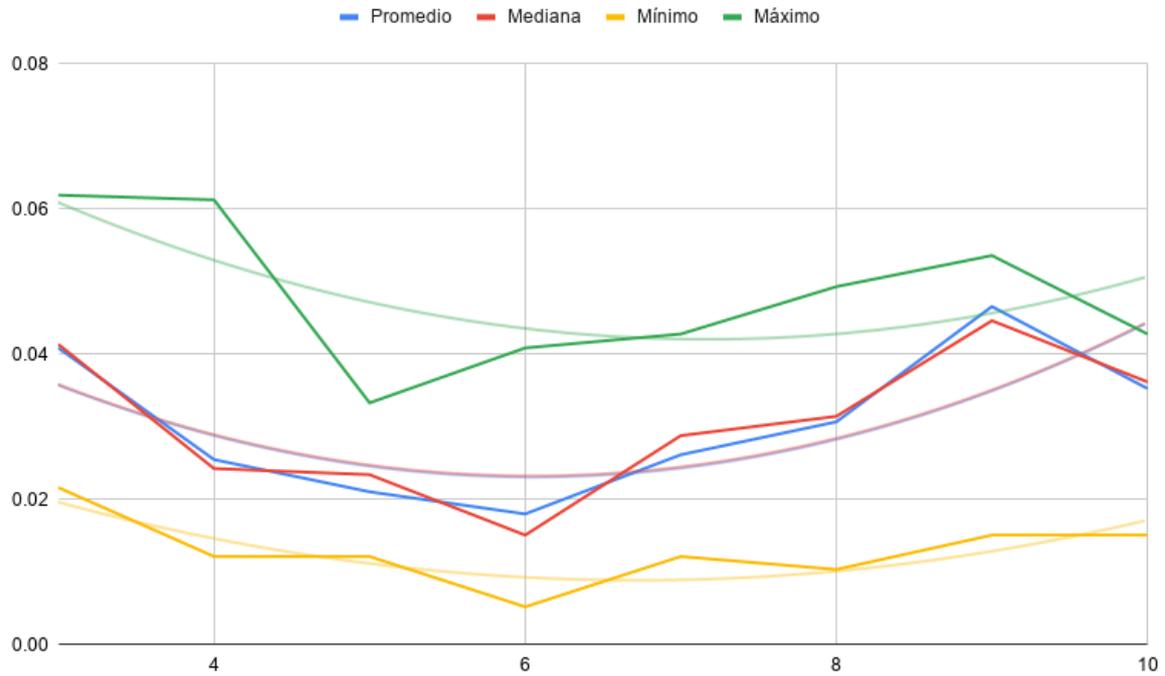
En primer lugar, el experimento consistió en levantar en simultáneo un número fijo de conjuntos de datos de distancia comenzando con tres puntos distintos (el mínimo para poder obtener la posición), procesar los datos en el estimador de posición y obtener las coordenadas estimadas. Luego, a partir de la obtención de la posición 1000 veces se calculó el error de cada una de ellas, como se comentó anteriormente, y a partir de este conjunto de errores se hallaron los valores de error promedio, mediana del error, error mínimo, error máximo y la desviación estándar. Una vez concluido el experimento, este volvió a realizarse agregando un nuevo conjunto de datos correspondiente a un nuevo punto y así sucesivamente hasta llegar a tener diez *anchors* corriendo simultáneamente.

La tabla 5.2 muestra los resultados numéricos de este experimento: en estos datos es posible ver que la mediana y el promedio del error se comportan de manera muy similar en cada escenario. Si se tiene en cuenta la columna de error promedio, es posible ver que los valores van disminuyendo hasta tener seis *anchors* y luego vuelven a subir, por lo que se puede afirmar que en seis se da un mínimo. Si el análisis se efectúa sobre la columna de desviación estándar, se ve un comportamiento similar. En este caso, el mínimo se encuentra en cinco *anchors*. Revisando ambos casos en conjunto, es posible ver que si bien el mínimo se da en seis en la columna promedio, el mínimo en la columna de desviación estándar es igualmente válido, ya que indica que los valores de los datos de la muestra están cerca del promedio para cinco *anchors* y este es muy cercano al promedio para seis *anchors*. Con estos datos, se podría concluir que la cantidad ideal de *anchors* con los que se logra una estimación de distancia con el menor error se encuentra entre cuatro y seis.

Cant. anchors	Promedio	Mediana	Mínimo	Máximo	Desviación estándar
3	0.040	0.041	0.021	0.061	0.0082
4	0.025	0.024	0.012	0.061	0.0052
5	0.021	0.023	0.012	0.033	0.0048
6	0.017	0.015	0.005	0.040	0.0083
7	0.026	0.028	0.012	0.042	0.0085
8	0.030	0.031	0.010	0.049	0.0069
9	0.046	0.044	0.015	0.053	0.0059
10	0.035	0.036	0.015	0.042	0.0049

**Tabla 5.2:** Valores del error promedio, mediana, máximo y mínimo en metros y la desviación estándar a medida que se van agregando *anchors* al sistema.

Al graficar la tabla se obtiene la figura 5.3, en la que se puede ver que existe un punto de inflexión al llegar a seis *anchors* y esto coincide con el razonamiento planteado en el párrafo anterior. Dicho punto de inflexión deja claro que a medida que se agregan dispositivos mejora la precisión, aunque hasta cierto grado, ya que luego se torna contraproducente. Por lo tanto, en el algoritmo de estimación, se debería implementar una selección dinámica que elija por medio de algún criterio hasta seis *anchors* para obtener la localización, con el fin de que el exceso de estos no la empeore.



**Figura 5.3:** Datos graficados de la tabla 5.2, que muestran cómo se mueven los valores para error promedio, mediana del error, error mínimo y error máximo a medida que se aumenta la cantidad de *anchors*. Las líneas curvas son líneas de tendencia de cada serie; se puede ver que los mínimos del promedio y la mediana se dan entre 4 y 6 *anchors*.

## 5.2. Errores en la estimación de distancia entre dispositivos

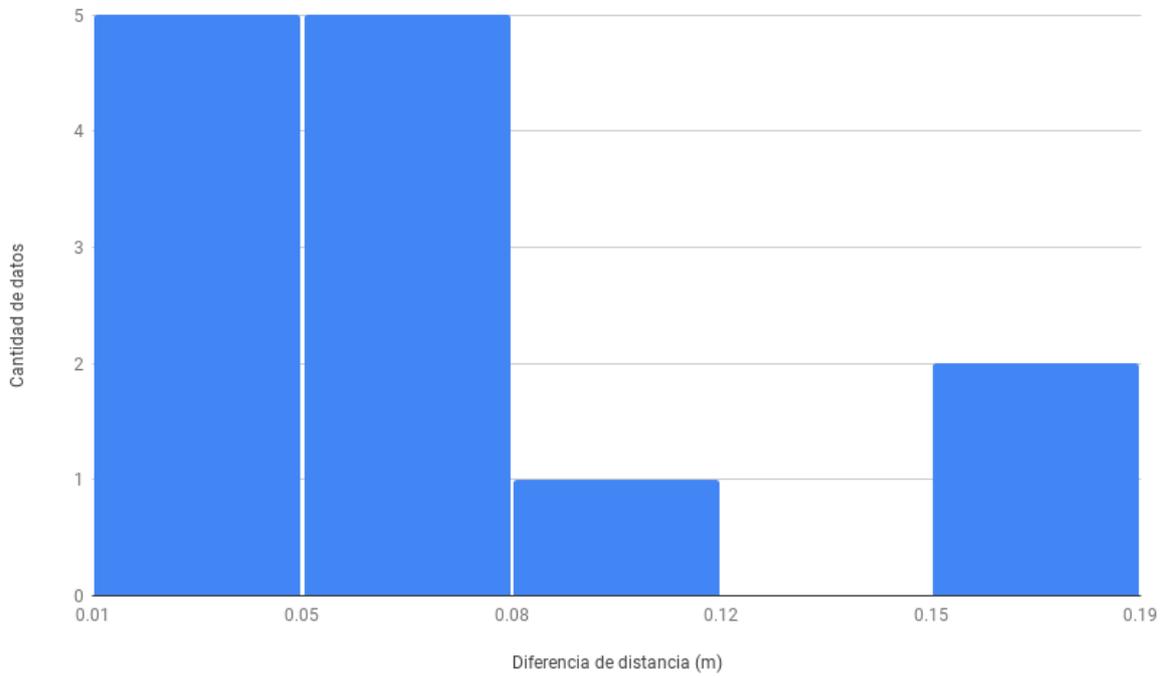
La estimación de la distancia realizada por los dispositivos no es precisa, por lo que introduce una fuente de error en el cálculo de la posición (sec. 5.1). Por esta razón, el siguiente experimento se centra en encontrar las diferencias entre la distancia calculada por la ejecución del protocolo entre los dispositivos y la distancia real. Esta última es calculada a partir de la posición de los puntos reales en el laboratorio.

El experimento se ejecutó de forma artificial, utilizando los conjuntos de datos de distancia entre el *tag* y el *anchor* colocado en cada punto que aparece en la figura 5.1. Por cada conjunto de datos, el error entre la distancia real y la distancia obtenida se halló 1000 veces. Con estos valores también se calculó la raíz cuadrada de la media, la que permite medir el error entre el conjunto de datos estimados y el valor real. Este experimento se realizó para 13 puntos dentro del laboratorio, donde se colocaron *anchors* según la tabla 5.1.

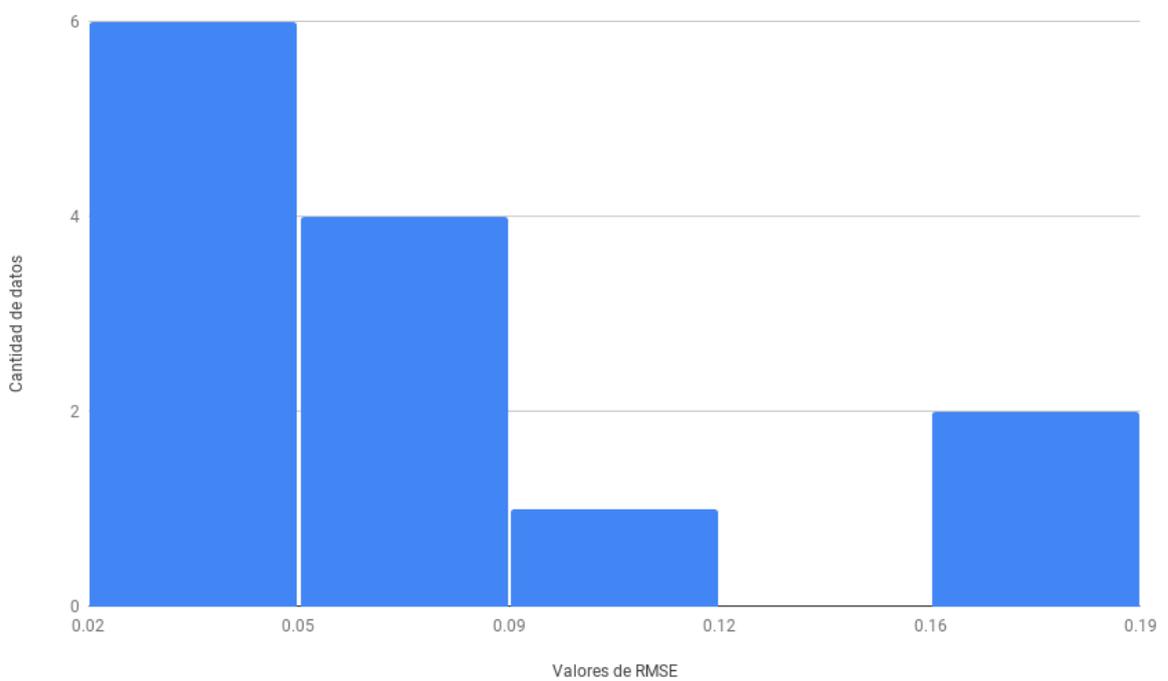
Se pueden ver los datos obtenidos en la tabla 5.3 y los histogramas de estos datos en las figuras 5.4 y 5.5. Como es posible observar, la mayoría de los datos están en el rango de error menor a los diez centímetros, lo que es aceptable y esta el rango de error que asegura el fabricante del módulo *DW1000*. Los datos de mayor rango de error corresponden a dos puntos que se encuentran sin línea de visión hacia el *tag*; estos son los de identificador 19 y 14 (ver tabla 5.1). Con la raíz cuadrada de la media (o RMSE) es posible ver que la mayoría de las muestras tomadas presenta bajo error. De este análisis se puede sacar en claro que el cálculo de distancias no genera un error mayor al que anuncia el fabricante bajo condición de línea de visión entre los dispositivos.

Id. anchor	Distancia real	Mediana del error	Error	Raíz del error cuadrático medio
5	3.484	3.498	0.014	0.021
10	3.961	3.980	0.019	0.022
1	5.046	5.083	0.036	0.037
16	10.251	10.288	0.037	0.038
17	8.183	8.224	0.041	0.042
13	4.139	4.091	0.048	0.052
2	6.10	6.163	0.058	0.059
18	6.212	6.272	0.060	0.061
7	3.463	3.388	0.075	0.076
4	3.443	3.522	0.078	0.079
3	4.905	4.808	0.096	0.097
14	5.653	5.841	0.188	0.188
19	5.309	5.120	0.188	0.189

**Tabla 5.3:** Resultados de la ejecución del experimento. Las filas están ordenadas por la columna error.



**Figura 5.4:** Histograma que muestra el error en el cálculo de distancias por parte de los dispositivos.



**Figura 5.5:** Histograma que muestra la raíz cuadrada de la media; esta representa la variación del error en la muestra.

