

SISTEMA DE LOCALIZACIÓN POR RADAR PARA UN ROBOT DE CONFIGURACIÓN ACKERMANN

RADAR LOCATION SYSTEM FOR AN ACKERMANN CONFIGURATION ROBOT

Juan Pablo Aguilera García

Tecnológico Nacional de México / IT de Celaya, México
16030343@itcelaya.edu.mx

Marco Antonio Magdaleno Cabrera

Tecnológico Nacional de México / IT de Celaya, México
16030324@itcelaya.edu.mx

Edgar Alejandro Rivas Araiza

Universidad Autónoma de Querétaro, México
erivas@uaq.mx

Salvador Manuel Malagón Soldara

Tecnológico Nacional de México / IT de Celaya, México
salvador.malagon@itcelaya.edu.mx

Recepción: 28/septiembre/2021

Aceptación: 30/noviembre/2021

Resumen

El proyecto consiste en desarrollar un sistema de localización capaz de llevar a cabo el control de la trayectoria de un robot móvil, este robot está diseñado bajo los parámetros de la configuración de movimiento Ackermann y fue construido con un chasis de acrílico de 3 mm. En él que se montaron los componentes que lo conforman, los cuales son: el sistema de dirección, sistema de tracción y el sistema de control. Además, se recurrió a la tecnología de impresión 3D para fabricar una base para evitar interferencias en el escáner láser.

Para la elaboración del sistema de localización se requiere el uso del escáner láser RPLiDAR modelo A1M8, que actúa como el principal componente para obtener una retroalimentación del ambiente. Esto se traduce como el uso de los datos que entrega el escáner, los cuales, mediante análisis matemáticos, ayudan al establecimiento de un sistema coordinado y el cálculo de la posición. La conexión del sistema de localización y el sistema de control brinda la posibilidad de programar

las trayectorias a seguir por el robot, sin embargo, debido a la existencia de errores, se implementa un sistema de recálculo de trayectoria que toma como principales herramientas de funcionamiento: la cinemática del robot y el filtro de partículas. Este recálculo reduce considerablemente el error en la trayectoria, lo cual queda demostrado en las pruebas realizadas en una trayectoria compuesta por: 3 movimientos rectos de 40 cm de longitud y 2 movimientos curvos con un radio de curvatura de 40 cm. La precisión en cuanto al cumplimiento de la trayectoria del sistema de recálculo es alta, ya que se observan variaciones por debajo de los 10 cm, comparado con un comportamiento ideal.

Palabras Claves: cinemática, configuración Ackermann, filtro de partículas, incertidumbre, LiDAR.

Abstract

The project consists of developing a location system capable of controlling the trajectory of a mobile robot, this robot is designed under the parameters of the Ackermann movement configuration and was built with a 3 mm acrylic chassis in the that the components that comprise it were assembled, which are: the steering system, traction system and the control system; In addition, 3D printing technology was used to manufacture a necessary base to avoid interference in the laser scanner.

For the development of the location system, the use of the RPLiDAR model A1M8 laser scanner is required, which acts as the main component to obtain feedback from the environment in which the robot moves, this is translated as the use of the data provided by the scanner, which through mathematical analysis help the establishment of a coordinate system and the calculation of the position. The connection of the location system and the control system offers the possibility of programming the trajectories to be followed by the robot, however, due to the existence of errors, a trajectory recalculation system is implemented that takes the kinematics as its main operating tools. of the robot and the particle filter created for the uncertainty calculation; This recalculation considerably reduces the error in the trajectory, which is demonstrated in the tests carried out on a trajectory composed

of 3 straight movements of 40 cm in length and 2 curved movements with a radius of curvature of 40 cm. The precision in terms of compliance with the trajectory of the recalculation system is high, since variations are observed below 10 cm compared to an ideal behavior.

Keywords: *Ackermann configuration, kinematics, LiDAR, particle filter, uncertainty.*

1. Introducción

En la actualidad los robots móviles desarrollan diversas tareas complejas como operaciones militares, rescate, agricultura, tareas civiles, exploración, y minería. Por ello, resulta imprescindible dotarlos de la habilidad de auto localización (posición y orientación), para que sean capaces de navegar en ambientes desconocidos. Sin embargo, existen una gran cantidad de inconvenientes en cuanto a localización se refiere, algunos pueden ser derivados por el mismo sistema del robot, también denominados errores sistemáticos, entre los que se encuentran los diámetros desiguales en llantas, la desalineación de éstas y errores debido al modelo cinemático. Por otra parte, se encuentran las condiciones poco favorables en el ambiente o errores estocásticos, algunas ejemplos de ellos son suelos dispares y los deslizamientos entre el suelo y la llanta [Malagón, 2015]. Este conjunto de errores genera un fenómeno conocido como incertidumbre, el cual puede entenderse como un parámetro que establece un intervalo alrededor del resultado de medición, este intervalo enmarca a los valores que también pudieran haberse obtenido con buena probabilidad durante dicha medición [Malagón, 2015].

Visto desde este punto, el trabajo citado busca la reducción en este incidente con la finalidad de encontrar un sistema más eficiente.

Existen algoritmos de localización tales como el de Markov [Thrun, 2000] y modelos como las redes neuronales [Valencia, 2014] que se utilizan para otorgar al robot la capacidad de ubicarse en el espacio de trabajo. Aunque hay métodos para ayudar a los robots a orientarse, se encuentra un grado de incertidumbre en la trayectoria. De acuerdo con la teoría, los sistemas de radar brindan una gran precisión en cuanto a posicionamiento de los sistemas de localización, por lo que, al usar este tipo de dispositivos, es posible crear un sistema que reduzca los índices de incertidumbre,

debido a que se puede tener una retroalimentación del medio en el que el robot móvil se desplaza [P. Bahl, 2000]. Una de las opciones más socorridas a la hora de intentar obtener información del ambiente, son las cámaras [Menegatti, 2006]. Sin embargo, este tipo de sensores provocan problemas como la necesidad de una gran capacidad de cómputo. Por otro lado, se encuentran los sensores láser [Gabriela Gallegos, 2010], estos sensores son muy precisos, pero tienen problemas en el correcto direccionamiento de su haz, debido a que el área de trabajo es demasiado puntualizada. Existen métodos en los cuales se busca calcular la incertidumbre con el fin de reducirla en los robots móviles mediante la implementación de un filtro de partículas [Malagón, 2015]. En [Mendoza-Hernandez Maria del Rayo, 2017] se explica que un sonar puede usarse como medio de localización acústica, la cual funciona por medio de impulsos sonoros para determinar la presencia de objetos, para el desarrollo de dicho trabajo se construyó un sonar, se usaron sensores ultrasónicos montados en una base, y dicha base cuenta con un servomotor que permitirá el giro de acuerdo con la programación de su posición.

