

Implementación de un robot móvil para optimizar el flujo documental en el área de logística usando tecnologías de robótica autónoma basada en la plataforma de software libre ROS

Implementation of a mobile robot to optimize the documentation flow in the logistics area using autonomous robotics technologies based on the free software platform ROS

Iván Calle¹, Rider Paredes¹, Cristopher Bazan¹, Aldo Guardia¹

¹Facultad de Ingeniería Mecánica, Universidad Nacional de Ingeniería, Lima-Perú.

Recibido : 16/08/2017 Aceptado: 02/10/2017

RESUMEN

El presente proyecto consistió en la implementación de un robot móvil autónomo capaz de facilitar el flujo de documentos entre las diferentes áreas de una empresa, universidad, etc. Este robot es capaz de navegar de manera completamente autónoma en ambientes reales tal como los ambientes del CTIC, FIM, FIEE, etc. Tan solo especificando el punto inicial, el mapa del ambiente de navegación, y el punto deseado, este robot es capaz de generar el camino óptimo para llegar a dicha meta, y luego seguir este camino con la capacidad de evitar obstáculos si estos se presentan. Dadas estas características, este robot se puede usar en aplicaciones logísticas en donde el robot debe llevar paquetes, cargas, etc., a algún punto especificado por el usuario. En el proyecto se tienen dos modelos, el primer robot llamado R2D2-R1 puede llevar cargas de hasta 3kg, y el segundo robot llamado R2D2-R2 puede llevar cargas de hasta 25kg. Cabe señalar que los algoritmos implementados en este proyecto representan el estado del arte del campo de la robótica autónoma, y su desempeño se ha comprobado en las diversas pruebas de navegación realizadas en ambientes de la UNI. Este proyecto contribuye a cumplir con la misión de la UNI en los temas de innovación y gestión tecnológica para contribuir al bienestar de la sociedad y desarrollo del país.

Palabras clave: Robot móvil, navegación autónoma, planificación de trayectorias, evitamiento de obstáculos, aplicaciones logísticas.

ABSTRACT

The present project is about the implementation of an autonomous mobile robot designed for logistic tasks in different areas of a company, university, etc. This robot is able to navigate autonomously in real environments, you just need to specify the initial position, the grip map of the world and the target locations, and the robot will generate automatically the optimal path to reach the target positions, and then will follow this path while avoiding obstacles such as persons, trash bins, etc. These characteristics allow that our robot can be used in logistic tasks where the robot needs to carry loads from one place to another. In this project we developed two robot models, the first one called R2d2-R1 can carry loads of up to 3kg, and the second one called can carry loads of up to 25Kg. The algorithms implemented in this project represent the state-of-the-art methods and its performance has been proved in the several experiments carried out with these two robots.

Keywords: Mobile robot, autonomous navigation, path planning, obstacle avoidance, logistic applications.

1. INTRODUCCIÓN

El desarrollo de robots móviles autónomos ha mostrado un gran avance en los últimos años y las técnicas desarrolladas se vienen usando no solo en robots

móviles sino en sistemas tales como los vehículos que no necesitan conductores [1]. De hecho estos sistemas ya se vienen adoptando fuertemente a nivel industrial [2] y se espera que al final de esta década se ofrezcan

* Correspondencia:
E-mail: ivan.calle.flores@gmail.com

de manera comercial, lo que provocara una revolución en los sistemas de transporte.

A nivel nacional se han desarrollado algunos proyectos vinculados a estos temas siendo el más resaltante el proyecto “Prevención de riesgos mortales debido a gases tóxicos en ambientes mineros mediante la recolección y análisis de data usando tecnología de robótica autónoma” en el cual se implementó un robot móvil especialmente diseñado para ser usado en minas.