### 5.3. Efecto de los obstáculos en la precisión del sistema

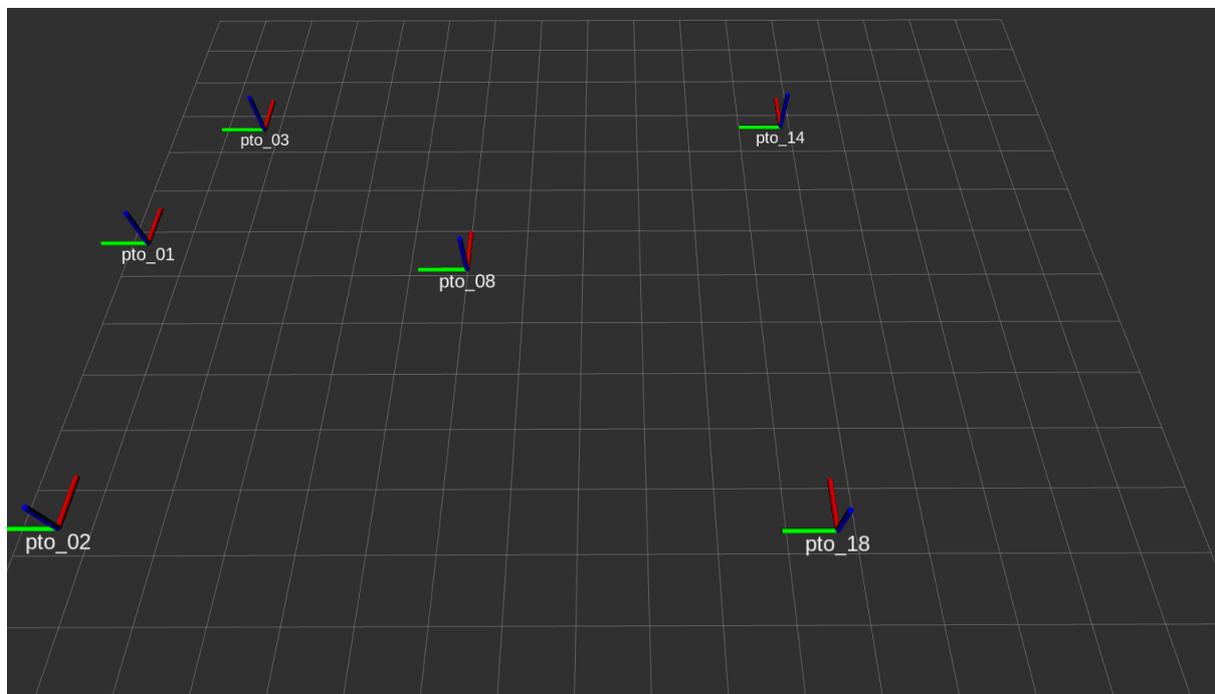
Otro aspecto positivo e importante que presenta la tecnología, en comparación con otras basadas en radiofrecuencia, es la posibilidad de utilizar bajas frecuencias, donde las señales pueden atravesar obstáculos más fácilmente. Para probar esta característica se diseñó el siguiente experimento, que tiene como objetivo medir el error al estimar la posición, utilizando cinco *anchors* fijos. Combinando algunos con línea de visión y otros con obstáculos (Non Line of Sight, Anexo B.2), para luego comparar los resultados.

Se comenzó con los cinco *anchors* con línea de visión y en cada iteración de toma de datos posterior se quitó la línea de visión de un nodo al agregarle un obstáculo. Los nodos sin línea de visión se fueron seleccionando en forma aleatoria hasta completar todos los *anchors* sin línea de visión.

A la hora de elegir el obstáculo a utilizar en este experimento, se buscó que tuviera características físicas similares a elementos que se pueden encontrar en un invernadero (realidad planteada en sec. 1). Así, se seleccionó un tablón madera prensada (de 2 metros de largo, 60 centímetros de ancho y 18 milímetros de espesor), que fue colocado a 2 metros de distancia de los *anchors*, en la línea de visión entre este y el *tag*. A su vez, los *anchors* fueron ubicados en cinco puntos seleccionados del laboratorio; la figura 5.6 muestra su disposición en el espacio.

Los datos trabajados en el experimento se tomaron a partir de un muestreo de mil estimaciones; este fue realizado a través de la simulación del funcionamiento en paralelo de los *anchors* utilizando *bags* de ROS.

Al igual que en otras pruebas, como medida de error se utilizó la *distancia euclídea* entre el punto estimado y el real, para realizar la comparación entre los distintos escenarios.



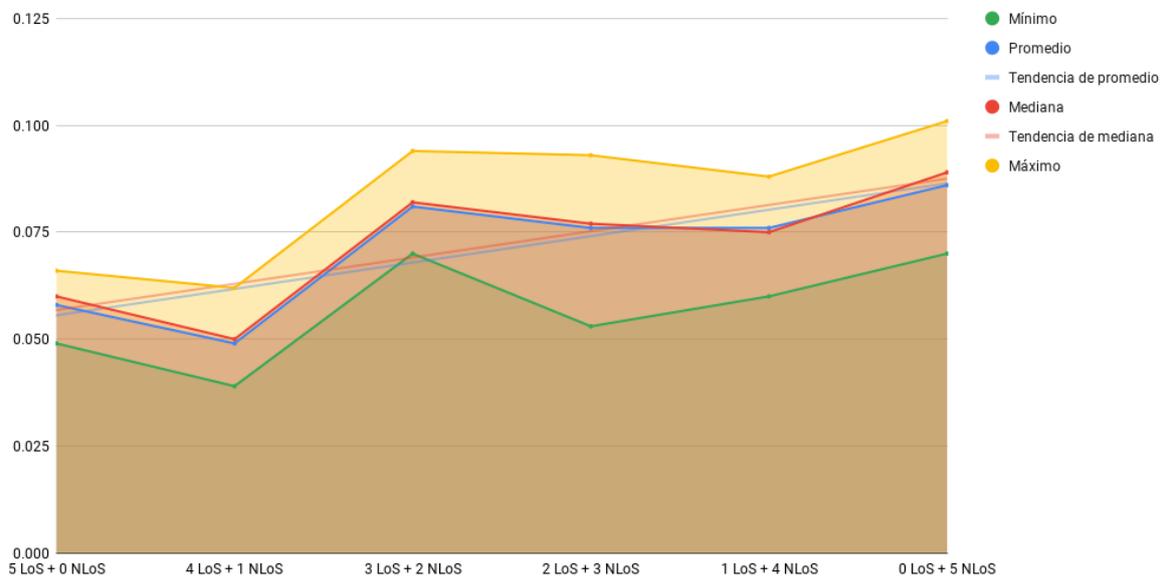
**Figura 5.6:** Imagen de RViz de los puntos utilizados en el experimento. El punto 8 corresponde a la posición del *tag*; en el resto de los puntos se colocaron los *anchors*.

A continuación se expone la tabla de datos que resultó de la ejecución del experimento.

Escenario	Promedio	Mediana	Mínimo	Máximo	Desviación estándar	Anchors en LOS	Anchors en NLOS
5LoS + 0NLoS	0.058	0.060	0.049	0.066	0.00521	1,2,3,14,18	
4LoS + 1NLoS	0.049	0.050	0.039	0.062	0.00524	2,3,14,18	1
3LoS + 2NLoS	0.081	0.082	0.070	0.094	0.00453	1,3,18	2,14
2LoS + 3NLoS	0.076	0.077	0.053	0.093	0.00793	2,14	1,3,18
1LoS + 4NLoS	0.076	0.075	0.060	0.088	0.00785	14	1,2,3,18
0LoS + 5NLoS	0.086	0.089	0.070	0.101	0.00590		1,2,3,14,18

**Tabla 5.4:** Variación del error según escenario, comenzando con cinco *anchors* con línea de visión y ninguno con obstáculo hasta llegar a la totalidad con obstáculos.

A continuación se muestran los datos de la tabla 5.4 en forma gráfica. Se puede observar que las líneas de tendencia del promedio y la mediana del error presentan una tendencia leve al alza, lo que es consistente con el aumento de los *anchors* con el tablón en el medio y la disminución de aquellos con línea de visión directa.



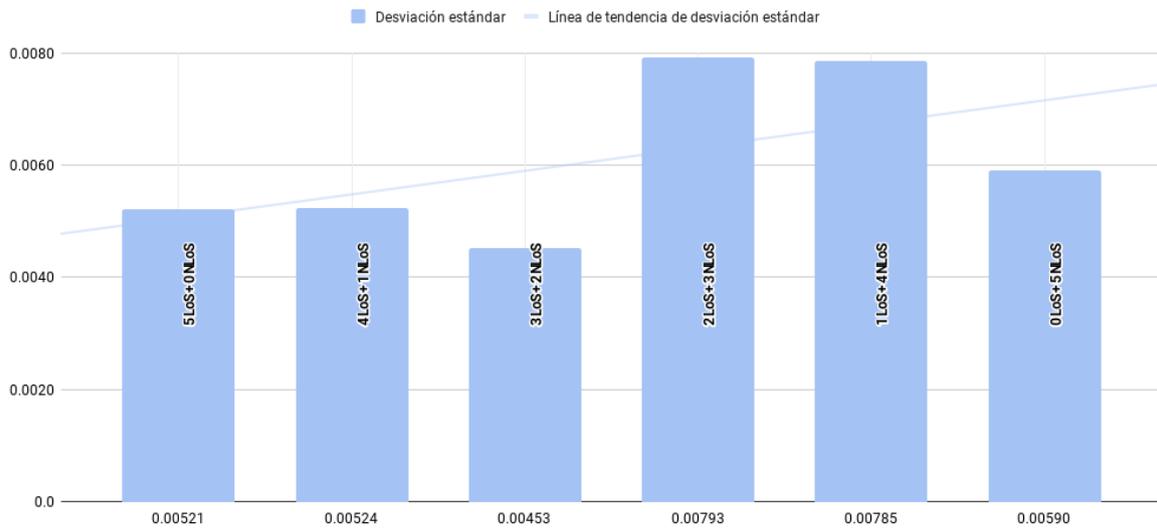
**Figura 5.7:** Gráfica de error según tabla de datos 5.4.

Según Yavari y Nickerson (2014), este comportamiento se debe a que en escenarios donde existe un obstáculo entre el nodo emisor y el receptor la señal que se transmite debe recorrer un camino indirecto más largo (sec. B.3) para alcanzar el nodo receptor. Por otro lado, las frecuencias bajas de la señal de UWB pueden atravesar ciertos materiales según su densidad y dimensiones (sec. 3.1.1). En ambas situaciones, el tiempo de viaje de la señal se incrementa, al seguir un camino más largo o traspasar un objeto con mayor densidad. Como consecuencia, el tiempo de llegada también aumenta y por lo tanto afecta la estimación de la distancia del nodo fuente al objetivo. Como resultado final, la precisión de todo el sistema se ve afectada en forma negativa.

Siguiendo con el análisis de la gráfica de la figura 5.7, a pesar de la afectación negativa de los obstáculos en la precisión en el proceso de estimación, esta es leve considerando la aplicación de este sistema en la realidad planteada. Tomando la mediana como valor comparativo del error entre los distintos escenarios, se observa que la diferencia entre la mediana del escenario con todos los nodos ocluidos y la del escenario con la totalidad de nodos sin oclusión no supera los tres centímetros.

A continuación se presenta la gráfica de desviación estándar del error (fig. 5.8) de acuerdo a los datos

en la tabla 5.4. En esta también se aprecia una línea de tendencia ascendente, que es consecuente con el análisis anterior. Si bien la medida de dispersión del error en todos los escenarios es baja, en general se observa que a mayor cantidad de nodos fuente con obstáculos, mayor es la dispersión en las distancias entre el punto estimado por el sistema y el punto real, en los distintos muestreos. Esto es un indicador de que la calidad de las estimaciones, estabilidad y rendimiento del sistema decrecen a mayor cantidad de nodos obstaculizados.



**Figura 5.8:** Gráfica de desviación estándar según tabla de datos 5.4.

En el anexo E se presentan los datos y un análisis de este experimento pero llevado a cabo aplicando filtros configurables al resultado del módulo de cálculo de posición.

## 5.4. Efecto de la inclinación vertical de la antena en la precisión del sistema

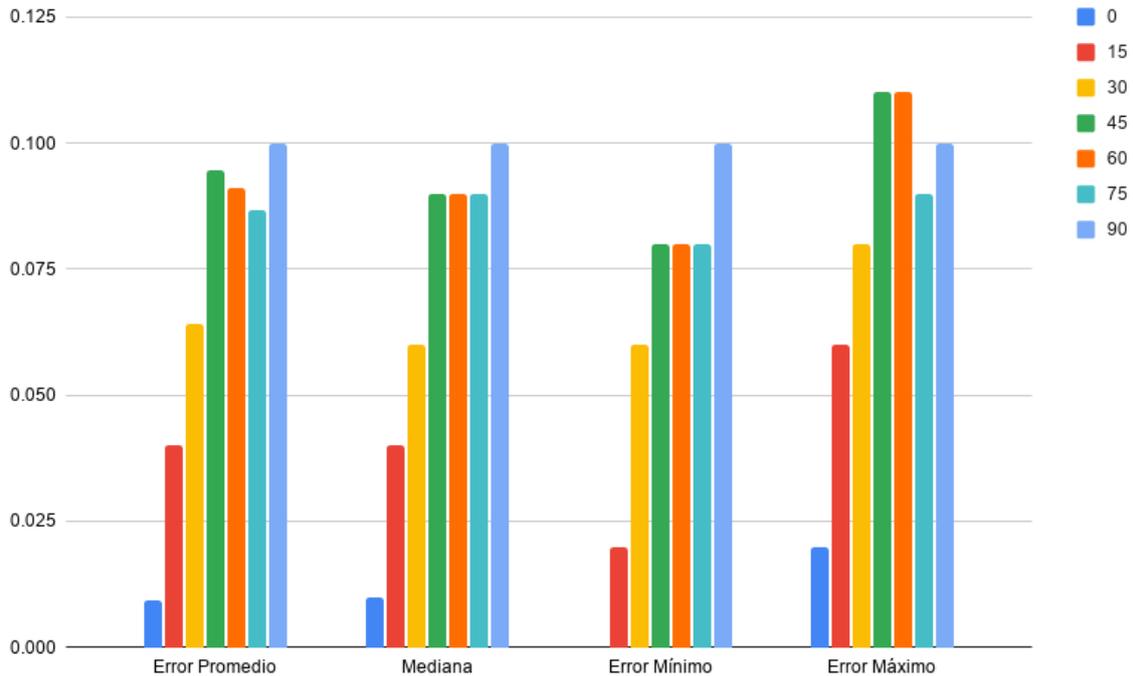
Una de las recomendaciones que se pueden encontrar en la documentación del fabricante es que las antenas deben estar en posición vertical para tener una mejor precisión en la estimación de distancias. Es por esto que se decidió experimentar cómo varía el error en la distancia en función del ángulo de inclinación con la vertical de la antena.