En el presente documento se desarrolla el estudio de un sistema de localización cartesiana con la integración de un sistema LiDAR (Laser Imaging Detection And Ranging), para ser más específicos, es el uso de un escáner láser modelo RPLiDAR A1M8, el cual, detectará los límites dentro de los que puede desplazarse un robot móvil en un entorno previamente establecido. A su vez, el robot se mostrará de manera virtual con un mapeo 2D (plano cartesiano) con un algoritmo en Python implementado en un sistema embebido, en este caso una raspberry Pi 3 modelo B. Se mostrarán los resultados obtenidos mediante la comparación con un sistema informático, reportando la diferencia de los niveles de error. En resumen, el propósito del trabajo consiste en reducir el fenómeno de incertidumbre existente en la trayectoria de un de un robot móvil.

2. Métodos

La metodología se puede entender de manera más sencilla si se tiene una visión general de cada uno de los puntos que comprenden el proyecto, en la figura 1 se puede observar la composición y jerarquía de estos.

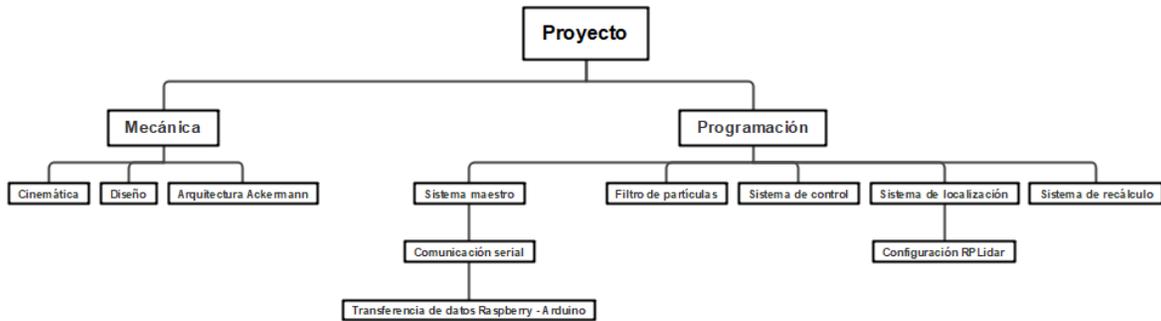


Figura 1 Estructura general del proyecto con los puntos principales para desarrollar la metodología.

Cinemática

El robot se diseñó bajo la configuración Ackermann, la cual consiste en un vehículo de 4 ruedas, de las cuales 2 están sujetadas a un sistema de dirección que varía el ángulo al que están dirigidas las ruedas, lo cual permite que la orientación del robot pueda ser modificada dependiendo de la necesidad.

Para el proyecto que aquí se desarrolla, se usó la cinemática de la configuración Ackerman equivalente a la configuración triciclo, esta puede observarse en figura 2.

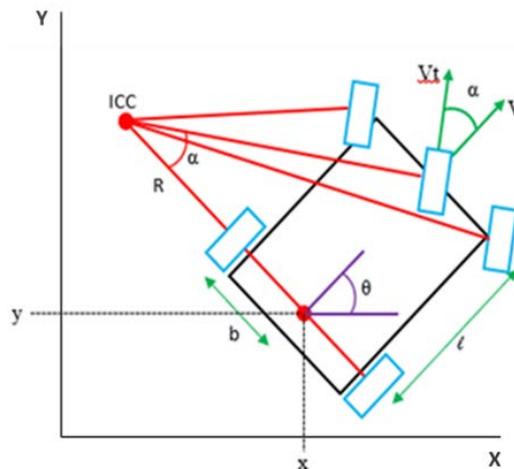


Figura 2 Representación de la configuración Ackermann y parámetros de la cinemática.

Esta configuración puede ser descrita por ecuaciones de cinemática que calculan el comportamiento matemáticamente. Dichas ecuaciones pueden observarse en la ecuación 1.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{\tan \alpha}{L} \end{bmatrix} vt \cdot \cos \alpha \quad (1)$$

Donde:

- vt : Velocidad tangencial del robot.
- L : longitud entre las llantas de tracción y dirección.
- \dot{x} : Velocidad en eje X.
- \dot{y} : Velocidad en eje Y.
- $\dot{\theta}$: Velocidad en eje Z o velocidad angular.
- R: Radio de curvatura.
- A: Ángulo de la dirección mecánica.
- V: Velocidad lineal del robot.

Diseño

Para realizar la construcción del robot, uno de los principales elementos a diseñar fue el sistema de dirección típico en la configuración Ackermann. Haciendo uso del software SolidWorks, se creó por completo dicha dirección mecánica con la finalidad de ser accionada por un servomotor que se encarga de variar el ángulo al que se dirigen las ruedas. En la figura 3 se muestra el ensamble de todas las piezas que componen este mecanismo.



Figura 3 Dirección mecánica tipo Ackermann ubicada en la parte delantera del robot.

Construido el sistema de dirección, puede montarse en el chasis del robot, que en este caso es una placa de acrílico de 3 mm. En esta placa también estará acoplado

el sistema de tracción compuesto por dos motores con su respectivo controlador, la batería de alimentación y el circuito de control. El chasis donde se montan todos los componentes se muestra en la figura 4.

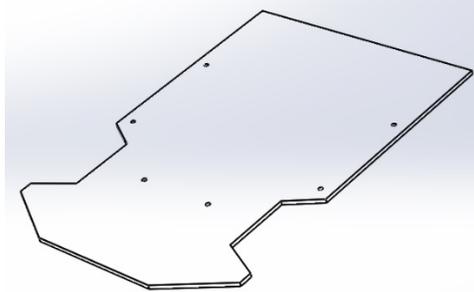


Figura 4 Chasis del robot ubicado en la parte inferior.

