


# Posicionamiento interno UWB para robótica cooperativa

## UWB Internal Positioning for Cooperative Robotics

E. Berra Villaseñor\*   
Universidad Anáhuac Puebla, Club Orión Norte S/N La Vista  
Country, San Andrés Cholula Puebla, C.P. 72810, Puebla, México

**Resumen.** La creciente demanda de automatización e inteligencia artificial ha impulsado el desarrollo de tecnologías de rastreo preciso. Tecnologías como el IoT y Big Data son esenciales para la inteligencia artificial que facilita la toma de decisiones. Sin embargo, la Revolución Industrial 4.0 exige soluciones de posicionamiento interior más precisas. La tecnología de banda ultra ancha (UWB) supera las limitaciones de otras tecnologías en interiores. Este artículo propone un modelo con balizas y triangulación de señales UWB para rastrear robots cooperativos. Se desarrolló un sistema con microcontroladores ESP32 y circuitos BU01, controlado por WiFi y visualizado mediante una GUI.

**Palabras Claves.** Posicionamiento; triangulación, UWB (Banda Ultra Ancha); robótica; automatización.

**Abstract.** The growing demand for automation and artificial intelligence has driven the development of precise tracking technologies. Technologies like IoT and Big Data are essential for artificial intelligence that facilitates decision-making. However, the Fourth Industrial Revolution demands more accurate indoor positioning solutions. Ultra-wideband (UWB) technology surpasses the limitations of other indoor technologies. This article proposes a model using beacons and UWB signal triangulation to track cooperative robots. A system was developed with ESP32 microcontrollers and BU01 circuits, controlled via WiFi and visualized through a GUI.

**Keywords.** Positioning; triangulation; UWB (Ultra-Wideband); robotics; automation.

**Como citar.** E. Berra Villaseñor, Posicionamiento interno UWB para robótica cooperativa. *Jou. Cie. Ing.*, vol. 16, no. 1, pp. 37-48, 2024. doi:10.46571/JCI.2024.1.6

**Recibido:** 22/04/2024    **Revisado:** 10/06/2024    **Aceptado:** 30/06/2024

### 1. Introducción

La demanda creciente en diversas industrias de automatización y aplicación de inteligencia artificial en la era hiperconectada actual marcada por la proliferación de dispositivos móviles y tecnologías que permiten el rastreo preciso de objetos [1], productos e incluso individuos en entornos complejos como almacenes, centros comerciales y hogares, ha experimentado un

\* e-mail: [eduardo.berra@anahuac.edu.mx](mailto:eduardo.berra@anahuac.edu.mx)

aumento exponencial. Esto ha convertido a la investigación y desarrollo en Internet de las cosas (IoT) y Big Data en prioridades cada vez más apremiantes [2]. Estas tecnologías son los pilares para la generación de inteligencia artificial, que a su vez facilita la toma de decisiones y la predicción de patrones. Es imperativo contar con soluciones efectivas que demuestren competitividad y eficiencia a costos accesibles.

Las tecnologías como los sistemas de posicionamiento global por satélite (GNSS) existen desde hace décadas [3], con el transcurso del tiempo, han sido objeto de estructuración y reestructuración, alcanzando un nivel de precisión adecuado para su uso en exteriores, ejemplos destacados de esta adaptación son los dispositivos GPS integrados en relojes inteligentes y los sistemas de navegación automotriz[4], que han sido ampliamente comercializados y son altamente efectivos.

Sin embargo, los requisitos de la Revolución Industrial 4.0, que está siendo experimentada por diversas áreas de la industria, demandan un nivel de posicionamiento más allá del alcance global en exteriores[4], el posicionamiento interno necesario abarca una amplia gama de aplicaciones y cubre diversas necesidades, desde el seguimiento de las rutas de los compradores en las tiendas hasta la ubicación de la robótica en almacenes o la gestión de la mercancía. Estos datos son invaluable, ya que proporcionan información crucial que puede ser utilizada para tomar decisiones informadas[4] y desarrollar soluciones adaptadas a las demandas de entornos controlados.

Algunas de las soluciones desarrolladas presentan diversas propuestas, entre las cuales se incluye el uso de triangulación de señales de diferentes tipos, como ultrasonido, WiFi, Bluetooth, entre otras [5]. Sin embargo, todas estas tecnologías tienen una limitación: las estructuras metálicas, como estantes, techos industriales de metal o paredes falsas, suelen obstaculizar la recepción clara de la señal, esto significa que el entorno de comunicación inalámbrica no garantiza un rendimiento óptimo de recepción[4].