El experimento consiste en colocar dos antenas, un *tag* y un *anchor* a una distancia de 3.23 m, distancia que permanece fija durante el transcurso del experimento. Luego, se obtienen 100 lecturas del proceso de estimación de distancia. A continuación de la toma de datos, se cambia la inclinación de la antena *tag* y se vuelven a realizar otras 100 lecturas. Este proceso se repitió incrementando el ángulo de a 15°, comenzando en 0° y finalizando en 90°, donde 0° es el caso base o de referencia donde ambas antenas están paralelas y 90° donde las antenas forman un ángulo recto. Con estos datos se calcula el error promedio, mediana del error, error mínimo, error máximo y la desviación estándar del error, para cada escenario.

A continuación se detallan los datos resultado del experimento.

Inclinación (°)	Error promedio (m)	Mediana (m)	Error mínimo (m)	Error máximo (m)	Desviación estándar
0	0.009	0.010	0.000	0.020	0.0046
15	0.040	0.040	0.020	0.060	0.0111
30	0.064	0.060	0.060	0.080	0.0051
45	0.095	0.090	0.080	0.110	0.0054
60	0.091	0.090	0.080	0.110	0.0075
75	0.087	0.090	0.080	0.090	0.0048
90	0.100	0.100	0.100	0.100	0.0000

**Tabla 5.5:** Variación de la distancia en función de la inclinación de la antena.



**Figura 5.9:** Gráfica que muestra la variación de los valores de error en las medidas a medida que se cambia el ángulo del *tag*.

Con respecto a la rotación en el eje vertical, el fabricante asegura que no se ve afectado el cálculo de distancias, por lo que no se hicieron experimentos.

En la figura 5.9 se grafican los valores de error obtenidos a medida que se cambia el ángulo. Es posible ver que la variación del ángulo implica una desviación de hasta 10 centímetros con respecto al escenario donde los dispositivos están paralelos y en posición vertical. Este valor es bastante cercano al valor del error máximo de las antenas UWB de Decawave, por lo que si se suma este valor al error que tienen las antenas normalmente, se podría llegar a tener un error de  $\pm 20$  cm en el cálculo de la distancia entre los dispositivos. Por lo tanto, lo deseable sería tener siempre las antenas en posición vertical, sobre todo la del dispositivo que está en movimiento.

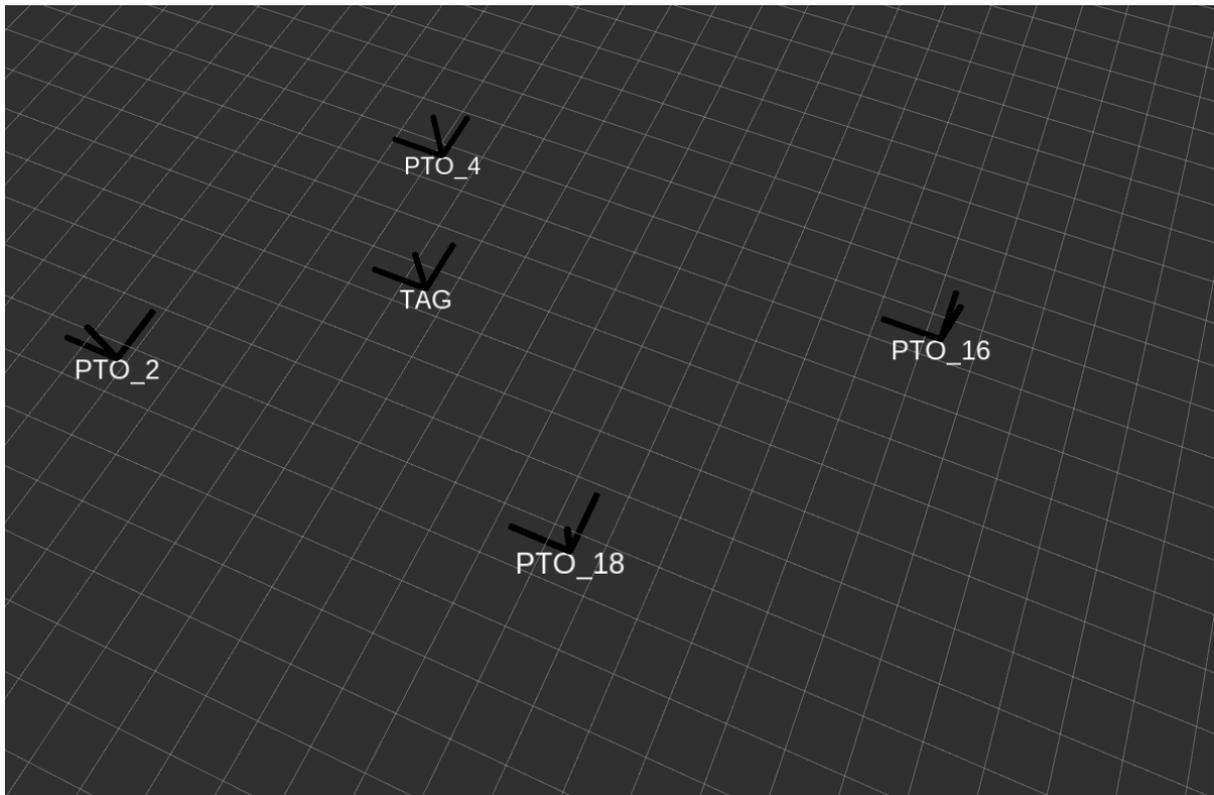
## 5.5. Efecto de la topología en la precisión del sistema

Un aspecto interesante de analizar es la disposición de los *anchors* con respecto al elemento que se quiere localizar. Por esta razón, se realizaron experimentos simulados en los que se modificaron las posiciones de los *anchors* y se dejó fija la posición del *tag*.

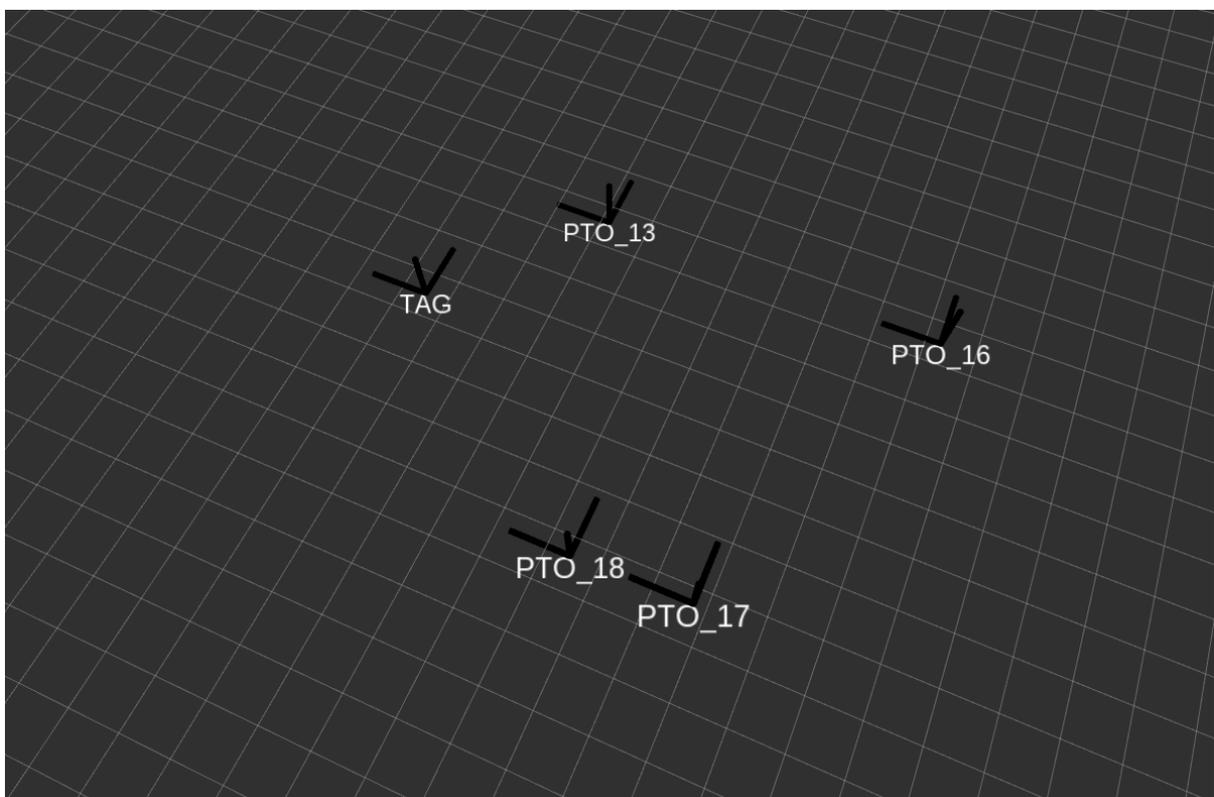
Es importante destacar que no es el mismo experimento que en la sección 5.1 con cuatro puntos, ya que en este experimento importa la disposición en el espacio de esos cuatro *anchors*, mientras que en el anterior no. Es por esto que puede verse una diferencia en los datos obtenidos.

A continuación se describen los escenarios y los resultados encontrados.

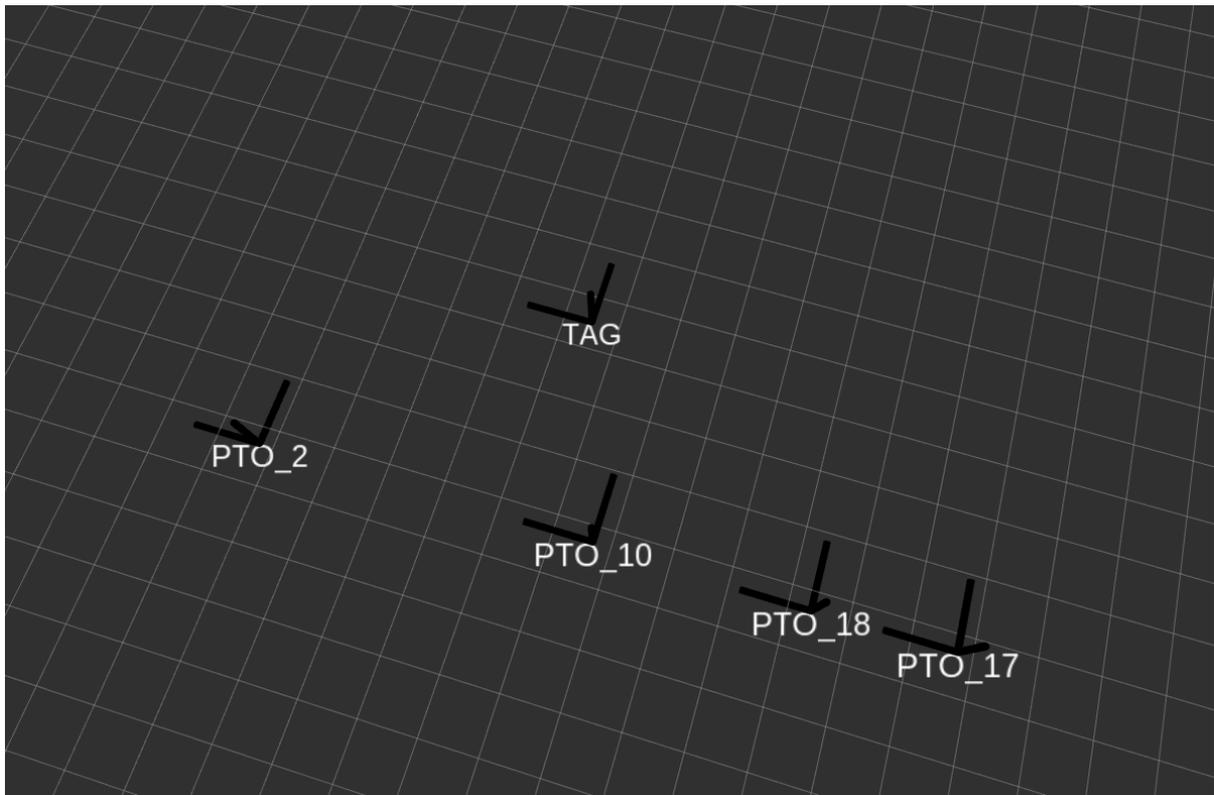
1. **El *tag* se encuentra dentro del grupo de *anchors*.** En este caso, el *tag* se coloca dentro del cuadrilátero formada por un grupo de cuatro *anchors*; en la figura 5.10 se puede ver la disposición de estos.
2. **El *tag* se encuentra fuera del grupo de *anchors*.** Para este escenario, los *anchors* forman un área de cuatro puntos, donde el *tag* se encuentra fuera de dicha área. Este escenario se podría ver como el opuesto al anterior y podría ser el más parecido a la realidad, porque el *tag* se mueve dentro de un grupo de *anchors*. Para visualizar este escenario, se puede ver la figura 5.11.
3. **Los *anchors* se encuentran alineados.** Un dato interesante planteado por el fabricante de los dispositivos en cuanto al cálculo de distancias y que por lo tanto afecta al cálculo de posición es que se puede ver un aumento del error al encontrarse todos los *anchors* alineados, motivo por el que se planteó este escenario. La figura 5.12 muestra la disposición de los dispositivos.
4. **Los *anchors* forman una “L”.** En este caso, los *anchors* se dispusieron formando una “L” y el *tag* queda por fuera de esta figura, como se puede ver en la figura 5.13.