Otro de los componentes principales es el sistema de tracción del robot, el cual está constituido por dos motores de 12 v con encoder integrado y un controlador para estos. Dicho sistema al igual que la dirección mecánica y el sistema de control, está montado en el chasis principal, en la figura 5 puede observarse este sistema de tracción.

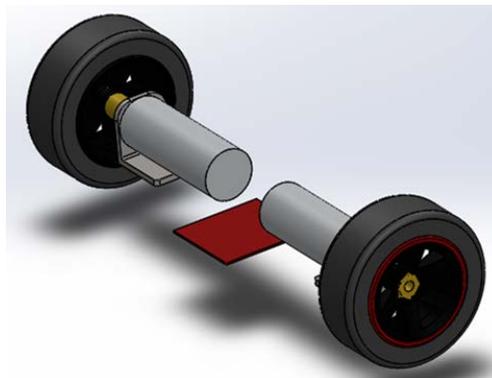


Figura 5 Sistema de tracción.

Ya que la forma en que trabaja el sensor láser requiere que no existan interferencias en la señal enviada, se buscó solucionar este problema. Se realizó el diseño de una base, mostrada en la figura 6, la cual cumple con la función de soportar la raspberry del sistema de control y una batería de alimentación para la misma.

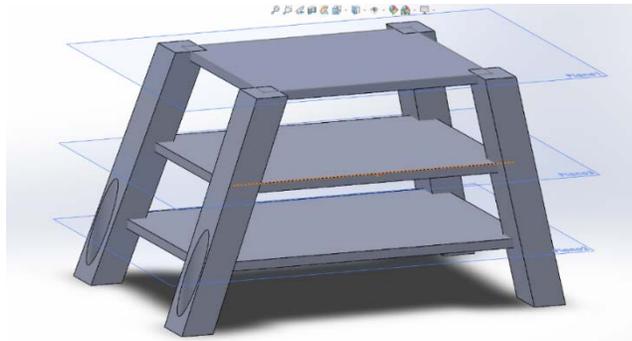


Figura 6 Base superior fabricada en impresión 3D.

Para manufacturar esta pieza se requiere el uso de impresión 3D, entre otras modificaciones necesarias para el desarrollo del proyecto, se hizo un cambio en la tornillería para dar un mejor ajuste a la nueva pieza, y así mismo, dar más estabilidad a los componentes. En la figura 7 se presenta la estructura del completamente armada.



Figura 7 Estructura completa del robot.

Navegación

Se estableció un espacio de trabajo con el fin de realizar las pruebas, dicho espacio cuenta con un área de 4 metros cuadrados, éste puede observarse en la figura 8.

Las pruebas realizadas en esta área tienen la finalidad de observar y medir los errores en los movimientos a través de las trayectorias programadas. Es posible determinar valores de incertidumbre tanto física como virtualmente, registrando las medidas que se obtienen de los movimientos del robot y los valores de coordenadas

que entrega el sistema de navegación. Tomando en cuenta estos datos, se pueden implementar las herramientas necesarias para mejorar el sistema de navegación del robot.



Figura 8 Área de trabajo construida con la finalidad de realizar pruebas.

Programación del sistema

El desarrollo del robot tiene una extensión considerable en cuanto a la programación, ya que se logra la unión de diferentes sistemas control. Todo el funcionamiento del sistema se explica a profundidad en esta metodología. Como apoyo visual, en la figura 9 se muestra un diagrama del funcionamiento de toda la programación implementada.

A continuación, se explica cada tarea realizada en los pasos del diagrama de una manera más detallada. En la sección correspondiente al escáner RPLiDAR se explica el primer punto del diagrama, el cual nos habla de la obtención de datos. Para los cuatro bloques siguientes, se abarca desde el cálculo de puntos hasta el cálculo de la posición actual.

En los siguientes párrafos, se desarrolló la sección de mapeo, que explica todo este procedimiento. Así mismo, se desarrolló la sección de filtro de partículas para la explicación de dicho bloque. Por último, se desarrolló la sección del sistema de recálculo con la finalidad de esclarecer la última sección del diagrama en la que se cierra el ciclo de la trayectoria.

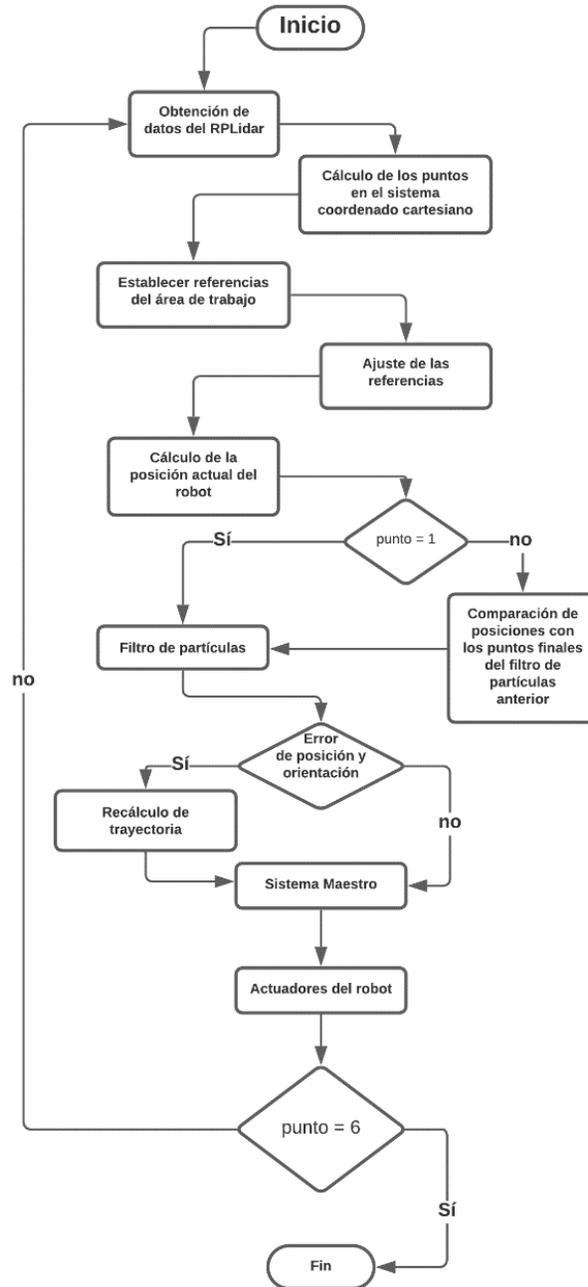


Figura 9 Diagrama de flujo del sistema de localización en el que se explica el funcionamiento general.