Para el desarrollo de robots que operen en entornos reales se requiere el uso de técnicas que tomen en cuenta las diversas fuentes de ruido en los sensores, actuadores y modelos. Las técnicas probabilísticas basadas en el Filtro de partículas y los filtros de Kalman representan el estado del arte en el desarrollo de sistemas de navegación autónoma [1], [3]. Estas técnicas se usan en las tareas de localización del robot, y la construcción de mapas de los entornos de navegación. En el caso de las tareas de planeamiento comúnmente se hace uso de las técnicas de búsqueda de la Inteligencia Artificial [1], [4]. Y para el caso de control de los motores se hace uso de las técnicas de la Ingeniería de Control [5].

A nivel de software la plataforma ROS se viene consolidando como el sistema operativo estándar en el desarrollo de sistemas robóticos ya que no solo provee una implementación de muchos algoritmos sino que tiene una serie de drivers para el manejo de una amplia gama de sensores y actuadores [6]. Además esta librería puede usarse en combinación con otros softwares como MATLAB, permitiendo esto que se pueda desarrollar algoritmos de navegación en este lenguaje de alto nivel, y que el manejo de los sensores y actuadores se haga usando programas en ROS.

A nivel de hardware existen diversas plataformas en el mercado, de hecho existen empresas como STARSHIP que ya vienen vendiendo robots capaces de hacer entregas, sin embargo a nivel nacional todavía no se han desarrollado robots capaces de replicar este tipo de autonomía. Los prototipos desarrollados en el proyecto no solo son de código abierto, fáciles de usar y de bajo costo, sino que se pueden usar (de hecho ya se han usado en los cursos de robótica dictados en la FIM) en el dictado de cursos de robótica, en donde los alumnos tienen acceso a toda la información necesaria para entender el funcionamiento de dichos sistemas.

2. FUNDAMENTOS TEÓRICOS

Cualquier robot con capacidades de navegación autónoma en entornos reales debe tener cuatro módulos: Localización, Planeamiento de trayectorias, Evitamiento de obstáculos, y Control de movimiento [1].

a. Localización. Este módulo es el encargado de estimar la configuración del robot respecto al mapa del entorno de navegación. El método más

usado está basado en el filtro de partículas el cual es capaz de trabajar con los sensores laser y los mapas de ocupación [3].

```

Algoritmo Localizacion_MC( $\{x_{t-1}^m\}, u_t, z_t$ )
L1: for  $m = 1:M$ 
L2:  $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$ 
L3:  $w_t^{[m]} = p(z_t | x_t^{[m]})$ 
L4: end
L5: for  $i = 1:M$ 
L6: Muestrear " $x^{[m]}$ " con probabilidad  $\propto w_t^{[m]}$ 
L7:  $x_t^{[i]} = x_t^{[m]}$ 
L8: end
return  $\{x_t^m\}$ 

```

Figura 1. Filtro de Partículas

La precisión en la estimación depende principalmente del tamaño de las celdas del mapa de ocupación, la precisión del sensor usado y la cantidad de obstáculos en el entorno de navegación. En casos de operación normal (mapas con celdas de 5cm y sensores del tipo laser) el error esperado es menor a 10cm en la estimación de posición, y menor a 8° en la estimación de orientación [7].

b. Planeamiento de trayectorias. Este módulo es el encargado de hallar el camino más óptimo entre la posición actual del robot y la posición deseada. Para esto se usa el mapa de ocupación del entorno de navegación y un método de búsqueda tal como el A* [4].

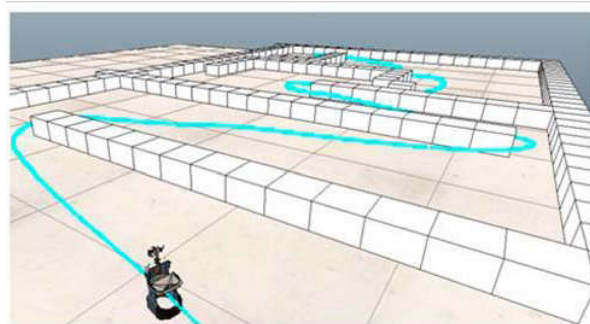


Figura 1. Planificación de trayectorias

En el caso del algoritmo A* el tiempo de generación de trayectorias depende del tamaño de las celdas del mapa de ocupación, y el tamaño del entorno de navegación.