La tecnología de banda ultra ancha (UWB) nos ofrece la capacidad de localizar objetos en entornos interiores complejos, caracterizados por múltiples rutas. Sus ventajas incluyen una alta precisión, incluso en presencia de interferencias significativas [6], lo que asegura la recepción precisa de la posición en todo momento.

Ante la creciente demanda de posicionamiento interior, se propone un modelo que utiliza balizas y triangulación de señales UWB. Este modelo se implementará para su aplicación en robótica cooperativa [7], con el fin de proporcionar un seguimiento preciso de la posición de las unidades robóticas que interactúan en un entorno definido. Esta información puede ser aprovechada posteriormente para permitir que los robots realicen sus tareas de manera autónoma, sin necesidad de intervención humana.

## 2. Metodología

El seguimiento y posicionamiento de dispositivos emisores-receptores de banda ultra ancha pueden realizarse a través de algoritmos de triangulación. Esta técnica se emplea en entornos interiores/exteriores y se basa en el uso de anclajes fijos [8], que actúan como puntos de referencia estáticos. Estos anclajes son la base necesaria para formular las ecuaciones matemáticas que sustentan el método de triangulación.

En el marco de investigación de esta técnica, se ha propuesto un diseño de anclajes que combina el circuito BU01 con un microcontrolador ESP32, el cual ofrece la capacidad de reconfiguración mediante WiFi. Esta característica resulta de suma importancia en entornos *indoor*, donde puede ser necesario ajustar los anclajes, debido a que estos ajustes permiten a los robots detectar señales codificadas [8], que facilitan el inicio del proceso de triangulación y la determinación de la ubicación espacial (Institute of Electrical and Electronics Engineers, n.d.). A partir de esta información, los robots pueden calcular y ajustar automáticamente sus rutas y movimientos.

Como se puede observar en el diagrama de bloques de la Figura 1.1, los anclajes presentan una conexión de procesamiento dual entre el microcontrolador BU01 y el ESP32, estos componentes están alimentados por una batería LiPo, complementada con un cargador USB.

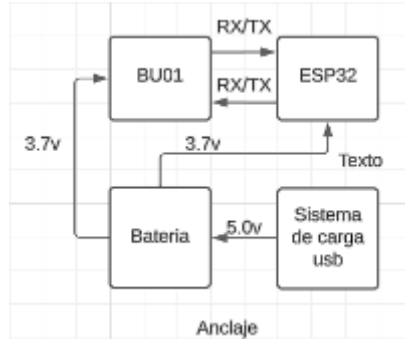


Figura 1: Diagrama de bloques anclaje.

La propuesta para el receptor en el robot móvil presenta una estructura similar a la de los anclajes, con la diferencia principal de incluir una conexión I2C. Gracias a esta conexión, la información preprocesada de la señal se transfiere a la microcomputadora que controla los movimientos y la dirección de la unidad [9], el diagrama de bloques de la figura 2 representa esta configuración.

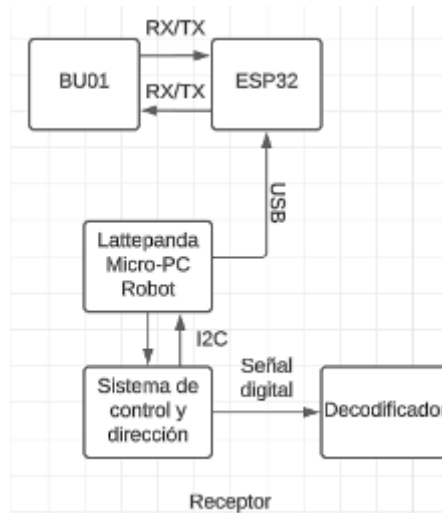


Figura 2: Receptor robot móvil.

El sistema de robot móvil desarrollado tiene como objetivo la detección de plagas bajo la supervisión y control remoto de un operador. Este sistema incluye dos motores que son controlados mediante señales de radiofrecuencia (RF) y una unidad cluster encargada de gestionar diversos sensores. Además, está equipado con una computadora LattePanda que opera bajo el sistema Windows 10, la cual maneja una cámara de alta resolución con capacidad de zoom. El diseño del robot se ilustra en la Figura 3

La zona en la que el robot desempeñará sus funciones de manera autónoma abarca una superficie de 44 metros cuadrados, tal como se ilustra en el plano de la Figura 4. Este plano detalla la ubicación de los anclajes, así como la del receptor instalado en el robot móvil.