Escáner RPLiDAR

El RPLiDAR A1M8 es un láser bidimensional de 360° desarrollado por SLAMTEC. El sistema puede realizar un escaneo de 360° dentro de un rango de 12 m. El escaneo de la frecuencia del RPLiDAR A1M8 alcanza los 5.5 Hz cuando muestra 1450 puntos por ronda. El escáner se puede configurar hasta 10 Hz como

máximo. RPLiDAR A1M8 es básicamente un sistema de medición de triangulación láser. Funciona en excelentes condiciones para ambientes interiores y exteriores, sin exposición directa a la luz solar. El escáner se muestra en la figura 10 [Iqbal, 2020].

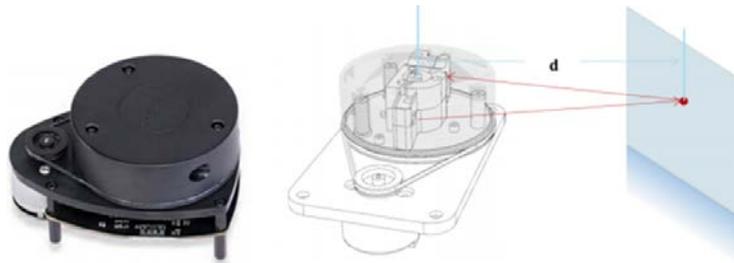


Figura 10 Escáner láser RPLiDAR A1M8 y su funcionamiento para calcular distancia.

Light Detection and Ranging (LiDAR) funciona de la misma manera que los telémetros ultrasónicos con luz láser que se utilizan en lugar de ondas sonoras. Existen muchas aplicaciones para los telémetros LiDAR, desde la realización de mediciones simples de distancias lineales en topografía hasta la creación de mapas increíblemente detallados del fondo marino en batimetría. BMW, Uber y Waymo están invirtiendo fuertemente en tecnología LiDAR para sus programas de vehículos autónomos. Las altas frecuencias de muestreo y las lecturas precisas son la clave para que los sensores LiDAR sean tan atractivos para la construcción de mapas [Iqbal, 2020].

El RPLiDAR A1 utiliza un puerto serie 3.3V-TTL (UART) como interfaz de comunicación. El procesador interno procesa la señal láser recibida y proporciona los datos correspondientes. Para obtener datos en vivo escaneados por el LiDAR, conectamos el LiDAR con raspberry pi 3b. La comunicación entre LiDAR y raspberry pi es comunicación UART. Los datos obtenidos en el raspberry pi sobre el LiDAR contendrán la distancia del obstáculo, y qué ángulo ha girado el LiDAR para escanear, tanto el punto como el factor de calidad. Luego, estos datos pueden usarse para emplear los algoritmos utilizados para la detección y evasión de obstáculos. El mapa de nube de puntos se puede ver en el software predeterminado llamado capturador de fotogramas [Schubert, 2018].

La mayoría de los sensores LiDAR de 360° que se utilizan en la actualidad utilizan un solo sensor que gira 360° mientras toma lecturas. Si bien es efectivo, existe una preocupación creciente sobre la vida útil de estos dispositivos debido a las partes móviles. Actualmente existe un enfoque en el desarrollo de tecnología que utiliza un diseño de estado sólido para omitir este problema. También hay esfuerzos de investigación para convertir los LiDAR 2D existentes en LiDAR 3D, todo esto para reducir el costo y aumentar la accesibilidad a la cartografía 3D [Schubert, 2018].

Mapeo

Para la representación gráfica en 2D del entorno en el que se desplaza el robot móvil, se usa el software Python 2, en el cual se integra la librería matplotlib, para representar los puntos por coordenadas de las esquinas del área de trabajo con dimensiones de 2 x 2 m. Primero hay que entender la comunicación del escáner RPLiDAR, el cual por defecto posee librerías para Python y tiene una comunicación en cierto ciclo de trabajo, en sí, este escáner envía una lista de datos, los cuales están seccionados por vectores y cada uno de estos vectores contiene tres datos en particular, el primero es la calidad de la señal del láser que se reflejó en algún lugar, este dato nos indica el porcentaje de error con el cual se puede confiar. El segundo dato que nos plantea el escáner es la distancia desde el centro del dispositivo hasta el lugar donde el láser se reflejó, este dato está en mm y por último, el dato que está colocado en la tercera posición del vector es el ángulo donde el láser se refleja con respecto al 0, como se muestra en figura 11.

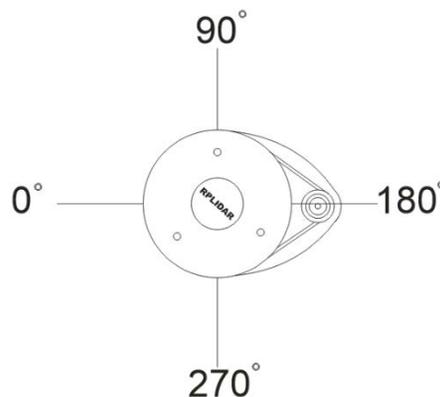


Figura 11 Cuadro de referencia del RPLiDAR para tener una referencia de ángulo.