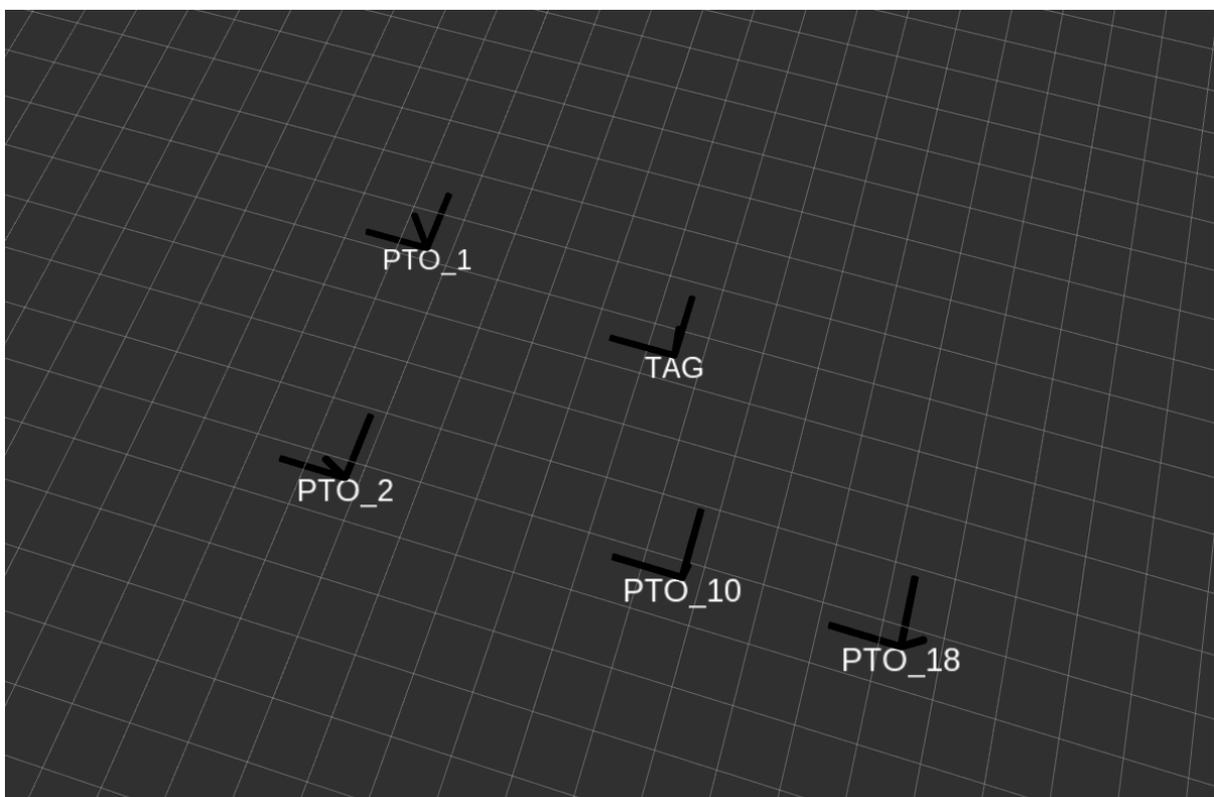
**Figura 5.10:** Posiciones de los dispositivos en el escenario generado con la herramienta RViz.



**Figura 5.11:** Posiciones de los dispositivos en el escenario generado con la herramienta RViz.



**Figura 5.12:** Posiciones de los dispositivos en el escenario generado con la herramienta RViz.



**Figura 5.13:** Posiciones de los dispositivos en el escenario generado con la herramienta RViz.

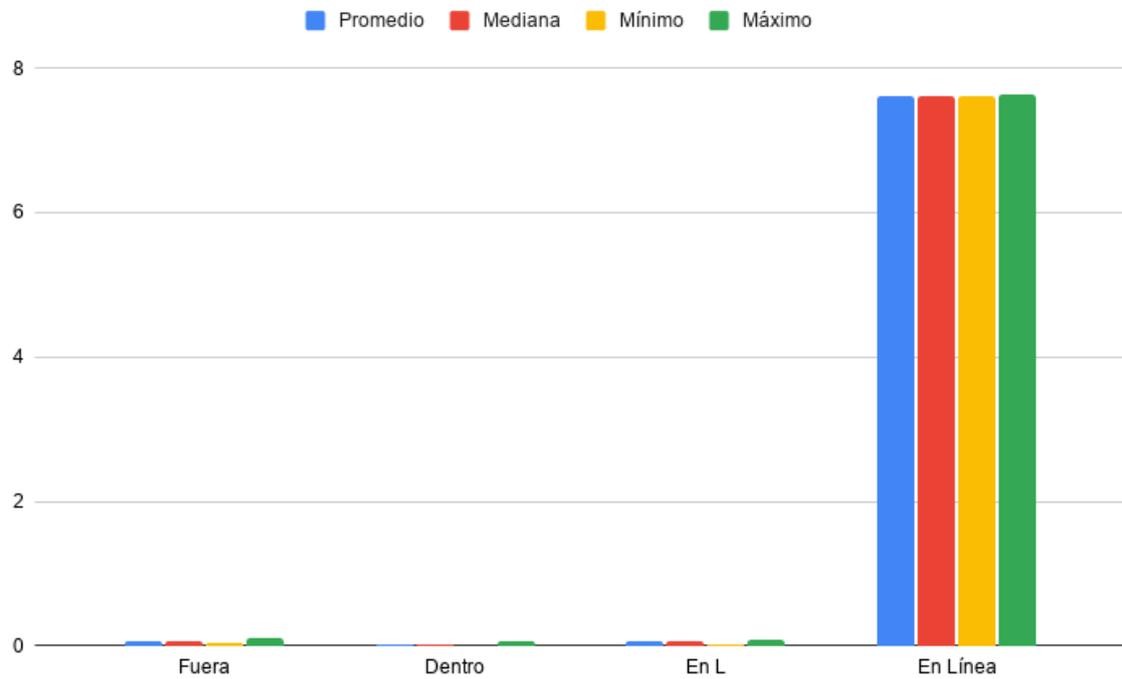
Los resultados obtenidos para cada escenario son los siguientes:

Disposición	Promedio	Mediana	Mínimo	Máximo	Desviación estándar
Fuera	0.072	0.070	0.048	0.099	0.0120
Dentro	0.011	0.012	0.005	0.065	0.0043
En L	0.053	0.053	0.029	0.079	0.0079
En línea	7.607	7.611	0.072	7.631	0.2388

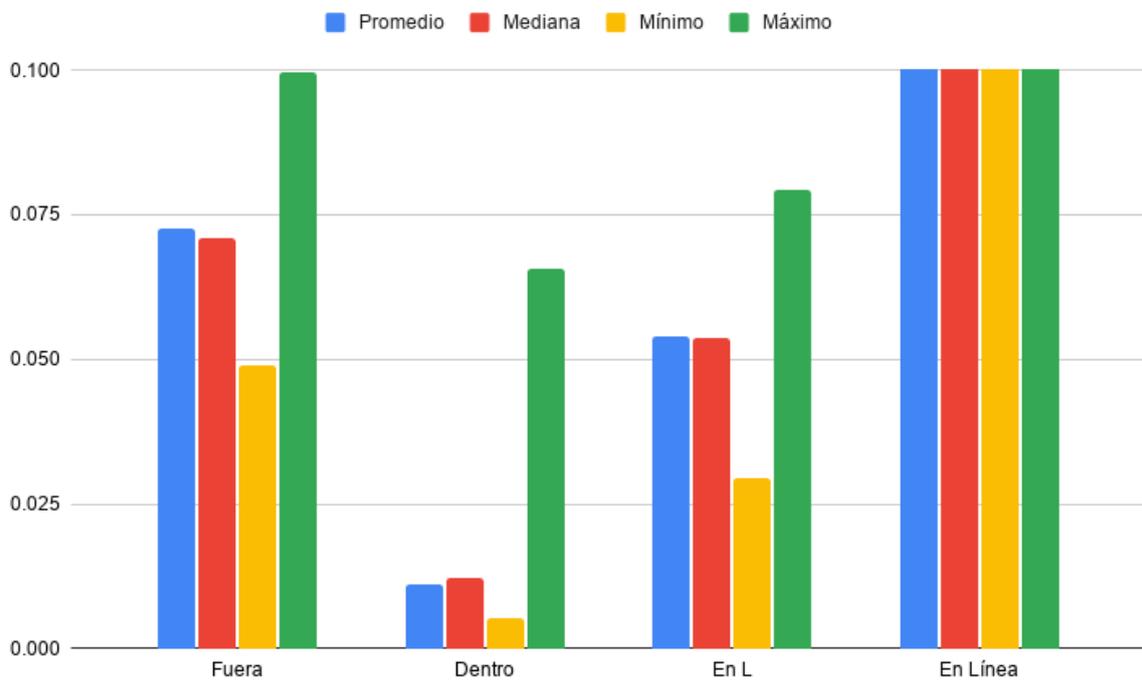
**Tabla 5.6:** Valores del error según la disposición del *tag* con respecto a los *anchors*.

A partir de los datos es fácil ver que cuando los *anchors* están alineados el error aumenta considerablemente, tal como afirma el fabricante, por lo que sería el escenario a evitar en un despliegue real. Por otro lado, el mejor escenario es aquel en el que el *tag* se encuentra rodeado por los *anchors*.

Es posible ver una gráfica comparativa de los valores en la figura 5.14. La diferencia de errores entre los escenarios es tan grande que los que tienen menor error apenas son visibles (fig. 5.14a), por lo que se muestran todos en una escala del eje de error tal que permita compararlos por igual (fig. 5.14b). En esta última gráfica se puede ver que el mejor escenario posible es cuando los *anchors* están rodeando al *tag*.



(a)



(b)

**Figura 5.14:** Comparación de los valores de error promedio, mediana del error, error mínimo y máximo por escenario. Dado que el error es muy grande para varios escenarios, como se puede apreciar en la figura (a), se ajusta la escala para ver todos los valores de todos los casos, como se muestra en la figura (b).

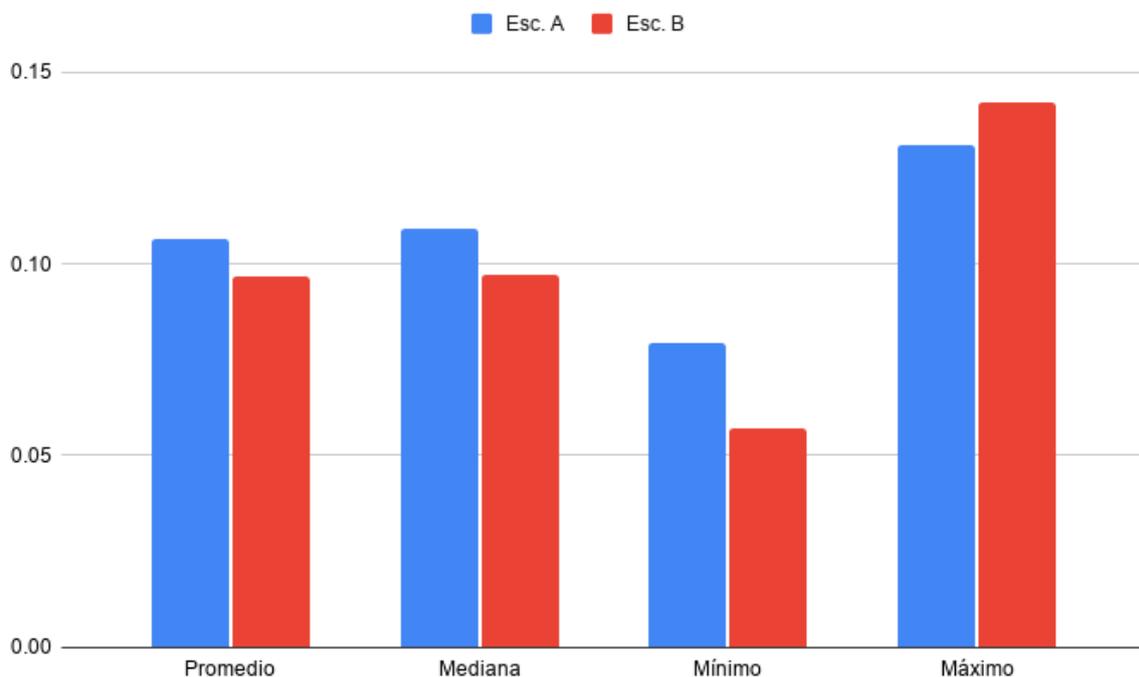
## 5.6. Efecto de la propagación multicamino

Otro punto importante que destaca la tecnología es que tampoco se ve alterada por el efecto de Propagación multicamino (Anexo B.3). En este caso, se tomaron los datos de tres *anchors* y se los comparó con un juego de datos de los mismos *anchors*, pero con el tablón usado en la sección 5.3 a un metro del *tag* y en la posición contraria a donde están dichos *anchors*, tratando de generar un efecto de “rebote” de las señales de estos. También se calculó el error provocado por la proximidad del tablón y se comparó con la misma distribución, pero sin el tablón.

Los datos recabados se muestran en la tabla 5.7, donde *Esc. A* corresponde a la disposición que no tiene el tablón y *Esc. B* a la que sí lo tiene. En todos los casos se calculó el error promedio, mediana del error, error mínimo, error máximo y desviación estándar, utilizando el promedio como valor para comparar ambos escenarios.

	Promedio	Mediana	Mínimo	Máximo	Desviación estándar
Esc. A	0.106	0.109	0.079	0.130	0.0099
Esc. B	0.096	0.097	0.057	0.142	0.0144

**Tabla 5.7:** Datos comparativos del experimento de efecto propagación multicamino.



**Figura 5.15:** Comparación de los dos escenarios utilizados para el caso del efecto de la propagación multicamino.

Al graficar los datos, en la figura 5.15 es posible ver que casi no hay diferencias en el error entre ambos escenarios tomando el promedio como referencia; esta diferencia es de apenas un centímetro. En ambos escenarios, el error está dentro del indicado por el fabricante del hardware. Para el *Esc. B*, esto se debe a las optimizaciones que tiene el hardware para resolver el problema de la propagación multicamino.

## Capítulo 6

# Conclusiones

El proyecto buscó desarrollar un sistema de posicionamiento interior que permita ser integrado a agentes robóticos. Así, el objetivo principal consistió en seleccionar una tecnología que lograra posicionar un agente dentro de un ambiente interior.

El sistema desarrollado utiliza la tecnología UWB, que permite sortear los principales inconvenientes que tiene la localización interior. A su vez, dicho sistema es fácilmente integrable a un agente robótico que utilice ROS.

Entre las primeras conclusiones obtenidas, se destaca que la tecnología utilizada es capaz de pasar los obstáculos más importantes que impone la localización interior, ya que por experimentación se pudo comprobar que dicha tecnología se vio poco afectada por el problema de propagación multicamino y la falta de línea de visión entre los dispositivos.