Figura 3: Robot Móvil.

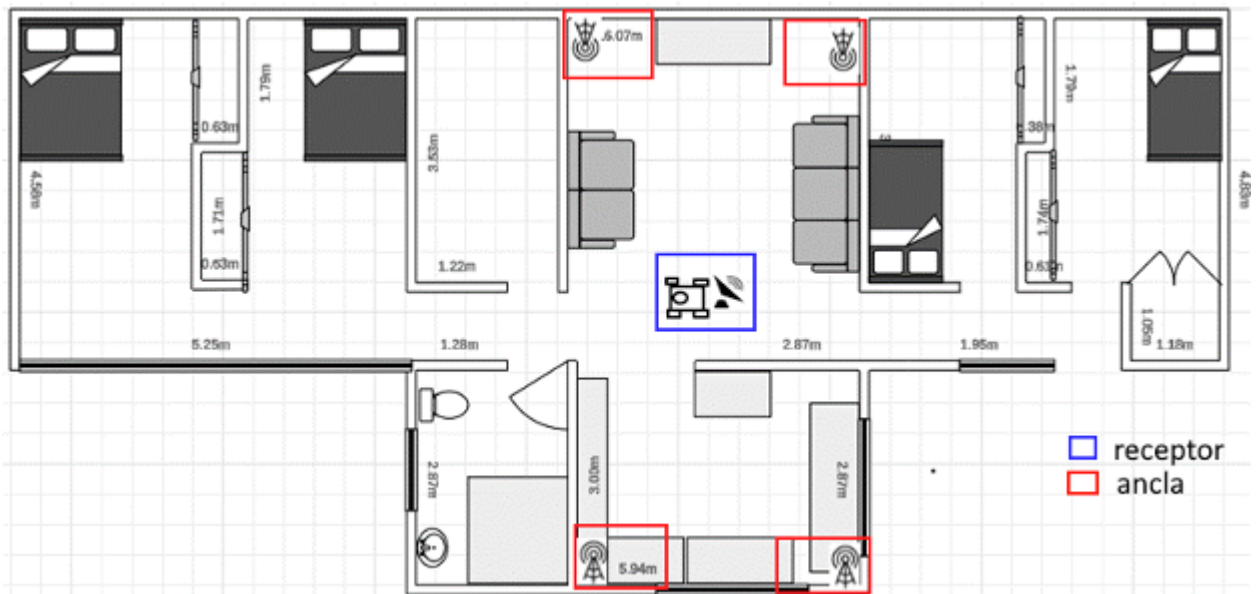


Figura 4: Plano de ubicación de anclajes y receptor.

El método de triangulación desarrollado con algoritmos en MicroPython [10] consta de cuatro anclas y un receptor móvil. Las ecuaciones para la triangulación se basan en el siguiente conjunto matemático de vectores, definidos por las anclas situadas en las esquinas del área a cubrir:

$$d^1 = (x + x_1)^2 + (x + y_1)^2$$

$$d^2 = (x + x_2)^2 + (x + y_2)^2$$

$$d^3 = (x + x_3)^2 + (x + y_3)^2$$

$$d^4 = (x + x_4)^2 + (x + y_4)^2$$

$$m_1 = X_m, Y_m$$

La figura 1.5 muestra la relación de las ecuaciones con la posición de las anclas y el dispositivo móvil.