Estos datos pueden variar el número de vectores dependiendo de la calidad del láser, ya que, si es demasiado grande el error, esos datos no se toman en cuenta. De esta forma, siempre se trata de encontrar la mayor cantidad de datos posibles en la lista, siempre y cuando la señal sea óptima. Ya con estos datos podemos usar cada vector para, mediante las ecuaciones 2 y 3, convertir los datos a coordenadas Cartesianas y representarlos tanto en el eje Y como en el eje X.

$$X = Distancia * Coseno(\acute{a}ngulo) \quad (2)$$

$$Y = Distancia * Seno(\acute{a}ngulo) \quad (3)$$

Estos puntos estarán esparcidos en el plano XY, por lo cual, después cada vector es transformado a coordenadas cartesianas. Por conveniencia se enfocó en sólo usar las esquinas del área de trabajo, por lo cual, el siguiente paso del algoritmo en Python, fue encontrar las esquinas. La mejor forma en la que se pueden encontrar es usando las distancias que hay de cada punto con los demás, esto debido a que, si se toma en cuenta que el ambiente es cuadrado, la distancia de las diagonales es mucho mayor a cualquier otra distancia con cualquier otro punto de la figura. De esta manera se puede saber cuáles son las esquinas, y después, representarlas en el plano Cartesiano. Para ello se usa la ecuación 4, la cual da como resultado la distancia entre dos puntos, después sólo se debe obtener la distancia de todos los puntos con los restantes, para al final comparar, encontrar las diagonales y saber a qué puntos corresponden las distancias.

$$r(PA, PB) = \sqrt{(Bx - Ax)^2 + (By - Ay)^2} \quad (4)$$

Se planea que después de encontrar las esquinas, para representar la posición del robot móvil en el área de trabajo, usar sólo una esquina como referencia, la cual por medios matemáticos se trasladará al punto origen del plano. Lo anterior con el fin de que los demás puntos de las esquinas también se trasladen y se roten de forma conveniente. De esta forma se logra tener siempre una posición del robot con valores positivos.

En la figura 12 se muestra un ejemplo de cómo es que el robot debe mostrarse en la gráfica 2D de matplotlib con representación de puntos.

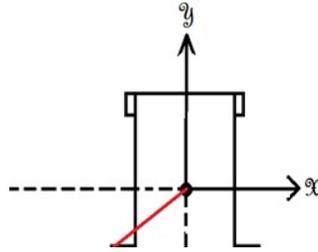


Figura 12 Representación gráfica del robot mostrado como un punto en el simulador.

La rotación y traslación mencionadas anteriormente se usan para establecer el origen en la coordenada (0,0), ya que, el escáner por defecto ubica su propia posición como el origen del sistema. Todo este procedimiento se realiza mediante un análisis matemático, en la figura 13 se representan gráficamente los movimientos establecidos dentro de la ventana de la librería matplotlib.

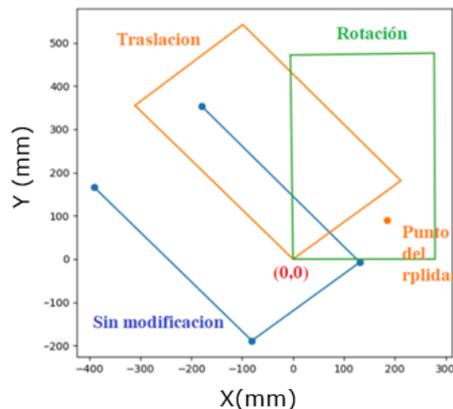


Figura 13 Ajuste del cuadro de referencia.

La ventana tiene un área de simulación de 500 x 400 mm, con la finalidad de lograr mostrar íntegramente el ambiente. Debido al sistema de referencia que el escáner posee, se puede observar en color azul el ambiente tal como es entregado por el RPLiDAR. En color naranja se muestra el ambiente una vez trasladado al origen de la ventana de matplotlib, o bien, la coordenada (0,0). Por último, en color verde se muestra el ambiente después de realizar la rotación, gracias a lo cual, se puede tener un sistema paralelo a los ejes X y Y.

El análisis matemático que se encuentra detrás de estos movimientos puede describirse al usar las fórmulas para cada caso, la ecuación 5 se utiliza para llevar

- XN : Coordenada en x después de la rotación
- YN : Coordenada en y después de la rotación

Se pueden observar gráficamente los parámetros de rotación para establecer el ambiente paralelo a los ejes de referencia en la figura 15.

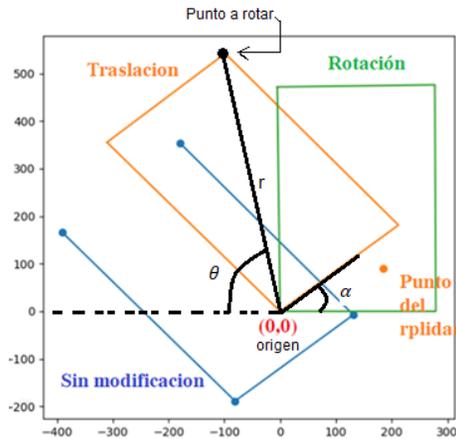


Figura 15 Ejemplo de rotación de un punto.

Filtro de partículas

El filtro de partículas se usa con la finalidad de “simular” el movimiento del robot, en este programa se encuentra establecida la cinemática a la cual responde dicho robot, y se pueden observar las variaciones que pueden existir en el movimiento a través de una trayectoria establecida. Este ejemplo puede simularse en cuestión de segundos un gran número de partículas, cada una de éstas puede suponerse como un robot con la misma configuración que el real, de modo que puede obtenerse un rango dentro del cual, existen altas probabilidades que termine situado el verdadero robot. Este rango como fue mencionado con anterioridad, es conocido como incertidumbre. En este proyecto, el filtro de partículas toma relevancia tanto para seguir una trayectoria, a manera de simulación, como para crear un sistema de recálculo capaz de mejorar la trayectoria seguida por el robot, ya que, con él, se puede predecir el comportamiento del robot en base a la posición actual. De este modo se pueden obtener los datos necesarios para reducir el error en el siguiente movimiento de la trayectoria. En la figura 16 se muestra un ejemplo del

funcionamiento del filtro creado. Se puede apreciar cómo el filtro simula el movimiento de 50 partículas que establecen un rango de posibles posiciones finales. Cada línea de color que se observa está compuesta por los distintos puntos por los que pasa la partícula, cada partícula posee un color diferente, por lo cual se muestra un espectro compuesto por 50 colores en total.

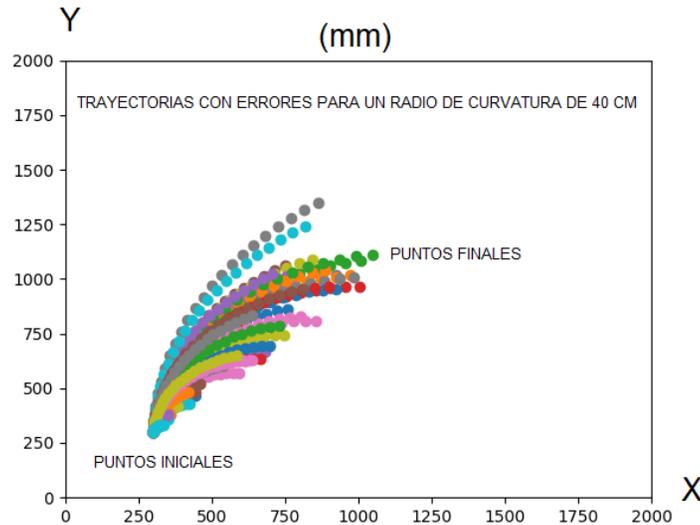


Figura 16 Ejemplo de funcionamiento del filtro de partículas para calcular la incertidumbre.