En cuanto al dispositivo utilizado en concreto, este es muy completo en lo que respecta a funcionalidades y posibilidades de expansión, simplificó el proceso de prototipado y evitó la necesidad de fabricación de una placa con las características que posee. Cabe destacar que el módulo *WiFi* puede ser útil si se piensa implementar un proceso de autocalibrado o enviar parámetros desde un servidor central, o durante la etapa de desarrollo para envío de registros de actividad y depuración del código. Otro aspecto importante a destacar es que el costo de la placa no la hace prohibitiva, aunque solamente es vendida para fines de investigación.

Con respecto a la integración con ROS, es interesante destacar que fue sencilla gracias a la utilización de Rosserial, que simplificó el uso de tópicos para la publicación de las distancias recibidas de los *anchors* y procesadas luego por el agente. A su vez, facilitó la implementación del calibrador utilizando los tópicos para publicar cambios de configuraciones en el *anchor* de calibración.

En lo que refiere a la solución implementada, a pesar de que no fue posible probar su funcionamiento integrado a la pila de navegación de un sistema robótico completo, quedaron las bases para que sea fácilmente integrable a cualquier agente robótico que utilice ROS.

Asimismo, este sistema podría ser utilizado como referencia (*ground-truth*) para la evaluación durante el desarrollo de otras soluciones de posicionamiento interior, debido a la capacidad de establecer la posición absoluta del agente robótico.

Aunque la versión de ROS utilizada en el proyecto (Kinetic) dejará de contar con mantenimiento en

breve<sup>1</sup>, la continuidad de este está asegurada por el uso de la herramienta Docker. Esta herramienta también permitiría ejecutarlo en un ambiente diferente al actual, siempre y cuando soporte contenedores Docker.

Para finalizar, en lo que respecta al proceso de desarrollo de este proyecto, se presentaron varios inconvenientes que lo enlentecieron. Entre los más importantes se encuentran: pérdida de conexión por obstáculos o largas distancias entre *anchor* y *tag*, falta de calibración de *antenna delay*, biblioteca *arduino-dw100* utilizada obsoleta y con errores en el protocolo de cálculo de distancias (pérdida de ciclos de estimación al no recibir respuesta del último *anchor* de la lista) y problemas de programación (pérdida de información en conversión de tipos, corrección en tamaños de estructuras, entre otros). Por más detalles de estos y otros problemas hallados en el proyecto, ver el anexo C. A pesar de estos inconvenientes se pudo llegar a buen puerto, según consta en esta investigación.

---

<sup>1</sup><https://index.ros.org/doc/ros2/Releases/>

# Capítulo 7

## Trabajo a futuro

A lo largo del presente trabajo se encontraron problemas que fueron solucionados y se implementaron algunas mejoras alineadas al objetivo del proyecto, pero también se hallaron otras posibles mejoras a realizar. Dado el alcance de este proyecto y a que dichas posibles mejoras deben ser investigadas y desarrolladas en su propio contexto, se presentan en esta sección como trabajo a futuro.

### 7.1. En dispositivos

- Entre las primeras mejoras que deberían realizarse se destaca revisar el protocolo de cálculo de distancias que implementa la biblioteca `arduino-dw1000`. Si bien para el proyecto cumple su cometido, esta biblioteca no está diseñada para que el *anchor* envíe el mensaje de *range report* del protocolo de forma ordenada. A pesar de funcionar ahora, se debería revisar todo el protocolo. Por otro lado, se encontró que la codificación es bastante engorrosa y no es autodocumental, lo que hace que el código sea inmantenible y genera dificultades para entender las distintas funcionalidades.
- Respecto al protocolo usado en la biblioteca (Double-sided Two-way Ranging), este podría ser reemplazado por un nuevo protocolo llamado *Double Two-Way Ranging (D-TWR)* propuesto por Kwak y Chong (2010) y basado en Time of Arrival (ToA) (sec. 3.3.2.1). Este nuevo protocolo supone una reducción en la cantidad de mensajes necesarios en un ciclo de estimación, es más robusto, mejora la precisión, disminuye el tiempo del proceso de cálculo de distancia entre los dispositivos y aumenta la vida útil de la batería.
- Otra mejora importante a implementar es que el *anchor* sea capaz de comunicarse con más de un *tag* a la vez, dado que en un escenario real puede haber más de un dispositivo móvil.
- Un detalle a solucionar es la capacidad de la red en cuanto a la posibilidad del *tag* de comunicarse con una mayor cantidad de *anchors*. En pruebas de laboratorio, el sistema llegó a soportar hasta cuatro *anchors* funcionando al mismo tiempo; al agregar el quinto, el sistema deja de funcionar correctamente: el *tag* se bloquea y se desconecta de todos los *anchors*. Esto se debe a que el protocolo desarrollado en la biblioteca `arduino-dw1000` utiliza mensajes con un marco de 127 bytes, en lugar del marco extendido que es de 1023 bytes. Por defecto la biblioteca utiliza el marco de tamaño 127 bytes (transmisión básica Decawave, 2017, sección 3.1), que se traduce en una limitación en el tamaño de los datos a enviar. Por lo tanto, para ciertos mensajes de difusión con datos enviados a todos los *anchors* de la red, el tamaño del marco implica una cota en la cantidad de dispositivos soportados simultáneamente. Como trabajo a futuro se podría analizar si la biblioteca soporta la configuración del marco extendido. Esto podría traer como ventaja el soporte de más *anchors* en forma simultánea y como contrapartida podría afectar el rendimiento del sistema e incluso podría

ser necesario volver a realizar el proceso de calibración para ajustar el valor del *antenna delay*.

- Sería interesante un desarrollo de una placa con las mismas características y ver si el tiempo que insumió solucionar problemas del producto comprado efectivamente es menos que el que lleva desarrollar la placa desde cero.
- En un escenario real de un agente robótico moviéndose en un ambiente interior con obstáculos, es posible que el *tag* pierda la línea de visión con alguno de los *anchors*. Dependiendo de las características del obstáculo, la señal puede ser afectada en mayor o menor medida. Por esta razón, es recomendable tener *anchors* de respaldo, es decir, una cantidad mayor a la mínima necesaria para la localización. Luego, se tendrían estimaciones de varios *anchors* en el ambiente; algunas podrían estar afectadas por NLoS de manera tal que su procesamiento por el estimador de posición se vea afectado negativamente. Basado en este problema se propone la implementación de un filtro, en el *tag* o en el *anchor*, que sea capaz de descartar las estimaciones afectadas. Esto se podría implementar utilizando el valor de la intensidad de la señal recibida, que junto con el valor de la distancia ya son reportados por los *anchors* al final de cada ciclo de estimación y recibidos por el *tag*. Si el valor de la intensidad de la señal recibida es inferior a un umbral definido como base, entonces se descartaría la estimación de distancia correspondiente y no se propagaría al estimador de posición, por lo que este solo usaría estimaciones de “buena calidad”. Es posible encontrar más detalles sobre el efecto de la calidad de la señal recibida sobre la estimación en Decawave (2017), sección 4.7.
- La antena UWB DW1000 posee sensores para medición de temperatura interna y voltaje en el microprocesador integrado (Decawave, 2017, sección 6.4). Según el fabricante, una alta temperatura del microprocesador y grandes cambios en el voltaje suministrado podrían afectar el rendimiento de la antena. Como métodos para compensar estos efectos y tener una precisión más refinada, se sugiere monitorear dichos valores y ajustar dinámicamente ciertos parámetros como el *antenna delay* (Decawave, 2017, sección 8). En la práctica se observó que en el uso prolongado de los dispositivos la antena alcanza altas temperaturas, por lo que también sería recomendable la instalación de un disipador de calor.

## 7.2. Desarrollo fuera de los dispositivos

- En cuanto al proceso de calibración, sería necesario investigar si existe una manera más óptima y rápida de calcular el valor del *antenna delay*, ya que actualmente todo el procedimiento insume demasiado tiempo (anexo D).
- Con respecto al proceso de cálculo de posición, se debería mejorar el filtro de Kalman para que las estimaciones de salida sean más precisas. Esto se puede observar con más detalles en el anexo correspondiente al experimento en la sección “Efecto de los obstáculos en la precisión del sistema” (anexo E). En principio se utilizaron los parámetros encontrados para este fin en Yavari y Nickerson (2014), en donde también se utilizó el módulo *DW1000*, pero parecerían no ser buenos para esta aplicación en particular.
- En el proceso que realiza el cálculo de posición a partir de las distancias se utiliza una biblioteca que es precisa al calcular en el plano, pero no en el espacio (coordenada  $Z$ ). Sería deseable investigar otras alternativas con las mismas prestaciones y que no tengan estos errores.
- Existe gran potencial para el desarrollo de nuevos filtros que mejoren la estimación de la posición. Actualmente, los filtros son estáticos y los ajustes se basan en cierto número de estimaciones previas. Este mecanismo podría mejorarse incluyendo ciertas medidas de calidad de la señal que provee el módulo DW1000, entre ellas la fuerza de la señal recibida (*Receive signal power*) y la fuerza de la señal recibida en el primer camino (*Signal power in the first path*). De esta manera, se podrían

atenuar los efectos de NLoS. A su vez, la estimación también podría ser ajustada en función de la temperatura y humedad ambiente. Otro enfoque en la implementación de filtros podría ser que el sistema seleccione los  $N$  “mejores” *anchors* (siempre que existan más de tres) para el cálculo de la posición, considerando  $N$  como el resultado del experimento en la sección 5.1. Esta selección de los “mejores” se basa en algún criterio que permita determinar si el valor de distancia de esos dispositivos se considera más cercano al real que el valor de otros. Un ejemplo de criterio puede ser el uso de la potencia de la señal, otro criterio podría ser la elección de los *anchors* cercanos según su disposición en el ambiente (ver sec. 5.5).

- En lo que refiere a la experimentación, sería interesante evaluar su funcionamiento con las estimaciones de distancia en tiempo real, en lugar de simular escenarios de múltiples dispositivos concurrentes.
- En relación con el punto anterior, sería necesario realizar mediciones de tiempo en el flujo de estimaciones, es decir, poder medir con qué velocidad el sistema actualiza la posición. Esta es una característica muy importante en sistemas de localización en tiempo real.
- Si bien el sistema quedó integrado a ROS, debido a la necesidad de acotar el alcance del proyecto quedó pendiente la realización de pruebas con un agente en movimiento.

## Apéndice A

# Comparativa de tecnologías

A continuación se presenta una tabla comparativa de las distintas tecnologías nombradas en la sección Marco teórico (sec. 3) y otras que fueron halladas en la investigación inicial y se pueden ver en el documento de Estado del Arte del proyecto (Furquez y Saldías, 2019), en la que se sintetiza la información encontrada.

Tecnología	Alcance máximo	Transferencia máxima	Consumo de energía	Ventajas	Desventajas
<b>IEEE 802.11n</b>	250 m exterior	600 Mps	Moderado	<ul style="list-style-type: none"> <li>- Ampliamente disponible</li> <li>- Alta precisión</li> <li>- No requiere equipamiento extra ni complejo</li> </ul>	<ul style="list-style-type: none"> <li>- Sensible al ruido ambiente</li> <li>- Requiere de algoritmos de procesamiento complejos</li> </ul>
<b>IEEE 802.11ac</b>	35 m interior	1,3 Gbps			
<b>IEEE 802.11ad</b>	2 m aprox.	4,6 Mbps			
<b>UWB</b>	10 - 20 m	460 Mbps	Moderado	<ul style="list-style-type: none"> <li>- Inmune a las interferencias</li> <li>- Alta precisión</li> </ul>	<ul style="list-style-type: none"> <li>- Rango más corto</li> <li>- Requiere equipamiento extra en dispositivos de usuario</li> <li>- Alto costo</li> </ul>
<b>Acústica</b>	2 m aprox.		Bajo a moderado	<ul style="list-style-type: none"> <li>- Puede ser usado por aplicaciones privadas</li> <li>- Alta precisión</li> </ul>	<ul style="list-style-type: none"> <li>- Afectado por polución sonora</li> <li>- Requiere balizas o equipamiento extra</li> </ul>
<b>RFID</b>	200 m	1,67 Gbps	Bajo	<ul style="list-style-type: none"> <li>- Bajo consumo de energía</li> <li>- Gran alcance</li> </ul>	<ul style="list-style-type: none"> <li>- Baja precisión</li> </ul>
<b>Bluetooth</b>	100 m	24 Mbps	Bajo	<ul style="list-style-type: none"> <li>- Bajo consumo de energía</li> <li>- Transmisión de datos y alcance de recepción altos</li> </ul>	<ul style="list-style-type: none"> <li>- Baja precisión</li> <li>- Sensible al ruido</li> </ul>
<b>Ultrasonido</b>	2 - 10 m	30 Mbps	Bajo a moderado	<ul style="list-style-type: none"> <li>- Comparativamente menor afectación por absorción de la onda</li> </ul>	<ul style="list-style-type: none"> <li>- Alta dependencia respecto del emplazamiento de los sensores</li> </ul>
<b>Luz visible</b>	1,4 km	10 Gbps	Relativamente alto	<ul style="list-style-type: none"> <li>- Disponibilidad a gran escala</li> <li>- Potencial para proveer gran precisión</li> <li>- Libre del problema de múltiples caminos</li> </ul>	<ul style="list-style-type: none"> <li>- Comparativamente mayor consumo de energía</li> <li>- El alcance es afectado por obstáculos</li> <li>- En principio requiere de <i>LoS</i> (sec. B.2)</li> </ul>
<b>SigFox</b>	50 km	100 bps	Extremadamente bajo	<ul style="list-style-type: none"> <li>- Gran alcance de recepción</li> <li>- Bajo consumo de energía</li> </ul>	<ul style="list-style-type: none"> <li>- Diseñado para uso en distancias grandes entre la estación base y el dispositivo</li> <li>- Atenuación severa de la señal de exterior a interior debido a estructuras internas de edificios</li> </ul>
<b>LoRa</b>	15 km	37,5 Kpbs	Extremadamente bajo	<ul style="list-style-type: none"> <li>- Gran alcance de recepción</li> <li>- Bajo consumo de energía</li> </ul>	<ul style="list-style-type: none"> <li>- Diseñado para uso en distancias grandes entre la estación base y el dispositivo</li> <li>- Atenuación severa de la señal de exterior a interior debido a estructuras internas de edificios</li> </ul>
<b>IEEE 802.11ah</b>	1 km	100 Kbps	Extremadamente bajo	<ul style="list-style-type: none"> <li>- Gran alcance de recepción</li> <li>- Bajo consumo de energía</li> </ul>	<ul style="list-style-type: none"> <li>- No explorado en profundidad para localización</li> <li>- Rendimiento pendiente de ser visto en ambientes interiores</li> </ul>
<b>Weightless</b>	<ul style="list-style-type: none"> <li>- P: 2 km</li> <li>- N: 3 km</li> <li>- W: 5 km</li> </ul>	<ul style="list-style-type: none"> <li>- P y N: 100 Kbps</li> <li>- W: 10 Mbps</li> </ul>	Extremadamente bajo	<ul style="list-style-type: none"> <li>- Gran alcance de recepción</li> <li>- Bajo consumo de energía</li> </ul>	<ul style="list-style-type: none"> <li>- Diseñado para uso en distancias grandes entre la estación base y el dispositivo</li> <li>- Atenuación severa de la señal de exterior a interior debido a estructuras internas de edificios</li> </ul>

**Tabla A.1:** Tabla de tecnologías inalámbricas de localización (Zafari et al., 2017).

# Apéndice B

## Fuentes de error

En Küpper (2005) se describen varias fuentes de error que pueden afectar la precisión de los sistemas de localización. A continuación, se enumeran factores que podrían afectar a los sistemas de posicionamiento interior.