- c. **Evitamiento de obstáculos.** Este módulo es el encargado de determinar si existe algún obstáculo en el camino del robot, y generar una nueva trayectoria en caso fuera necesario.

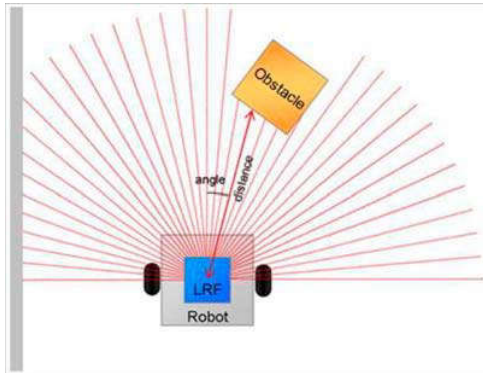


Figura 3. Evitamiento de obstáculos

Este módulo debe detectar si existe la posibilidad que el robot colisione con un obstáculo, y en caso sea así, debe ordenar que el módulo de generación de trayectorias genere un nuevo camino tomando en cuenta la posición del obstáculo. Debe notarse que una presencia masiva de obstáculos podría hacer que el robot falle en la estimación de su posición lo que podría causar que el robot colisione con algún obstáculo. Por eso es importante considerar un sistema de seguridad que desconecte el sistema de alimentación de los motores si esto ocurre (para evitar que los motores y las tarjetas electrónicas se quemen).

- d. **Control de movimiento.** Este módulo es el encargado que el robot siga la trayectoria definida por el módulo de planificación de trayectorias. Es decir debe establecer las velocidades angulares de cada motor del robot.

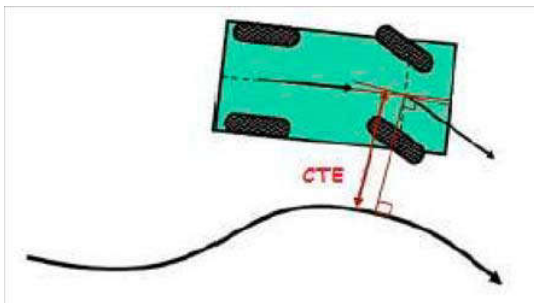


Figura 3. Control de movimiento

Para esto, se puede usar un controlador PID el cual es ampliamente usado en los sistemas de control industriales [5]. Debido a que en los robots móviles se usa una estimación de estado en ciertos intervalos de tiempo, el controlador PID sería del tipo discreto.

Adicionalmente a estos módulos se requiere que el robot tenga un mapa del entorno de navegación. El mapa de ocupación divide el entorno en celdas ocupadas, libres, y no exploradas; y dependiendo del tamaño de la celda se tendrá una mayor o menor precisión en la tarea de localización del robot. Para generar estos mapas se puede usar la librería ROS la cual tiene implementada algoritmos de mapeo muy avanzados. En la figura 5 se muestra el mapa del segundo piso del CTIC usando el paquete *hector_mapping* el cual solo necesita de un sensor láser.

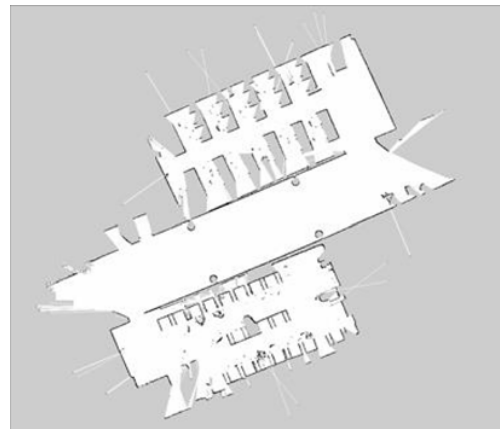


Figura 5. Mapa de ocupación del segundo piso del CTIC-UNI. Además el robot requiere de sensores y actuadores para su navegación autónoma. El sensor más usado es el sensor láser el cual mide las distancias hasta el objeto más cercano. La ventaja de este sensor sobre los ultrasónicos es que es más preciso, tiene un mayor alcance, y tiene un mayor número de mediciones. Para el movimiento del robot se han usado motores DC con cajas de reducción y encoders los cuales se usan para controlar la velocidad de las ruedas.