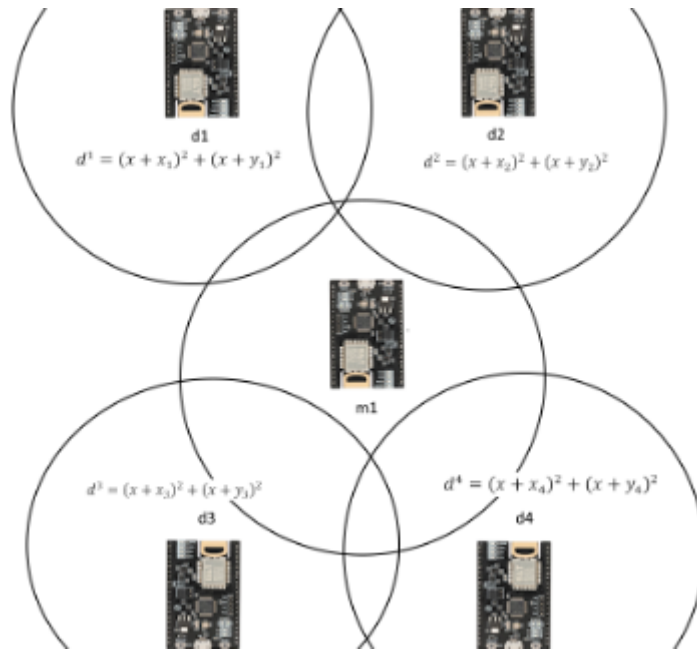


Figura 5: Ecuaciones para triangulación de posición.

El método de triangulación se realiza mediante el Tiempo de Llegada (ToA, "Time of Arrival"), que proporciona información sobre la distancia entre el ancla y el móvil [11]. Al medir el tiempo de transmisión y el tiempo de recepción de la comunicación entre el móvil y el ancla, se puede calcular la distancia como la diferencia entre ambas mediciones. Sin embargo, un inconveniente de este método es que, a menos que se conozcan las características de la señal [12], el receptor puede tener dificultades para determinar con precisión el tiempo de llegada en condiciones de señal débil.

El componente UWB-BU01 aborda este problema, ya que permite lograr altas relaciones señal-ruido (SNR) y alta relación señal-interferencia, además de limitar la distorsión por trayectos múltiples en todos los receptores. Gracias a esta tecnología, es posible mejorar la exactitud del método utilizando información proporcionada por señales moduladas digitalmente[6], según lo especificado por el proveedor del circuito "Decawave", como se muestra en la figura 6

Para este desarrollo, se combina la tarjeta UWB01 con la tarjeta ESP32, como se muestra en la figura 7, para utilizar los datos a través de WiFi y generar aplic[12]aciones de control para el robot móvil. La prueba se centra en la ubicación del robot, que será controlado manualmente a distancia. En trabajos posteriores, el robot será guiado para realizar tareas preprogramadas en conjunto con un sensor *LiDAR*.

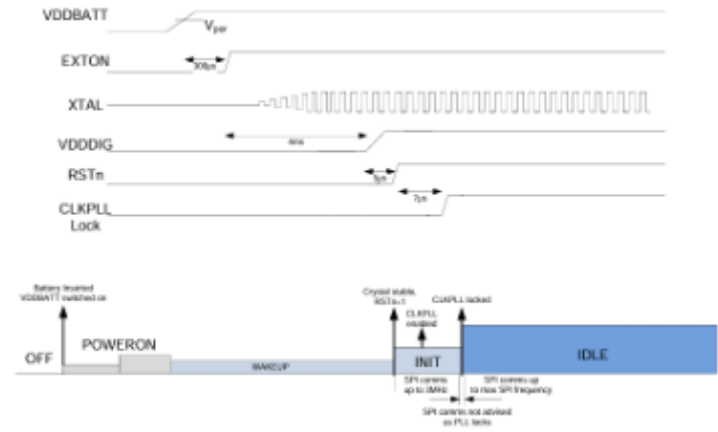


Figura 6: Señales moduladas (según fabricante IA Decawave).

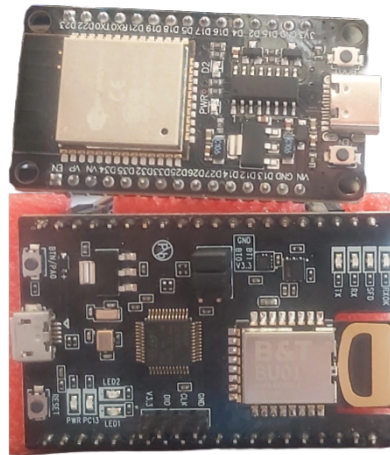


Figura 7: Combinación esp32 y UWB01.