### B.1. Relojes

Relojes inadecuados o de mala calidad dentro de los sistemas electrónicos usados pueden influir directamente en los cálculos de tiempo o rango de tiempo. Además, los mecanismos de sincronización o cálculos matemáticos usados siempre tienen un error potencial. Los relojes siempre son una fuente de error en mecanismos que usen el tiempo para determinar distancias, como lo son *Tiempo de vuelo* (sec. 3.3.2.1) o *Diferencia de tiempo de llegada* (sec. 3.3.2.2).

### B.2. Ausencia de línea de visión

Con línea de visión (*Line of Sight - LoS*) se hace referencia a que existe un camino directo entre un emisor y el receptor de una señal (acústica, visual o de radiofrecuencia) sin ningún obstáculo. Si no hay línea de visión (NLoS), significa que existe un obstáculo entre el emisor y el receptor.

Cuando hay *LoS* entre el receptor y el emisor, se asume que el primer pulso que llega al receptor es el que toma el camino directo y el resto son repeticiones dadas porque la onda se reflejó en objetos del ambiente. Por lo tanto, cuando no se tiene *LoS* todos los pulsos que llegan al receptor son reflejos del pulso original que toman distintos caminos, por lo que el tiempo que demora en llegar la onda reflejada es mayor que si esta hubiese llegado directo. Esto es conocido como propagación multicamino y su efecto provoca que se tengan distintas medidas de tiempo para la misma señal. Esto genera errores al medir los tiempos, que se traducen en errores al estimar distancias. Este es un error muy común al usar *Angulación* (sec. 3.3.3) con posición fija, dado que la señal llega desde casi cualquier dirección.

### B.3. Propagación multicamino

Este error, también conocido como *Multipath Propagation* o *Multipath Fading* por el efecto que causa, es muy común, afecta la amplitud y la fase de la señal y puede generar distorsión en la onda. La propagación multicamino es provocada por reflejos de la señal emitida en objetos del ambiente y su efecto también genera “copias” de la señal original. Por lo general ocasiona errores en las técnicas que determinan la distancia del emisor por medio de fuerza de señal recibida (*Received Signal Strength (RSS)*) (sec. 3.3.2.5) o medición de la fase de la señal. Si se usa la técnica de *Fingerprinting* (sec. 3.3.5.1) al

recolectar los valores de la señal en un ambiente, estos se vuelven inservibles si el ambiente cambia, por ejemplo, al quitar o agregar obstáculos, dado que esto genera cambios en los caminos que recorren las señales.

## B.4. Mala geometría

Si las posiciones conocidas de las balizas (dispositivo de posición fija que sirve de referencia) no son correctas, los cálculos por *Lateración* se verán afectados y agregarán error al cálculo matemático. Por otra parte, que las posiciones de estas balizas no sean adecuadas también afecta el cálculo e incrementa el error que ya de por sí tiene el mecanismo; un ejemplo de esto podría ser que algunas balizas estén muy cerca entre ellas. En Küpper (2005), el autor acota que la mejor geometría que se puede obtener es posicionando las balizas de forma tal que los ángulos entre ellas sean todos rectos.

## Apéndice C

# Inconvenientes encontrados

Si bien el fabricante de las placas vendía el dispositivo con la antena de UWB, esta no viene soldada a la placa principal, por lo que iba a ser necesario soldarla. Al llegar los dispositivos, no solo se encontró que no tenían las antenas UWB soldadas, sino que tampoco traían soldados los módulos *WiFi* ni los pines de conexión a la placa, como se puede ver en la imagen C.1a. Para el equipo fue todo un desafío soldar los módulos, ya que eran módulos de soldadura superficial y no se tenía experiencia en esta área.



(a) Dispositivo sin los módulos ni cabezales.

(b) Placa con el módulo UWB soldado.

**Figura C.1:** Ensamblado de componentes.

Luego de tener los componentes ensamblados se procedió a cargar los *firmwares* correspondientes, tanto en el módulo *WiFi* como en el microcontrolador *SMT32 (Arduino)*. Para el primero se utilizó un *firmware* llamado *ESPEasy*<sup>1</sup>, que expone una interfaz sencilla, capaz de transmitir datos a través de la red *WiFi* utilizando comandos de texto que se envían por un puerto serial dedicado a la comunicación con dicho módulo. En el caso del microcontrolador *SMT32*, se contaba con dos *firmwares* proporcionados por el fabricante: uno para operar como *anchor* y el otro como *tag*. A nivel de componentes físicos, no existe distinción entre *anchor* y *tag*; su comportamiento es determinado por el *firmware* instalado. En cuanto a la configuración inicial de los dispositivos, surgió el contratiempo de que el manual del fabricante de Localino describía procedimientos y uso de programas basados en el sistema operativo Windows, mientras que para el proyecto el ambiente de trabajo seleccionado fue una distribución de Linux. Luego de encontrar las herramientas adecuadas y adaptar los procedimientos no hubo problemas para cargar el *firmware*.

El siguiente paso consistió en hacer una prueba con el programa que el fabricante del dispositivo proporcionaba para posicionar el *tag*. Se encontró que la aplicación de posicionamiento no funcionaba; si bien los *anchors* transmitían sus respectivas distancias con el *tag* a través de *WiFi*, dicha aplicación daba errores al procesar los mensajes. Luego de investigar el *firmware* del *anchor* y ver el código del programa de posicionamiento, se encontraron diferencias en el protocolo de envío de distancias entre el *anchor* y el programa de posicionamiento, por lo que se sospechó que fuesen versiones incompatibles. En consultas al proveedor sobre este problema, este nunca dio una respuesta concreta ni proveyó otro aplicación o solución alguna. Luego de hacer una revisión en profundidad del código del programa de posicionamiento, fue posible corregir el formato del mensaje en el *firmware* del *anchor* para que fuera compatible con lo esperado por el estimador de posición.

Con los *anchors* informando las distancias en el formato correcto, se encontró que habían diferencias considerables (mayores al error máximo descrito por el fabricante del módulo DW1000) entre las distancias estimadas y la real, incluso en distancias cortas. Además, al haber objetos entre los dispositivos, estos dejaban de interactuar. El mismo efecto se observó en distancias más largas: en el orden de los diez metros los dispositivos dejaban de comunicarse entre sí. Los principales motivos de la elección de la tecnología eran su gran alcance y la poca afectación de la señal ante obstáculos. El fabricante del módulo UWB aseguraba en la hoja de datos del componente que el cálculo de distancias no tendría este tipo de problemas y que además podría cubrir distancias de hasta 290 metros.

Tras leer varios documentos de Decawave se llegó al manual de usuario y se encontró un apartado que hablaba sobre un parámetro configurable llamado *antenna delay*. Este parámetro es calculable, hace referencia al tiempo que tarda en pasar la información desde el dispositivo al medio de propagación e influye en los cálculos de distancia. Este tiempo incluye tiempos de proceso en la placa donde está instalado el módulo. Para dicho parámetro, el fabricante proporciona un algoritmo que ayuda a determinar el *antenna delay* a partir de dos dispositivos. El algoritmo se puede consultar en Decawave (2017). A partir de este algoritmo se desarrolló un procedimiento para la calibración de los dispositivos (descrito en el anexo D), basado en el supuesto de que todos son iguales, por lo que basta ejecutar el procedimiento para un dispositivo y el valor encontrado se aplicaría al resto.

Con el proceso de calibración se lograron obtener medidas más acordes a las reales, pero persistía el problema generado al interponer un objeto entre el *anchor* y el *tag*, que provocaba que estos dejaran de interactuar. Analizando la documentación de usuario de Decawave, se encontró que existen parámetros que influyen en el alcance de la señal, la detección de orígenes de múltiples caminos y la precisión. Los principales parámetros son: el largo del preámbulo del mensaje, la frecuencia del pulso y en la tasa de transmisión de datos. El preámbulo refiere a una secuencia de pulsos repetidos siguiendo un patrón definido en el estándar IEEE 802.15.4, que permite la sincronización entre el dispositivo emisor y el receptor para la correcta detección de una transmisión. Según el documento Decawave (2018b, sección 5.1), una mayor longitud del preámbulo resulta en una detección más precisa del tiempo de llegada de la señal por el primer camino y el rendimiento de largo alcance, en perjuicio de un mayor tiempo de

---

<sup>1</sup><http://espeasy.com>

transmisión y por lo tanto un mayor consumo de energía (Decawave, 2018a, sección 3.7).

Este preámbulo está preconfigurado en la biblioteca *arduino-dw1000*, que fue provista por el fabricante del dispositivo. En esta biblioteca, el cambio de preámbulo está atado a un cambio de frecuencia y de tasa de transmisión, a pesar de que el fabricante del módulo DW1000 no lo especifica así. Si bien las configuraciones de preámbulo, tasa de transmisión y frecuencia preestablecidas por la biblioteca son las recomendadas por el fabricante, al usarla se está condicionando a cambiar estos valores al tratar de configurar uno específicamente. En función de la documentación de Decawave (Decawave, 2018b), se seleccionó la preconfiguración que más se ajustaba a las recomendaciones para afrontar el problema de *Non Line of Sight* (anexo B.2) o falta de visión directa entre los dispositivos. Por obvias razones, al haber un cambio en estos parámetros (frecuencia, tasa de transmisión y largo del preámbulo) hubo que volver a hacer la calibración del valor de *antenna delay*. Luego de hacer el cambio en los parámetros y recalcular el *antenna delay*, el sistema dio valores bastante más cercanos a los reales y además quedó más robusto con respecto a la interferencia de objetos.

Luego de tener el sistema base andando se empezó a hacer pruebas para que los *anchors* le reporten la distancia calculada al *tag* y este sea el que envía la distancia al software de localización. El protocolo de la biblioteca que se utiliza en la comunicación del módulo *SMT32* con el módulo UWB es el mismo en ambos dispositivos, pero varía su funcionamiento según a qué dispositivo representa. Se encontró que el *tag* perdía comunicación en el medio del proceso de cálculo de distancias cuando interactuaba con más de dos *anchors*. El fabricante del dispositivo aseguraba que se podrían tener hasta diez *tags* interactuando con el *anchor*, pero esto es utilizando el *sistema con posicionamiento basado en la red* (sec. 3.2), que es la arquitectura que usa Localino. En este proyecto se plantea implementar *posicionamiento basado en la terminal* (sec. 3.2), donde es el *tag* el que recaba la información para localizarse (o enviarlo al software de localización), y este hecho cambia las características en cuanto a la capacidad de la red. Por lo tanto, se decidió analizar la biblioteca que proveía Localino (una versión inicialmente desarrollada por *Thomas Trojer*<sup>2</sup>) para utilizar el módulo UWB. Una vez realizado este análisis, se optó por descartar la versión que usaba Localino y migrar a la última versión de la biblioteca de *Thomas Trojer* (*arduino-dw1000*).

En este cambio se encontraron varios errores solucionados y nuevas funcionalidades, como por ejemplo poner explícitamente el identificador del dispositivo al iniciar el protocolo de cálculo de distancias con que ya cuenta la biblioteca. Igualmente persistían algunos problemas de pérdida de conexión entre los dispositivos. Básicamente, el *anchor* luego de terminar el protocolo de cálculo de distancias reporta el resultado al *tag*, pero este último no espera lo suficiente y pierde el mensaje, por lo que el *tag* no envía la información al servidor. En este punto, el equipo empezó a analizar el funcionamiento de esta biblioteca íntegramente para entender por qué se generaban estos problemas, lo que entretuvo mucho el proceso de desarrollo ya que el código es muy poco “amigable”. Además, hubo que analizar procesos que ocurren a la vez en un dispositivo y en la contraparte y que tienen tiempos de acción muy cortos. A su vez, el código encargado de todo el proceso de estimación está implementado en una sola clase monolítica, es decir, el *tag* y el *anchor* ejecutan un mismo método que, dependiendo de si es uno u otro, cambia su flujo de ejecución y por lo tanto su comportamiento.

Al final se encontraron las fallas en el protocolo. Dicho protocolo utiliza la técnica de acceso múltiple por división de tiempo, también llamado en inglés Time Division Multiple Access (TDMA), donde el *tag* actúa como coordinador de acceso al medio asignando un espacio de tiempo a cada *anchor* para que responda y luego espera por la respuesta del último *anchor* para pasar a la siguiente fase del protocolo de estimación. El problema se daba porque podían responder todos los *anchors*, pero no el último (o demorarse mucho), y el algoritmo descartaba todo el ciclo de estimación a raíz de un tiempo de espera, reiniciando el proceso y perdiendo así el *tag* la posibilidad de obtener la distancia de aquellos *anchors* de los que sí logró obtener respuesta. Dado este hallazgo, se hicieron modificaciones para que la ejecución siguiera con los dispositivos que respondieron y también modificaciones en los valores de los tiempos de espera entre los cambios de estado.