3. PROTOTIPOS

Componente Mecánico

En la figura 6 se muestran los dos prototipos finales obtenidos en el proyecto.



Figura 6. Robots $R_2D_2-R_1$ y $R_2D_2-R_2$

El robot pequeño, llamado R2D2-R1 es capaz de llevar cargas de hasta 3kg, y el robot grande, llamado R2D2-R2, es capaz de llevar cargas de hasta 25kg. Estos robots llevan baterías de ácido y tienen una autonomía de navegación de dos a tres horas. Ambos robots son del tipo diferencial para que tenga una mejor capacidad de navegación en ambientes reducidos como los ambientes de una casa, empresa, universidad, etc.

Componente Electrónico

El sistema electrónico del robot consta de una tarjeta que se encarga del control de las velocidades de los motores, y de tomar las mediciones de los encoders. Esta tarjeta consta además de los puentes H encargados de suministrar la potencia necesaria a los motores. Los motores DC del robot R2D2-R1 son de 12V, y los motores DC del robot R2D2-R2 son de 24V (Fig. 07).

Esta tarjeta se comunica con una tarjeta Teensy o Arduino, la cual se encarga de recibir los comandos de velocidad enviados desde la PC, y enviar las lecturas de los encoders. En la figura 7 se muestran ambas tarjetas en el caso del robot R2D2-R1. La comunicación con la PC se da a través del puerto USB por lo que la velocidad de transmisión es lo suficientemente alta para controlar el robot en tiempo real. Esta tarjeta además puede usarse para el desarrollo de un sistema de seguridad en el caso que el robot choque con algún obstáculo.

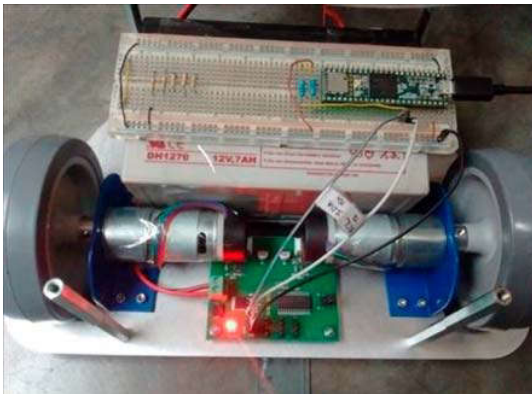


Figura 7. Sistema electrónico del robot R2D2-R1

Sensor láser

Para la navegación del robot se usa frecuentemente un sensor para medir distancias. La ventaja de usar sensores láser sobre otros sensores como los ultrasónicos es que son más precisos y entregan una mayor cantidad de información sobre el entorno. El sensor usado en el proyecto es el Hokuyo UST-10LX el cual tiene un alcance de 10 metros, frecuencia de medición de 25 milisegundos, y un rango de visión de 270°. Se debe notar además que este sensor está optimizado para trabajar en interiores.