Sistema de recálculo

El sistema de recálculo de trayectoria utiliza como herramientas el filtro de partículas, la cinemática y el análisis matemático, con la finalidad de reducir en error en el seguimiento de la trayectoria programada. Esto se logra recalculando el movimiento siguiente con base en los datos obtenidos (posibles posiciones finales) en la simulación realizada por el filtro de partículas. A su vez, estos datos se aplican a las ecuaciones 8 a 11.

$$\Delta x (x_{ob} - x) \cos \theta + (y_{ob} - y) \sin \theta \quad (8)$$

$$LA = \sqrt{(x_{ob} - x)^2 + (y_{ob} - y)^2} \quad (9)$$

$$r = \frac{l}{R} = \frac{2\Delta x}{LA^2} \quad (10)$$

$$\tan \alpha = \frac{l}{R'} \quad (11)$$

Donde:

- Δx : Diferencia en eje x
- LA : Distancia del robot al punto objetivo
- r : Curvatura descrita por el robot
- R : Radio de curvatura
- l : Distancia entre las ruedas de tracción y las de dirección
- α : Ángulo de la dirección mecánica

En la figura 17 se muestran de manera gráfica los parámetros usados en estas fórmulas, lo cual ayuda a tener una mejor comprensión de este funcionamiento.

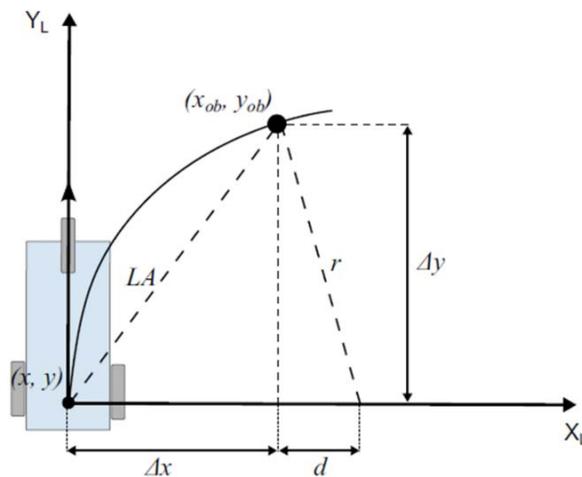


Figura 17 Vista gráfica de los parámetros.

Los parámetros de las fórmulas cubren las necesidades del sistema. Ejemplo de esto, es que, gracias a estas fórmulas y los datos obtenidos en el filtro de partículas, es posible predecir el siguiente movimiento del robot dentro de la trayectoria programada.

Trayectoria

Con la finalidad de realizar pruebas para conocer el funcionamiento de los sistemas, se diseñó una trayectoria que se ajusta fácilmente al tamaño del ambiente. La trayectoria está compuesta por movimientos rectos y curvos con la

finalidad de que sean más completas las muestras. Se establecieron distancias de 40cm de longitud para las líneas rectas, y como radio de curvatura en los movimientos curvos. Esta trayectoria se puede observar de manera física en la figura 18.



Figura 18 Trayectoria mostrada físicamente.

3. Resultados

Los resultados se basan en el comportamiento del robot en el ambiente y trayectoria mostrados en la metodología, estos resultados se dividen en dos grupos principales que son: el sistema con recálculo y el sistema sin recálculo. Lo anterior con la finalidad de observar que la corrección de errores sea funcional.

Se realizaron mediciones físicas con uso de un flexómetro en el movimiento real del robot, al igual que se registraron los datos de las mediciones realizadas por el sistema de localización, al ser un sistema coordinado en las dimensiones (x, y), se pueden realizar gráficas de las mediciones en ambas coordenadas y obtener un apoyo visual bastante bueno.

Sistema sin recálculo

En el sistema sin recálculo el robot realiza la trayectoria sin hacer comparaciones con los puntos del filtro de partículas, de modo que no hay corrección en caso de error. En la tabla 1 se enlistan las mediciones obtenidas en cada coordenada.

Tabla 1 Mediciones en cada punto de la trayectoria (sin recálculo).

Coordenada	Trayectoria Real recorrida		Trayectoria medida por RPLiDAR		Trayectoria ideal	
	X (mm)	Y(mm)	X(mm)	Y(mm)	X(mm)	Y(mm)
punto 1	179	202	156	181	179	202
punto 2	174	612	153.0439	597.4597	179	602
punto 3	599	977	474.42	1046.2045	579	1002
punto 4	931	1212	770.94	1290.508	979	1002
punto 5	1479	1132	1183.33	884.72	1379	602
punto 6	1884	1027	1733.42	998.207	1379	202

En la figura 19 se presenta la trayectoria ideal programada, la cual, como se mencionó en la metodología, es el movimiento exacto que se debería realizar si no existieran errores. Gracias a este gráfico es posible comparar los resultados obtenidos y verificar cuál es la cantidad de error existente.

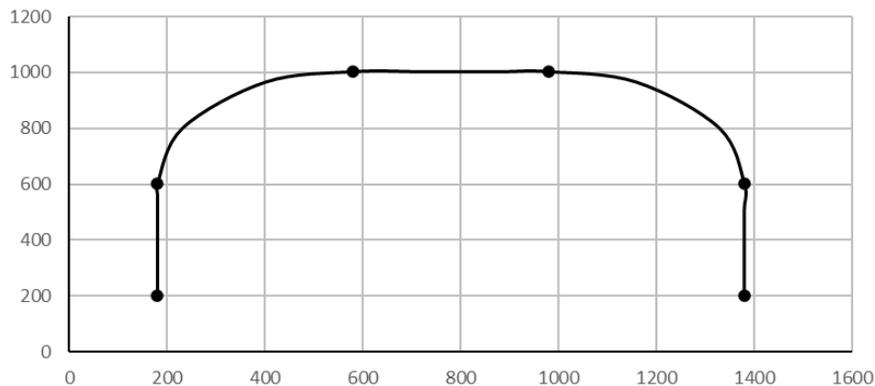


Figura 19 Trayectoria ideal programada.