---

<sup>2</sup><https://github.com/thotro/arduino-dw1000>

Durante el desarrollo, en algunos momentos se notaba que los dispositivos quedaban bloqueados, sobre todo en los procesos de calibración del *antenna delay*, por lo que se investigó la posibilidad de poner disparadores de bloqueos (también conocidos como *Watchdog*). La versión de las bibliotecas nativas enviada por el fabricante del dispositivo para trabajar con el microprocesador *STM32* no posee esta funcionalidad, por lo que se optó por cambiar la versión de estas bibliotecas a una más nueva. Esto llevó a que se hicieran cambios en la manera de subir el *firmware* a los dispositivos instalando software propio del fabricante del microprocesador (algo que no era necesario con la versión anterior) y actualizando el *firmware* del adaptador ST-linkV2 que se utiliza para subir el *firmware* a los dispositivos. Además, se actualizó el IDE de Arduino en el que se trabajaba, para poder soportar la nueva versión de las bibliotecas nativas.

Con todas estas actualizaciones, sumadas a las mejoras en la biblioteca *arduino-dw1000*, se puede decir que el sistema está al día en cuanto al software, *firmware* base y ambiente de desarrollo, dado que esta migración no tuvo efectos sobre el sistema que ya estaba funcionando. Esto es altamente positivo a la hora de evaluar la viabilidad de continuar con el proyecto en un futuro y que este evolucione.

## Apéndice D

# Proceso de calibración del parámetro Antenna Delay

El parámetro *antenna delay* es un parámetro que se encuentra dentro de los módulos UWB de Decawave y es parte fundamental en el cálculo de distancias entre los dispositivos. Es el tiempo que tarda el dispositivo en transmitir el mensaje al medio. Dentro de este tiempo están las demoras introducidas por la antena y demoras internas del módulo DW1000, del PCB donde la antena está emplazada y de componentes externos.

En su documentación, Decawave muestra un mecanismo para poder calibrar este parámetro. Para este proceso de calibración, es necesario tener dos dispositivos separados a una distancia fija cuyo valor depende de otros parámetros según el uso que se le quiera dar a los dispositivos. Estos parámetros son la frecuencia de la señal con la que se va a trabajar (16 Mhz en el caso del proyecto) y el canal de transmisión (canal 5). Esta configuración es la ideal para aplicar en el cálculo de distancias y para minimizar el efecto de los obstáculos en el ambiente. Según estos parámetros y la tabla provista en el manual, la distancia a utilizar entre los dispositivos para la calibración del *antenna delay* debe ser de 7.94 m.

Los actores de este proceso serán el receptor (o *anchor*) y el transmisor (o *tag*). A este último se le pone como *antenna delay* el valor cero, ya que el receptor es quien cambiará este valor en el transcurso de la calibración.

El algoritmo propuesto por Decawave es el siguiente (Decawave, 2017, sección 8.3.1):

- Colocar los dispositivos según la distancia especificada.
- Usando el protocolo *Two-way Ranging* (sec. 3.3.2.1), tomar 1000 medidas de distancia.
- Ajustar el valor de *antenna delay* en función de si el promedio de las distancias se aproxima a la distancia del punto uno o se aleja; la condición de parada es que este promedio esté lo más cercano posible a la distancia original.

Sin embargo, este algoritmo no tiene una condición de parada clara, ya que cuando dice “lo más cercano posible a la distancia original” depende del criterio elegido por quienes hagan ese procedimiento, por lo que claramente no es automatizable.

En función de este algoritmo se diseñó un programa que permite cambiar el valor de *antenna delay*

dinámicamente y a su vez parametrizar la condición de parada. Este procedimiento se ejecuta en una computadora. Con él, usando tópicos de ROS, se cambia el valor de *antenna delay* en el receptor y se intercambia información con dicho receptor, a medida que se realiza el proceso de calibración. El programa aplica búsqueda binaria para encontrar el o los valores de *antenna delay* posibles. Al experimentar con el calibrado, se encontró que el valor óptimo no es único, por lo que la condición de parada no se cumple al encontrar un valor de distancia (o el primero) aceptable, sino que dicha condición se alcanza cuando los rangos máximo y mínimo de búsqueda se crucen, ya que esto marca que se cubrió todo el rango de valores posibles. Luego, con los valores óptimos de *antenna delay* y las distancias que estos generan, se decide cuál valor es el que se usará en los dispositivos.

A continuación se presenta un pseudocódigo del programa que se mencionó anteriormente.

---

```

1  LECTURAS = 1000
2  DISTANCIA_FINAL = 7.94
3  DELTA = 0.05
4  tope_max = 34000
5  tope_min = 31501
6
7  mientras (tope_max ≥ tope_min):
8      valor_antenna_delay = (tope_max + tope_min) div 2
9
10     #Se cambia el valor de Antenna Delay del dispositivo por el nuevo valor.
11     cambiar_antenna_delay_dispositivo(valor_antenna_delay)
12
13     #distancias es un vector que tiene las lecturas que se tomaron
14     distancias = obtener_lecturas(LECTURAS)
15
16     #se calcula la mediana de los valores almacenados en distancias
17     mediana = calcular_mediana(distancias)
18
19     #Si la mediana esta dentro del rango DISTANCIA_FINAL +/- DELTA
20     #es un valor aceptable de Antenna Delay
21     si ((DISTANCIA_FINAL - DELTA) < mediana < (DISTANCIA_FINAL + DELTA)):
22         imprimir "Se encontro un Antenna Delay aceptable: ", valor_antenna_delay
23         imprimir "distancia: ", mediana
24     fin
25
26     #Se calcula el nuevo rango
27     si (mediana < DISTANCIA_FINAL):
28         tope_max = val_med - 1
29         tope_min = tope_min
30     sino:
31         tope_max = tope_max
32         tope_min = val_med + 1
33     fin
34 fin

```

---

**Algorithm D.1:** Pseudocódigo del proceso de calibración desarrollado para el proyecto.

En el pseudocódigo se usaron funciones que representan el proceso que tienen de nombre y que no se detallan porque no son relevantes para describir cómo funciona el proceso de calibración.

Este algoritmo creado tiene como debilidad el tiempo que toma obtener los valores: por ejemplo, para realizar 500 mediciones por rango de valores, el tiempo total del proceso fue de 45 minutos. Sin embargo, el proceso podría ser más lento aún si la elección de los valores de muestreo de *antenna delay* fuese lineal, como se plantea en el algoritmo propuesto por Decawave.

## Apéndice E

# Efecto de los obstáculos en la precisión del sistema (aplicando filtros)

En este experimento se utilizaron filtros existentes en el estimador de posición del agente robótico, que sirven para ajustar la salida del algoritmo de cálculo de la posición y “mejorarla”. El objetivo es relevar si existe algún incremento de la precisión al aplicar filtros respecto a la salida sin filtro.

- Filtro de Kalman: Es un algoritmo desarrollado por Rudolf E. Kalman en 1960 que sirve para poder identificar el estado oculto (no medible) de un sistema dinámico lineal.
- Filtro  $\alpha$ - $\beta$ : También llamado *filtro gh*, es la forma más simple de un filtro que combina observación y estimación.
- Promedio: Este filtro se basa en promediar las últimas cuatro salidas para obtener la posición.
- Sin filtro: No se aplica filtro alguno al resultado de posicionamiento.

En la siguiente tabla se muestran los errores según los filtros aplicados en los distintos escenarios.

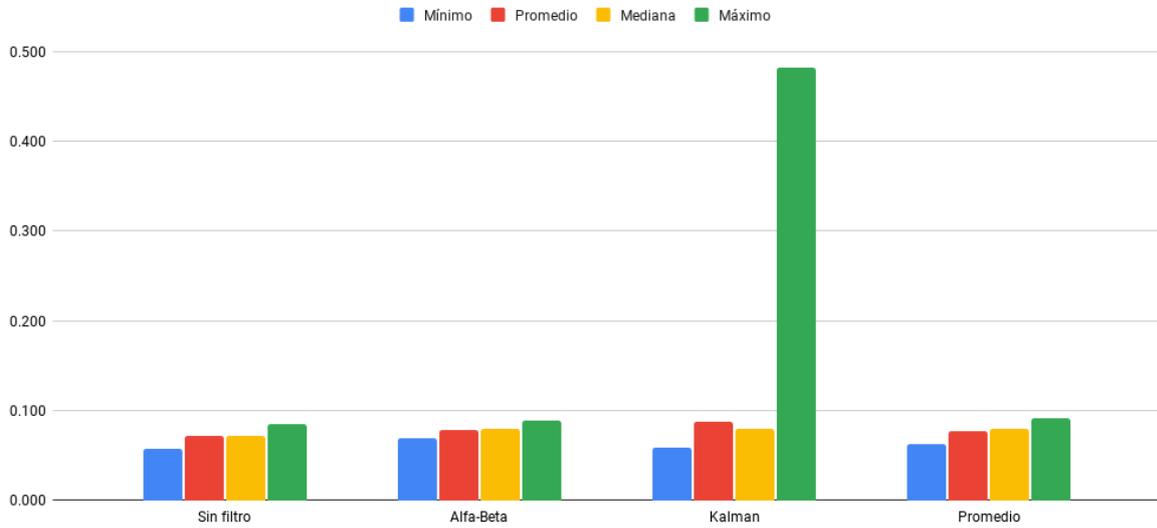
Filtro	Promedio	Mediana	Mínimo	Máximo	Desviación estándar
Sin filtro	0.071	0.072	0.057	0.084	0.00600
Alfa-Beta	0.078	0.079	0.069	0.088	0.00526
Kalman	0.087	0.079	0.058	0.482	0.04451
Promedio	0.077	0.079	0.062	0.091	0.00638

**Tabla E.1:** Comparación del error según filtro aplicado a la estimación.

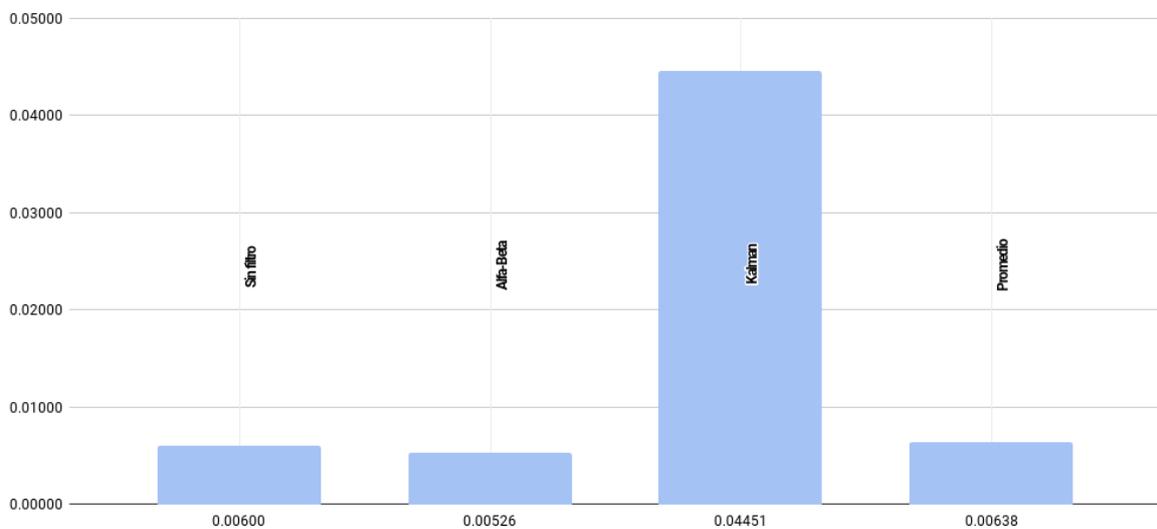
En el gráfico de la figura E.1 se puede apreciar que los valores resultado de la aplicación de filtros se comportan de forma similar, a excepción del máximo en el filtro de Kalman. El equipo cree que esto último se debe a que en los primeros valores el filtro aún no se había inicializado y por esa razón se dieron valores muy alejados de la posición original.

Si bien la ventaja en la aplicación de esta clase de filtros es más apreciable en la experimentación con dispositivos en movimiento donde la estimaciones son “suavizadas”, el equipo estimó que podría ser interesante contar con una comparativa de precisión entre los distintos filtros. El gráfico comparativo de

la desviación estándar se muestra en la figura E.2.



**Figura E.1:** Gráfica de error según escenario, aplicando los filtros en la estimación del punto.



**Figura E.2:** Gráfica de error según escenario, aplicando el promedio de los últimos cuatro valores en la estimación del punto.

En resumen, sobre la base de estimaciones con dispositivos estáticos, es posible ver que con los parámetros utilizados en el filtro de Kalman, este es el que presenta mayor dispersión en sus resultados, probablemente debido al valor máximo tan elevado. Mientras tanto, el filtro  $\alpha$ - $\beta$  es el que presenta mayor estabilidad de todos, con mínimos y máximos cercanos al promedio y la mediana de errores.

# Glosario

## A

### anchor

Es el dispositivo que está estático, tiene posición conocida y es usado como referencia por el *tag* para poder localizarse. 1–3, 21, 22, 24–29, 31–36, 38, 39, 41, 43, 46, 48–53, 59, 60, 62

### angulación

Es un algoritmo de localización que utiliza propiedades geométricas de los triángulos para realizar la estimación de la posición de un objeto. Por más detalles, ver sec. 3.3.3. 3, 16, 17

### antenna delay

Es el tiempo que el hardware demora en transmitir la señal al medio. Esta demora depende de la antena *UWB* y del hardware en la que está emplazada. 23, 50, 52, 59–63

## Arduino

Es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. 19, 21, 24, 59, 61

## B

### banda ultra-ancha

El término *banda ultra-ancha*, *Ultra-WideBand (UWB)* en inglés, se usa para hacer referencia a cualquier tecnología de radio que utiliza un ancho de banda mayor de 500 MHz o del 25% de la frecuencia central. Puede usar frecuencias que van desde 3.1 GHz hasta 10.6 GHz. 1, 20

### bluetooth

Es una tecnología de radiofrecuencia de corto alcance, que define un estándar para las redes *WPAN* (*Wireless Personal Area Network*) utilizando señales de radio UHF de onda corta en el espectro de banda de 2.400 a 2.485 GHz y genera Personal Area Networks. El estándar más nuevo disponible es el llamado *BLE*, por *Bluetooth Low Energy* (también llamado *Smart Bluetooth*), que está especialmente diseñado para tener el menor consumo de energía posible. 3, 7, 8, 55

## D

### distancia euclídea

En matemática aplicada a la geometría, se trata de una función no negativa usada en diversos contextos para calcular la distancia entre dos puntos, en el plano o en espacios de tres o más dimensiones. 18, 34, 38

### Docker

Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. 1, 31, 50

## F

### filtro $\alpha$ - $\beta$

El filtro  $\alpha$ - $\beta$ , también llamado filtro  $gh$ , es la forma más simple de un filtro que combina observación y estimación. 64, 65

### filtro de Kalman

Es un algoritmo desarrollado por Rudolf E. Kalman en 1960, que sirve para poder identificar el estado oculto (no medible) de un sistema dinámico lineal. 52, 64, 65

### fingerprinting

Se trata de un caso particular de la técnica de localización denominada *Reconocimiento de patrones*, donde se utilizan valores físicos, no visuales y medibles del ambiente, para estimar la posición. Por más detalles, ver sec. 3.3.5.1. 3

### firmware

Es un programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. 21, 59, 61

### framework

Un entorno de trabajo (del inglés framework), o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.. 19

## G

### ground-truth

Término utilizado para referirse a la medición más precisa disponible, que puede ser usado para cuantificar las mediciones de nuevos métodos, algoritmos y sensores a través de su comparación..