El código generado para las anclas utiliza la librería *"DW1000Ranging.h"* esta librería nos obtiene los decibelios de la señal recibida por el robot móvil, así como el rango de distancia calculada a través de las ecuaciones definidas en la misma. Las funciones utilizadas son las siguientes:

```
1 DW1000Ranging.getDistantDevice()-\textgreater getRXPower()
2 DW1000Ranging.getDistantDevice()-\textgreater getRange().
```

El código utilizado para la etiqueta del robot móvil emplea dos librerías: *"DW1000Ranging.h"* e *"link.h"*. Se configura el acceso a la red de área local mediante una IP estática para evitar conflictos de acceso, utilizando la instrucción `const char *host = "192.168.1.103"`. Con la instrucción

```
1 fresh_link(uwb_data,
2 DW1000Ranging.getDistantDevice()->getShortAddress(),
3 DW1000Ranging.getDistantDevice()->getRange(),
4 DW1000Ranging.getDistantDevice()->getRXPower()
5 ),
```

se actualiza el valor del rango de señal de las anclas circundantes. Para visualizar los elementos, se desarrolló una interfaz gráfica de usuario (GUI) sencilla en Python. Esta GUI

muestra cuatro puntos verdes que identifican a las anclas, etiquetadas como A1782, A1783, A1784 y A1785. Además, se incluye un punto azul que representa al robot y está etiquetado como TAG. Este punto azul se utiliza para rastrear los movimientos y calcular la distancia entre las anclas, obteniendo así una posición calculada a partir de los códigos de las etiquetas y la información del TAG a través de WiFi. Esta interfaz se presenta en la figura 8.

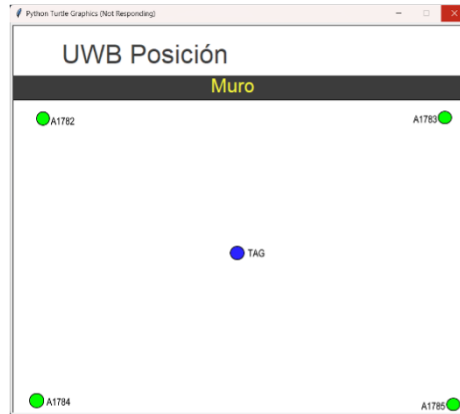


Figura 8: GUI para detección de posición.

Se programaron los elementos involucrados con dos códigos diseñados específicamente para las anclas y la etiqueta. Para las anclas, se empleó el siguiente algoritmo de configuración:

```

1 Serial.begin(115200);
2
3 delay(1000);
4
5 // Los acentos se han omitido intencionalmente
6 //inicia la configuracion
7
8 SPI.begin(SPI\_SCK, SPI\_MISO, SPI\_MOSI);
9
10 DW1000Ranging.initCommunication(PIN\_RST, PIN\_SS, PIN\_IRQ); //Reset,
11 CS, IRQ pin
12
13 //se define el sketch como ancla.
14
15 DW1000Ranging.attachNewRange(newRange);
16
17 DW1000Ranging.attachBlinkDevice(newBlink);
18
19 DW1000Ranging.attachInactiveDevice(inactiveDevice);
20
21 //Habilita el filtro para suavizar la distancia
22
23 //DW1000Ranging.useRangeFilter(true);
24
25 Mientras que para el TAG que identifica el robot se utiliza la siguiente
26 l gica de programacion:
27
28 //inicia la configuraci n
29
30 SPI.begin(SPI\_SCK, SPI\_MISO, SPI\_MOSI);
31
32 DW1000Ranging.initCommunication(PIN\_RST, DW\_CS, PIN\_IRQ);

```

```

33
34 DW1000Ranging.attachNewRange(newRange);
35
36 DW1000Ranging.attachNewDevice(newDevice);
37
38 DW1000Ranging.attachInactiveDevice(inactiveDevice);
39
40 // Iniciamos el modulo como una etiqueta
41
42 DW1000Ranging.startAsTag("7D:00:22:EA:82:60:3B:9C",
43 DW1000.MODE\_LONGDATA\_RANGE\_LOWPOWER);
44
45 uwb\_data = init\_link();
46
47 La definicion de la funcion para el calculo de la distancia es la
48 siguiente:
49
50 void tag\_pos (d1,d2,d3,d4,m1)
51
52 \{
53
54 cos\_a=(d1*d1+d2*d2+d3*d3+d4*d4-m1*m1) / (2*d1*d2*d3*d4)
55
56 x=(d1*cos\_a) + (d2*cos\_a)+ (d3*cos\_a)+ (d4*cos\_a);
57
58 y=(d1*cmath.sqrt(1-cos\_a*cos\_a))+ (d2*cmath.sqrt(1-cos\_a*cos\_a))+
59 (d3*cmath.sqrt(1-cos\_a*cos\_a))+ (d4*cmath.sqrt(1-cos\_a*cos\_a));
60
61 x1=round(x);
62
63 y=round(y);
64
65 \}

```

Tomando en cuenta el siguiente esquema de triangulación para el cálculo cercano a la distancia del tag y las anclas como se muestra en la figura 9.