Usando los puntos enlistados en la tabla 1, se pueden realizar gráficas para un apoyo visual, en la figura 20 se muestra la trayectoria recorrida por el robot de manera física. En la figura 21 se muestra la trayectoria medida por el RPLiDAR y el sistema de localización, este gráfico se puede entender como el recorrido virtual del robot.

Sistema con recálculo

En el sistema con recálculo se realiza una corrección del error en busca de reducir el mismo, en la tabla 2 se enlistan las coordenadas medidas durante la prueba al sistema con recálculo implementado.

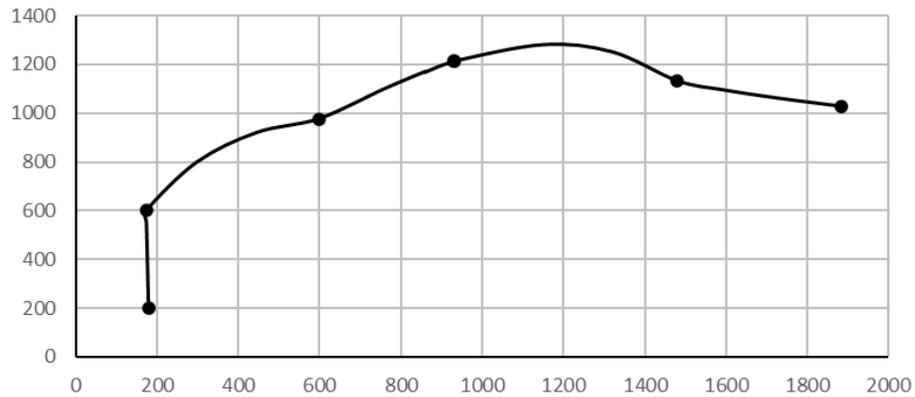


Figura 20 Trayectoria real recorrida sin recálculo medida físicamente.

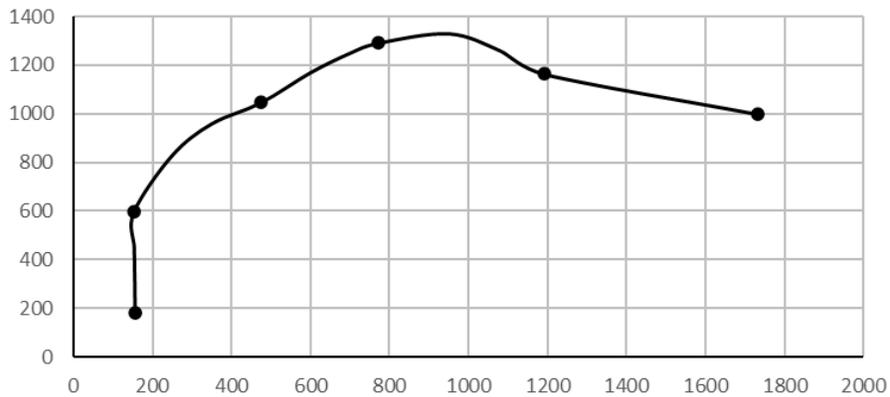


Figura 21 Trayectoria medida por RPLiDAR sin recálculo.

Tabla 2 Mediciones en cada punto de la trayectoria (con recálculo).

Coordenada	Trayectoria Real recorrida		Trayectoria medida por RPLiDAR		Trayectoria ideal	
	X (mm)	Y(mm)	X(mm)	Y(mm)	X(mm)	Y(mm)
punto 1	164.597	174.32	158.692	172.307	179	202
punto 2	176.059	597.34	184.8283	593.3807	179	602
punto 3	594.59	959.32	327.7754	814.7922	579	1002
punto 4	1084.597	974.32	906.185	1148.5632	979	1002
punto 5	1414.597	566.32	1216.98	846.1733	1379	602
punto 6	1369.597	99.32	1302.1369	207.7896	1379	202

Con el fin de realizar la comparación entre la trayectoria realizada y la trayectoria ideal, se muestra en la figura 22 la trayectoria ideal que establece el movimiento que el robot debería realizar, si no existiera ningún tipo error.

En la figura 23 se presenta el gráfico obtenido una vez implementado el sistema de recálculo, con las coordenadas medidas físicamente en la trayectoria real del robot.

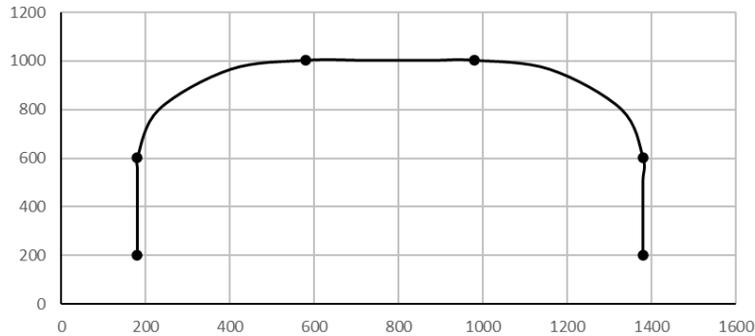


Figura 22 Trayectoria ideal programada.

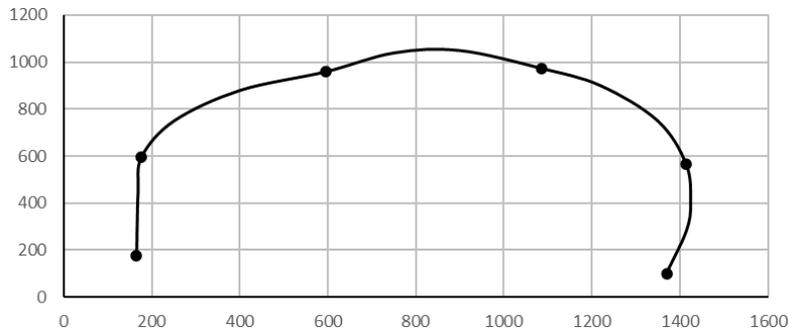


Figura 23 Trayectoria real recorrida con recálculo.