49

## H

### HaLow

Es un protocolo de red inalámbrica publicado en 2017 como una enmienda del estándar de red inalámbrica IEEE 802.11-2007. Utiliza bandas exentas de licencia de 900 MHz para proporcionar redes WiFi de rango extendido, en comparación con las redes WiFi convencionales que operan en las bandas de 2,4 GHz y 5 GHz. También se beneficia de un menor consumo de energía, lo que permite la creación de grandes grupos de estaciones o sensores que cooperan para compartir señales, apoyando el concepto de *Internet of Things*. 8

## I

### Internet of Things

*Internet of Things*, *Internet de las Cosas* es español, es un concepto que refiere a una interconexión digital de objetos cotidianos con internet. 68, 69

## L

### lateración

Es un algoritmo de localización, en el que el receptor conoce el rango de la señal o diferencias de rango de señal entre él y al menos tres nodos emisores. Estos datos son usados en una serie de  $n$  ecuaciones no lineales para calcular la posición del nodo receptor, donde  $n$  es la cantidad de los nodos receptores. Para  $n=3$ , el método se llama *trilateración*. 10, 11, 24, 70

## M

### middleware

Middleware o lógica de intercambio de información entre aplicaciones (interlogical), o Agente Intermedio, es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware o sistemas operativos. Este simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones y sincronizaciones que son necesarias en los sistemas distribuidos. De esta forma, se provee una solución que mejora la calidad de servicio, así como la seguridad, el envío de mensajes, la actualización del directorio de servicio, etc.. 19

## O

### odometría

En robótica, la *odometría* es el proceso de determinar la posición y orientación de un robot utilizando solamente la medición de la rotación de sus ruedas, a partir de una posición inicial. 17

### odometría visual

En robótica y visión por computadora, la odometría visual es el proceso de determinar la posición y orientación de un robot mediante el análisis de secuencias de imágenes relacionadas, obtenidas

con una cámara. 17

### **open source**

Se califica como *open source* a los programas informáticos que permiten el acceso a su código de programación, lo que facilita modificaciones por parte de otros programadores ajenos a los creadores originales del software en cuestión. 20, 21

## **P**

### **Personal Area Network**

Red de área personal es un estándar de red para la comunicación entre distintos dispositivos (computadoras, puntos de acceso a internet, teléfonos celulares, PDA, dispositivos de audio, impresoras) cercanos al punto de acceso. 66

### **propagación multicamino**

La propagación multicamino, también conocida como *Multipath Propagation*, es un error muy común que afecta la amplitud y la fase de la señal. Se genera por reflejos de la señal al ser emitida, ya que se obtienen distintas "copias" de esta. Por lo general, genera errores en las técnicas que determinan la distancia del emisor por medio de diferencias de tiempo. 2, 4, 15, 20, 48, 49, 56

## **R**

### **raíz cuadrada de la media**

En matemática, la raíz cuadrada de la media (abreviada RMS o rms) es una medida estadística de la magnitud de una cantidad variable. También se la conoce como la media cuadrática. 3, 36, 37

## **ROS**

ROS (Robot Operating System) provee bibliotecas y herramientas para ayudar a los desarrolladores de software a crear aplicaciones para robots. También provee abstracción de hardware, controladores de dispositivos, bibliotecas, herramientas de visualización, comunicación por mensajes, administración de paquetes, entre otros. Está bajo la licencia open source, BSD. 19, 23, 24, 28, 29, 31, 38, 49, 53

## **S**

### **subgigahertz**

Se refiere a las ondas del espectro que están por debajo de 1 GHz. Son frecuencias en auge por el aumento de las aplicaciones de *Internet of Things*. 8

## **T**

## **tag**

Hace referencia al dispositivo que intenta localizarse y está en movimiento. 2–4, 21–34, 36, 38, 41–43, 46, 48, 50–52, 59, 60, 62, 66

## **trilateración**

Es un caso particular de lateración, donde la cantidad de nodos emisores (de referencia) es igual a tres. Por más detalles, ver sec. 3.3.2. 10

## **U**

### **unidad de medición inercial**

Es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giroscopios. 17

## **W**

### **watchdog**

Es un mecanismo de seguridad que provoca un reinicio del sistema en caso de que este se haya bloqueado. 61

## **WiFi**

Es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con *WiFi* pueden conectarse entre sí o a internet a través de un punto de acceso de red inalámbrica. *WiFi* es una marca de la *Alianza Wi-Fi*, la organización comercial que cumple con los estándares 802.11 relacionados con redes inalámbricas de área local. 3, 8, 10, 20, 21, 49, 58, 59

# Acrónimos

## A

### ADS-TWR

Asymmetric Double-sided Two-way Ranging. 14

### AoA

Angle of Arrival. 16

## B

### BLE

Bluetooth Low Energy. 3, 7–9, 66

## C

### CoO

Cell of Origin. 10

## D

### DoA

Direction of Arrival. 16

### DS-TWR

Double-sided Two-way Ranging. 14, 51

## F

## **FM**

Frecuencia Modulada. 3

## **G**

### **GPS**

Global Positioning System. 9, 11, 17

### **GSM**

Global System for Mobile communications. 18

## **I**

### **IFSR**

Indicador de Fuerza de la Señal Recibida. 4, 15

### **IMU**

Inertial Measurement Unit. 3, 17, 18

### **IoT**

Internet of Things. 4, 8

### **IR**

Infrared Radiation. 6

### **ISM**

Industrial, Scientific and Medical. 8

## **L**

### **LED**

Light Emitting Diode. 9

### **LoRa**

Long Range Radio. 4, 55

### **LoS**

Line of Sighth. 6, 15, 39, 55, 56

## **M**

### **MIMO-OFDM**

Multiple-Input, Multiple-Output Orthogonal Frequency-Division Multiplexing. 8

## **N**

### **NLoS**

Non Line of Sight. 15, 20, 38, 39, 52, 53, 56, 60

## **P**

### **PAN**

Personal Area Network. 6

### **PCB**

Printed Circuit Board. 62

### **PoA**

Phase of Arrival. 14, 15

## **R**

### **RF**

Radiofrecuencia. 6

### **RFID**

Radio Frequency Identification. 3, 7, 8, 17, 55

### **RSS**

Received Signal Strength. 15, 56

### **RSSI**

Received Signal Strength Indicator. 15

### **RToF**

Round-Trip Time of Flight. 14, 28

## **RTT**

Round-Trip Time. 14

## **S**

### **SDS-TWR**

Symmetric Double-sided Two-way Ranging. 14

## **T**

### **TDMA**

Time Division Multiple Access. 60

### **TDoA**

Time Difference of Arrival. 13

### **ToA**

Time of Arrival. 12, 24, 51

### **ToF**

Time of Flight. 12

### **TW-ToA**

Two-way Time of Arrival. 14

### **TWR**

Two-way Ranging. 3, 14, 24, 25, 28, 62

## **U**

### **UHF**

Ultra High Frequency. 7, 66

### **UWB**

Ultra-wideband. 2, 5–7, 20, 21, 24, 25, 28, 39, 42, 49, 52, 55, 58–60, 62

## **V**

**VLC**

Visual Light Communication. 9

**W****WAP**

Wireless Access Point. 1

**WLAN**

Wireless Local Area Network. 7, 8

**WPAN**

Wireless Personal Area Network. 7, 66

# Bibliografía

- Acevedo, F., Coduri, G., y Perera, G. (2018). CattleNet.
- Ahson, S. y Ilyas, M. (2011). *Location-based services handbook*. CRC Press.
- Decawave (2017). *DW1000 User Manual*. [https://www.decawave.com/sites/default/files/resources/dw1000\\_user\\_manual\\_2.11.pdf](https://www.decawave.com/sites/default/files/resources/dw1000_user_manual_2.11.pdf), Accedido: 13-02-2020.
- Decawave (2018a). *Maximising range in DW1000 based systems*. [https://www.decawave.com/wp-content/uploads/2018/10/APS017\\_Max-Range-in-DW1000-Systems\\_v1.1.pdf](https://www.decawave.com/wp-content/uploads/2018/10/APS017_Max-Range-in-DW1000-Systems_v1.1.pdf), Accedido: 13-02-2020.
- Decawave (2018b). *Non line of sight operation and optimizations to improve performance in DW1000 based systems*. [https://www.decawave.com/wp-content/uploads/2018/10/APS006\\_Part-2-NLOS-Operation-and-Optimizations\\_v1.5.pdf](https://www.decawave.com/wp-content/uploads/2018/10/APS006_Part-2-NLOS-Operation-and-Optimizations_v1.5.pdf), Accedido: 13-02-2020.
- DiGiampaolo, E. y Martinelli, F. (2014). Mobile Robot Localization Using the Phase of Passive UHF RFID Signals. *IEEE Transactions on Industrial Electronics*, 61(1):365–376.
- DiGiampaolo, E. y Martinelli, F. (2018). A robotic system for localization of passive UHF-RFID tagged objects on shelves. *IEEE Sensors Journal*, pages 1–1.
- Farid, Z., Nordin, R., y Ismail, M. (2013). Recent Advances in Wireless Indoor Localization Techniques and System. *Journal of Computer Networks and Communications*, 2013:1–12.
- Furquez, N. y Saldías, O. (2019). Documento de estado del arte. Technical report, Facultad de Ingeniería, UdeLaR.
- Gu, Y., Fan, J., Wu, M., y Huang, X. (2016). Indoor mobile robot positioning based on wireless fingerprint matching. *International Journal of Computer and Information Engineering*, Vol:10,(6).
- Gualda, D., Ureña, J., García, J. C., García, E., y Alcalá, J. (2019). Simultaneous calibration and navigation (scan) of multiple ultrasonic local positioning systems. *Information Fusion*, 45:53 – 65.
- IEEE (2016a). Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 2: Sub 1 ghz license exempt operation. *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pages 1–594.
- IEEE (2016b). Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534.
- Kaemarungsi, K. y Krishnamurthy, P. (2004). Modeling of indoor positioning systems based on location fingerprinting. In *IEEE INFOCOM 2004*, volume 2, pages 1012–1022 vol.2.

- Kwak, M. y Chong, J. (2010). A new double two-way ranging algorithm for ranging system. In *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, pages 470–473.
- Küpper, A. (2005). *Location-Based Services: Fundamentals and Operation*. John Wiley and Sons.
- Moghtadaiee, V., Dempster, A. G., y Lim, S. (2011). Indoor localization using FM radio signals: A fingerprinting approach. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7.
- Munishwar, V. P., Singh, S., Mitchell, C., Xiaoshuang Wang, Gopalan, K., y Abu-Ghazaleh, N. B. (2009). RFID based localization for a miniaturized robotic platform for wireless protocols evaluation. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–3, Galveston, TX, USA. IEEE.
- Pu, Y.-C. y You, P.-C. (2018). Indoor positioning system based on BLE location fingerprinting with classification approach. *Applied Mathematical Modelling*, 62:654–663.
- Trinidad, G. (2020). Dispositivo de asistencia ante el congelamiento de la marcha en pacientes con parkinson. Accepted: 2021-02-02T17:59:46Z Publisher: Udelar.FI.
- Vargas, C. V. y López, W. E. (2007). Sistemas de comunicación inalámbrica MIMO - OFDM. *Acta Nova*, 3(4):750–760.
- Yavari, M. y Nickerson, B. G. (2014). Ultra wideband wireless positioning systems. *Dept. Faculty Comput. Sci., Univ. New Brunswick, Fredericton, NB, Canada, Tech. Rep. TR14-230*.
- Ye-Sheng, K., Pannuto, P., Ko-Jen, H., y Dutta, P. (2014). Luxapose: Indoor positioning with mobile phones and visible light. *MobiCom: International Conference on Mobile Computing and Networking*, pages 447 – 458.
- Zafari, F., Gkelias, A., y Leung, K. (2017). A Survey of Indoor Localization Systems and Technologies. *arXiv:1709.01015 [cs]*. arXiv: 1709.01015.

## Apéndice F

# Agradecimientos

De las primeras personas a los que deseamos agradecer es a nuestras familias y amigos, que nos han dado el apoyo necesario para poder seguir adelante en la carrera y no abandonar en esos momentos en los la cosa se ponía difícil. Especialmente agradecer a nuestras parejas por el aguante y el apoyo incondicional en los meses que transcurrió el proyecto. También agradecer a esos amigos y compañeros de carrera con los que crecimos a la par y que compartimos largas horas de biblioteca, laboratorios o entregas hasta altas horas de la madrugada.

Especialmente queremos dar las gracias a Mariana por el tiempo que le dedico a la corrección de la tesis.

Agradecer el apoyo de quienes nos acompañaron desde un principio con el proyecto y lo vieron crecer y mutar, ellos son Gonzalo Tejera y Lucas Micol, que con su guía este proyecto no hubiese sido posible y por confiar en nosotros para el desarrollo de este.

Agradecemos también la ayuda por parte del Instituto de Agrimensura en especial al Ingeniero Agrimensor Gabriel Barreiro por su gran ayuda en el relevamiento de los puntos en el laboratorio. También a la Ingeniera Agrimensora Ana Fernández por la ayuda en dudas puntuales en el área de localización.