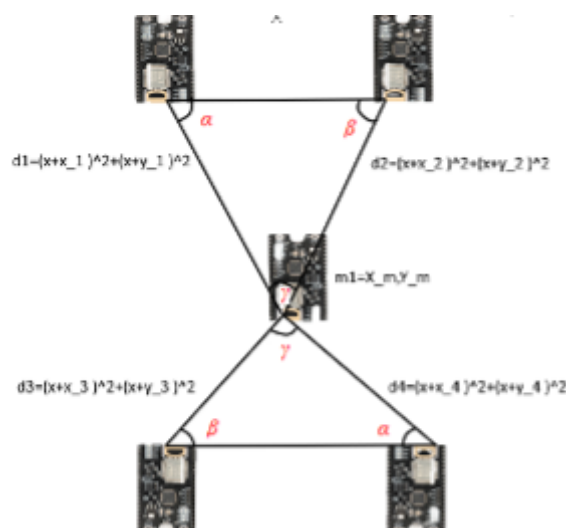


Figura 9: Triangulación bajo función tag-pos.



### 3. Resultados y discusión

Se ubicó el robot en seis puntos diferentes mediante un recorrido manual en la zona donde podía desplazarse. Esta área estaba cubierta por la señal de los puntos ancla. Se verificaron los puntos obtenidos y se midió la distancia, generando las siguiente tablas (1 y 2) de mediciones y datos :

pos(x,y)	Distancia en metros	Distancia real al tag
(0.9,2.8)	d1-2.9m, d2-3.5m, d3-3.0m, d4-2.8m	d1-2.7m, d2-3.4m, d3-2.8m, d4-2.6m
(2.2,2.9)	d1-3.6m, d2-3.0m, d3-1.6m, d4-2.6m	d1-3.4m, d2-2.7m, d3-1.9m, d4-2.0m
(2.4,2.7)	d1-3.2m, d2-2.8m, d3-2.0m, d4-2.4m	d1-3.0m, d2-3.0m, d3-2.1m, d4-2.2m
(1.6,3.1)	d1-3.5m, d2-3.4m, d3-1.0m, d4-3.0m	d1-3.7m, d2-3.0m, d3-0.9m, d4-3.0m
(1.6,2.7)	d1-3.4m, d2-3.3m, d3-0.9m, d4-3.4m	d1-3.2m, d2-3.3m, d3-1.2m, d4-3.3m
(2.0,2.3)	d1-3.0m, d2-2.5m, d3-3.1m, d4-2.7m	d1-2.8m, d2-2.2m, d3-3.0m, d4-2.2m

Tabla 1: Mediciones de seis puntos diferentes.

#	Distancia medida vs real	Errores
1	d1-2.9m vs d1-2.7m, d2-3.5m vs d2-3.4m, d3-3.0m vs d3-2.8m, d4-2.8m vs d4-2.6m	d1: 2.9 - 2.7 = 0.2, d2: 3.5 - 3.4 = 0.1, d3: 3.0 - 2.8 = 0.2, d4: 2.8 - 2.6 = 0.2
2	d1-3.6m vs d1-3.4m, d2-3.0m vs d2-2.7m, d3-1.6m vs d3-1.9m, d4-2.6m vs d4-2.0m	d1: 3.6 - 3.4 = 0.2, d2: 3.0 - 2.7 = 0.3, d3: 1.6 - 1.9 = -0.3, d4: 2.6 - 2.0 = 0.6
3	d1-3.2m vs d1-3.0m, d2-2.8m vs d2-3.0m, d3-2.0m vs d3-2.1m, d4-2.4m vs d4-2.2m	d1: 3.2 - 3.0 = 0.2, d2: 2.8 - 3.0 = -0.2, d3: 2.0 - 2.1 = -0.1, d4: 2.4 - 2.2 = 0.2
4	d1-3.5m vs d1-3.7m, d2-3.4m vs d2-3.0m, d3-1.0m vs d3-0.9m, d4-3.0m vs d4-3.0m	d1: 3.5 - 3.7 = -0.2, d2: 3.4 - 3.0 = 0.4, d3: 1.0 - 0.9 = 0.1, d4: 3.0 - 3.0 = 0.0
5	d1-3.4m vs d1-3.2m, d2-3.3m vs d2-3.3m, d3-0.9m vs d3-1.2m, d4-3.4m vs d4-3.3m	d1: 3.4 - 3.2 = 0.2, d2: 3.3 - 3.3 = 0.0, d3: 0.9 - 1.2 = -0.3, d4: 3.4 - 3.3 = 0.1
6	d1-3.0m vs d1-2.8m, d2-2.5m vs d2-2.2m, d3-3.1m vs d3-3.0m, d4-2.7m vs d4-2.2m	d1: 3.0 - 2.8 = 0.2, d2: 2.5 - 2.2 = 0.3, d3: 3.1 - 3.0 = 0.1, d4: 2.7 - 2.2 = 0.5