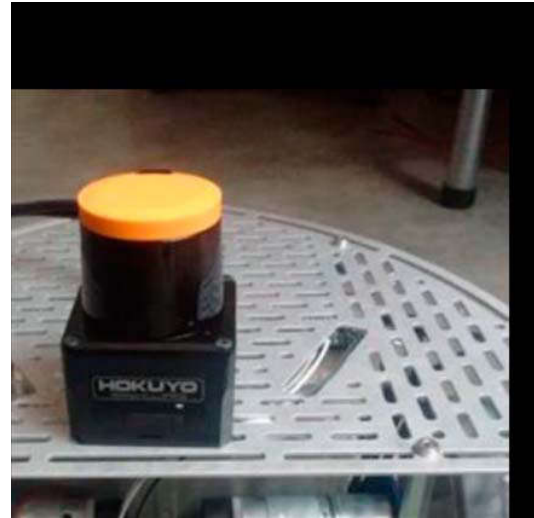


Figura 8. Sensor Láser

Para obtener las mediciones del sensor se puede usar la librería ROS la cual tiene paquetes especializados para controlar su funcionamiento. Estos datos se pueden usar en otros softwares como MATLAB ya que es posible establecer una comunicación entre esos software. La otra opción es desarrollar un propio driver usando el protocolo de comunicación especificado por el vendedor. En la figura 9 se muestra las mediciones obtenidas cuando el robot se encuentra frente al final de un corredor.

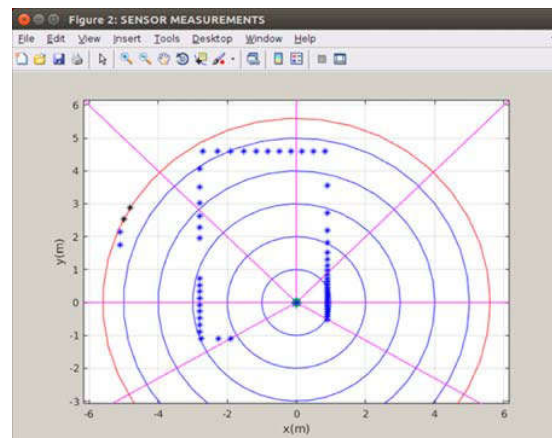


Figura 9. Mediciones del sensor láser

4. RESULTADOS OBTENIDOS

Sistema de Localización

Luego de implementar el filtro de localización descrito en la sección 2, se obtuvieron los resultados a nivel de simulación que se muestran en la figura 10.

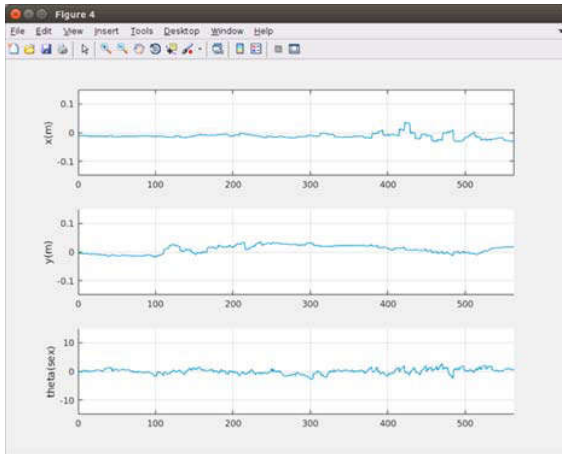


Figura 10. Precisión de la localización

Estos resultados muestran que el error en la estimación de la posición es menor a 5cm, y el error en la estimación de la orientación es menor a 5° . En la navegación real del robot estos errores son algo mayores debido a la presencia de obstáculos y al alcance del sensor. La presencia de obstáculos hace que el robot pueda considerar como pared a algunos obstáculos y hace inútil el uso de las mediciones del sensor. El problema del alcance del sensor se da sobre todo en el caso de ambientes con grandes corredores (fig. 11), ya que en este caso no es posible distinguir la posición del robot en la dirección del corredor usando mediciones de distancia. La solución sería adquirir sensores con un mayor alcance (en el mercado existen sensores para robots con un alcance de hasta 80 metros).

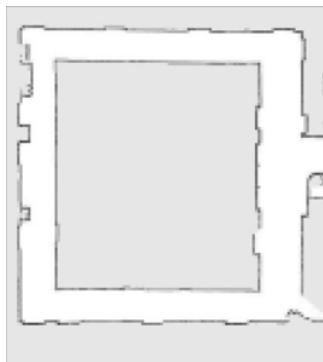


Figura 11. Ambiente con pasadizos extensos

Sistema de Planeamiento

El algoritmo de planeamiento seleccionado fue el A*, debido a que es del tipo óptimo y completo. Tal como se muestra en la figura 11, este algoritmo genera el camino más corto para que el robot llegue a un punto deseado. Nótese que para el planeamiento se usa el mismo mapa de ocupación que se usa en la localización.