La figura 24 representa la trayectoria medida por el sistema de localización una vez implementado el sistema de recálculo.

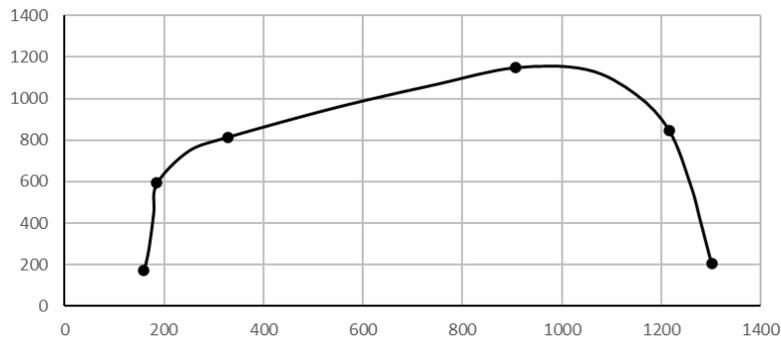


Figura 24 Trayectoria medida por RPLiDAR con recálculo.

Con la finalidad de tener una referencia más clara de los resultados, en las tablas 3 y 4, se enlistan las cantidades de error en las coordenadas de cada punto medido, comparado con la trayectoria ideal. La tabla 3 contiene los valores del sistema sin recálculo y la tabla 4 los valores del sistema con recálculo.

Tabla 3 Error en las coordenadas de cada punto de la trayectoria sin recálculo.

Coordenada X (mm)	Coordenada Y (mm)
0	0
5	10
20	25
48	210
100	530
505	825

Tabla 4 Error en las coordenadas de cada punto de la trayectoria con recálculo

Coordenada X (mm)	Coordenada Y (mm)
0	0
11.462	23.02
29.993	15
120	0
50	8
5	75

Estas coordenadas pueden graficarse para obtener una visión más clara del error. En la figura 25 puede observarse la gráfica obtenida con los datos de las tablas 3 y 4. Gracias a esta gráfica se entiende la magnitud del error en cada sistema. En color negro, se muestra el comportamiento en la coordenada Y tanto para el sistema sin recálculo (derecha) como para el sistema con recálculo (izquierda). En color azul se presenta el comportamiento en la coordenada X en ambos sistemas, sin recálculo (izquierda) y con recálculo (derecha).

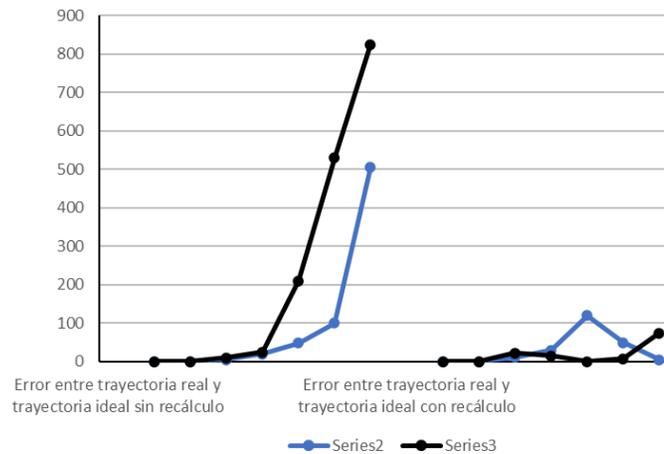


Figura 25 Comparación del error de cada sistema para verificación de efectividad

4. Conclusiones

El sistema de localización muestra resultados muy positivos, se logra una precisión bastante buena en cuanto al cálculo de la posición de robot, gracias a esto se puede ajustar de manera más sencilla el área de desplazamiento del robot. Los resultados más importantes que se obtienen del proyecto son los pertenecientes al sistema de recálculo de la trayectoria, en estos se puede observar una gran diferencia en cuanto al seguimiento de la trayectoria. Fue reducido considerablemente el error y esto da pie a creer que este trabajo tiene mucho futuro, la capacidad del robot de conocer su posición actual da oportunidad a reprogramar su avance gracias a la introducción del filtro de partículas. Este filtro nos entrega datos útiles para realizar el análisis matemático necesario para la acción de recálculo. Al dimensionar la cantidad de error que se puede reducir, no es difícil creer que este sistema puede ser la diferencia entre el cumplimiento de una tarea o la falla de ésta. Si bien aún hay mucho que desarrollar en esta área, se cree que es un excelente inicio para realizar trabajos más avanzados.

5. Bibliografía y Referencias

- [1] Bahl, V. P. RADAR: an in-building RF-based user location and tracking system. Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. pp 26-30. 2000.
- [2] Barrientos, A. Fundamentos de robótica. Mcgraw Hill. 2da edición. 2007.
- [3] Demirli, K. Y. Sonar based mobile robot localization by using fuzzy triangulation. Robotics and Autonomous Systems. pp. 1-15. 2000.
- [4] Gabriela Gallegos, P. R. Indoor SLAM based on composite sensor mixing laser scans and omnidirectional images. 2010 IEEE International Conference on Robotics and Automation. Anchorage, AK, USA: IEEE. pp. 3519-3524. 2010.
- [5] Malagon-Soldara, S. M., Rivas-Araiza, E. A. Cálculo de incertidumbre en un filtro de partículas para mejorar. Cuernavaca Morelos, México: XI Congreso Internacional sobre Innovación y Desarrollo Tecnológico. 2015.

- [6] Malagón, S. Cálculo de incertidumbre en un filtro de partículas para mejorar la localización en robots móviles. Cuernavaca Morelos, México. 2015.
- [7] Mendoza-Hernandez, M. R. Sonar programado en una Raspberry Pi para emplear en navegación de robots móviles. Asociación Mexicana de Mecatrónica A.C, pp. 1-2. 2017.
- [8] Menegatti, E. Omnidirectional vision scan matching for robot localization in dynamic environments. *IEEE Transactions on Robotics*. pp. 523-535. 2006.
- [9] Thrun, S. Probabilistic Algorithms in Robotics. *AI Magazine*, 21(4), pp. 93. 2000.
- [10] Valencia, D. Diseño e implementación de un prototipo de robot asistente para personas con discapacidad motriz y adultos mayores, basado en inteligencia artificial. Universidad Politécnica Salesiana. 2014.