Tabla 2: Distancia medida vs real y sus errores

Posteriormente se obtuvo la desviación estándar:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (1)$$

Media o promedio de los errores:

$$\mu = \frac{3,4}{24} = 0,1417 \quad (2)$$

Suma de los cuadrados de las diferencias entre cada error y el promedio:

$$\sum_{i=1}^N (x_i - \mu)^2 = (0,2 - 0,1417)^2 + (0,1 - 0,1417)^2 + \dots + (0,5 - 0,1417)^2, \quad (3)$$

donde el promedio de los errores es aproximadamente  $\mu = 0,125$  metros y la desviación estándar  $\sigma = 0,224$  metros. Los cálculos del promedio y la desviación estándar de los errores de medición para cada conjunto de datos son los siguientes:

- d1:  $\mu_{d1} = 0,1333$ ,  $\sigma_{d1} = 0,1491$
- d2:  $\mu_{d1} = 0,1500$ ,  $\sigma_{d1} = 0,2062$

Para d1 y d2, el promedio es positivo y la desviación estándar es relativamente baja, lo que sugiere que los errores son consistentemente pequeños y positivos.

- d3:  $\mu_{d1} = -0,0500$ ,  $\sigma_{d1} = 0,1979$

Para d3, el promedio es negativo pero cercano a cero, y la desviación estándar es un poco más alta, lo que indica variabilidad en los errores de esta medición.

- d4:  $\mu_{d1} = 0,2667$ ,  $\sigma_{d1} = 0,2134$

Para d4, el promedio es el más alta entre todas las mediciones y la desviación estándar también es alta, sugiriendo una mayor variabilidad y errores más grandes en esta medición.

En la siguiente gráfica (figura 10 ) se presenta la dispersión de errores.

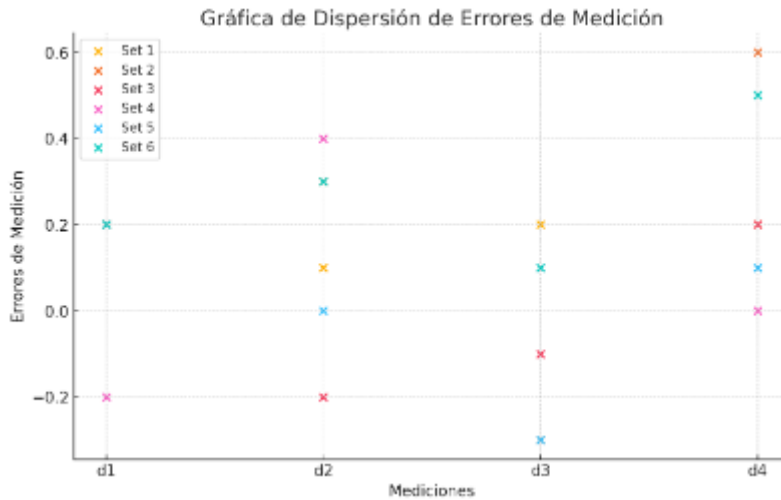


Figura 10: Gráfica de dispersión de errores.

La desviación estándar de las distancias medidas con respecto a las distancias reales al tag es aproximadamente  $=0.224$  metros, lo que indica una variabilidad de 22 centímetros. Este valor es aceptable, aunque puede mejorarse. Para reducir la variabilidad y acercarse más a la posición real del tag, se pueden realizar más mediciones y calcular promedios acumulados. Así, se podría alcanzar una precisión de 10 centímetros de diferencia a la posición real, de acuerdo con los parámetros del fabricante.