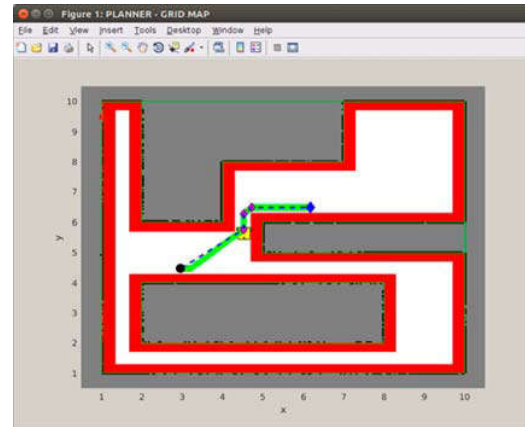


Figura 12. Planeamiento de Trayectorias

Además debe notarse que la trayectoria generada no hace contacto con las paredes del mapa (zona de color negro). Esto es debido a que se aplicó un proceso de dilación a la imagen que representa el mapa con el fin que el robot no golpee las paredes. Esta dilación está representada por la región de color rojo.

Sistema de Evitamiento de Obstáculos

El algoritmo de evitamiento de obstáculos debe ser capaz de detectar la presencia de un obstáculo en el camino del robot, y ser capaz de generar un nuevo camino cuando sea necesario. En la figura 13 se muestra el camino inicial usando el módulo de planeamiento, nótese que los obstáculos mostrados no son considerados en este proceso.

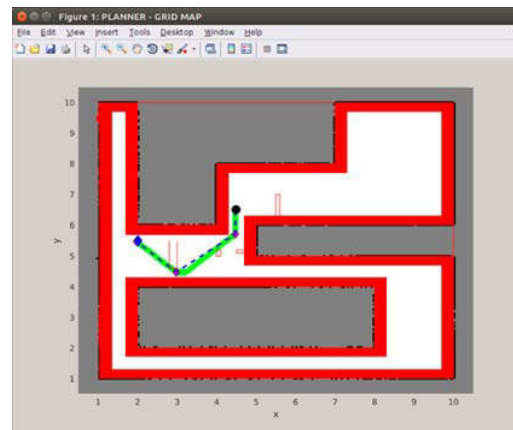


Figura 13. Trayectoria de obstáculos

A medida que el robot avanza llega un momento en que detecta estos obstáculos y procede a activar el módulo de planeamiento para que genera una nueva trayectoria desde la posición actual del robot. Estos resultados se pueden visualizar en la figura 14, donde se observa que la nueva trayectoria generada por el robot evita los obstáculos presentes.

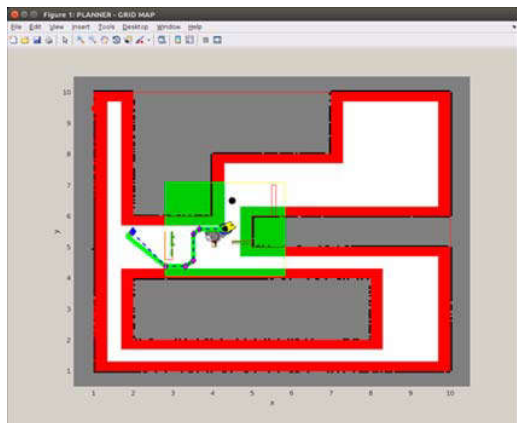


Figura 14. Evitamiento de obstáculos

Sistema de Navegación

Luego de implementar los diversos módulos se procedió a la integración de los mismos en un solo software de navegación autónoma. La arquitectura implementada es la que se muestra en la figura 15.