#### 4. Trabajo a futuro

Se plantea generar un algoritmo para mejora de precisión y exactitud de las lecturas de posición analizando el algoritmo para incremento de muestras en un periodo T de señal UWB mediante la siguiente ecuación:

$$v_{RMS} = \sqrt{\frac{1}{t} \int_0^T V^2(t) dt} \quad (4)$$

De igual forma se propone generar una GUI (*graphical user interface*) que ofrezca el plano donde el robot se moverá de forma autónoma, considerando la ganancia de precisión y un sensor *lidar*, mediante el uso de inteligenciar artificial para genera mapas y recorridos con forme a tareas preestablecidas.

#### 5. Conclusiones

Durante el desarrollo de los algoritmos de triangulación, se observó que el ancla d4 causaba la mayor variabilidad en las mediciones. Por ello, se procedió a revisar tanto el firmware de esta ancla como las pistas del circuito. Se identificó una pista que requería refuerzo para mejorar su conductividad. Tras aplicar estas mejoras, se logró un incremento satisfactorio del 15 % en la calidad de recepción de las mediciones.

Respecto a las mediciones entre las anclas, el sistema demostró ser adecuado para implementar la ubicación de robótica cooperativa en espacios *indoor* y procesos autónomos mediante IA(Inteligencia artificial). La discrepancia entre las mediciones y la ubicación real del tag es mínima, lo que permite aplicar algoritmos adicionales para mejorar la calidad de las anclas y, por ende, la precisión en la ubicación espacial del tag.

Considerando los aspectos mencionados anteriormente, la interfaz gráfica de usuario (GUI) puede migrar a un mapa arquitectónico vinculado a una base de datos de navegación, la cual se alimenta mediante la interacción de los robots con su entorno. Esto abre la posibilidad de mejorar y escalar los sistemas en entornos interiores.

#### Referencias

- [1] R. J. Fontana, Recent system applications of short-pulse ultra-wideband (UWB) technology," *IEEE Transactions on Microwave Theory and Techniques*, pp. 2087–2104, Sep. 2004, doi: 10.1109/TMTT.2004.834186.
- [2] E. M. Staderini, ÜWB Radars in Medicine."
- [3] IEEE Consumer Electronics Society and Institute of Electrical and Electronics Engineers, *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*.
- [4] S. Shyam, S. Juliet, and K. Ezra, ".^ UWB based Indoor Asset Tracking Architecture for Industry 4.0,in *Proceedings - 4th International Conference on Smart Systems and Inventive Technology, ICSSIT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 501–506, doi: 10.1109/ICSSIT53264.2022.9716281.
- [5] R. C. Qiu, ".^ study of the ultra-wideband wireless propagation channel and optimum UWB receiver design," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 9, pp. 1628–1637, Dec. 2002, doi: 10.1109/JSAC.2002.805249.
- [6] Z. Hao, B. Li, and X. Dang, ".^ Method for Improving UWB Indoor Positioning," *Math Probl Eng*, vol. 2018, 2018, doi: 10.1155/2018/8740872.
- [7] A. P. Chandrakasan *et al.*, "Low-power impulse UWB architectures and circuits," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 332–352, 2009, doi: 10.1109/JPROC.2008.2008787.
- [8] IEEE, *2011 IEEE 73rd Vehicular Technology Conference: Budapest, Hungary, 15-18 May 2011*.
- [9] IEEE International Symposium on Antennas and Propagation 2016 Fajardo, Institute of Electrical and Electronics Engineers, IEEE International Symposium on Antennas and Propagation 2016.06.26-07.01 Fajardo, IEEE Antennas and Propagation Society International Symposium 2016.06.26-07.01 Fajardo, APSURSI 2016.06.26-07.01 Fajardo, and AP-S 2016.06.26-07.01 Fajardo, *2016 IEEE Antennas and Propagation Society International Symposium proceedings: June 26-July 1, 2016, El Conquistador Resort, Fajardo, Puerto Rico*.

- [10] C. Vol, X.; A. Cardona, L. Garelli, and J. M. Gimenez, ".Advanced algorithms for indoor positioning through the combination of different types of sensors," 2019.
- [11] T. Fin De Máster, "Desarrollo de un sistema de localización en interior para robots móviles basado en UWB."
- [12] Q. Tian, K. I. K. Wang, and Z. Salcic, "Human body shadowing effect on uwb-based ranging system for pedestrian tracking," *IEEE Trans Instrum Meas*, vol. 68, no. 10, pp. 4028–4037, Oct. 2019, doi: 10.1109/TIM.2018.2884605.