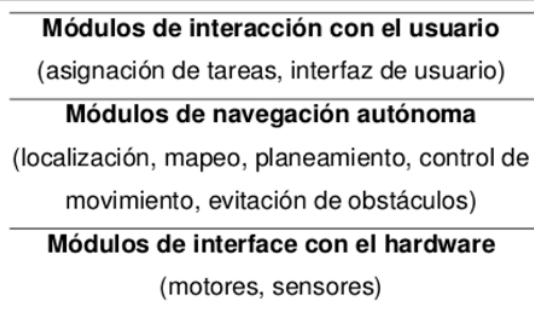


Figura 15. Arquitectura del sistema software

A nivel de usuario la interface de navegación está definida por el siguiente proceso: Primero el usuario establece el mapa de navegación, y activa el sistema de sensores y actuadores. Luego el sistema pide las coordenadas deseadas, aquí el usuario puede establecer varias coordenadas; luego el robot planifica sus trayectorias y empieza el movimiento. En el movimiento del robot se activan el sistema de localización, evitamiento de obstáculos, y control de movimiento. Finalmente cuando el robot cumple todas sus metas, tiene la opción de apagarse, o pedir al usuario nuevas tareas de navegación.

Pruebas de navegación

En la última etapa del proyecto se realizó las pruebas de navegación del robot en diferentes ambientes de la UNI. En todas estas pruebas el robot demostró que es capaz de cumplir con sus tareas de navegación autónoma, no solamente llegando a los puntos deseados, sino también esquivando obstáculos

si estos se presentasen. Cabe señalar que estas pruebas asumen que el terreno es plano, y se tiene un mapa del entorno de navegación.

CONCLUSIONES

Se ha implementado una serie de módulos (localización, planeamiento, evitamiento de obstáculos, control de los sensores y actuadores) que permiten lograr la navegación autónoma de un robot, lo que permite que este pueda usarse en tareas reales de navegación autónoma. Una de estas tareas, es la logística, donde se requiere que el robot lleve paquetes de un lugar a otro.

A nivel de prototipo, se han implementado los robots R2D2-R1 Y R2D2-R2. Donde el primero es capaz de llevar cargas de hasta 2.5kg, y el segundo es capaz de llevar cargas de hasta 25kg. Siendo el nivel de autonomía de ambos robots de aproximadamente 2 horas.

Se ha evaluado la navegación autónoma de los robots en diferentes ambientes de la UNI, en donde han demostrado que son capaces de realizar las tareas establecidas, y debido a su sistema de evitamiento de obstáculos es capaz de navegar en ambientes reales.

Queda como trabajo futuro adicionar otros tipos de sensores para aumentar las capacidades del robot. Por ejemplo se puede implementar un sistema de visión para hacer experimentos de reconocimiento y detección de objetos. También se puede usar un sistema de profundidad como el sensor Kinect, esto para que el robot pueda detectar obstáculos en 3D. Esto es muy importante ya que el sistema de detección de obstáculos al estar basado en el láser solo funciona en un plano 2D (Fig. 16), sin embargo en tareas reales el obstáculo podría estar encima o debajo de ese plano de detección.

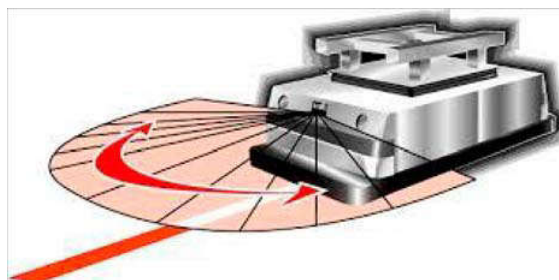


Figura 16. Zona de detección de obstáculos (Fuente: <https://en.manu-systems.com/HOK-PBS-03JN.shtml>).

AGRADECIMIENTOS

Al Instituto General de Investigación de la UNI por el financiamiento del presente trabajo (IGI-CALLE.F-2016).

BIBLIOGRAFÍA

- [1] Siegwart R, Nourbakhsh I, Scaramuzza D 2011 Introduction to Autonomous Mobile Robots. 2nd Edition. MIT Press. Cambridge.
- [2] Hassler, S. (Publicada 30/12/16). <https://spectrum.ieee.org/transportation/advanced-cars/2017-the-year-of-selfdriving-cars-and-trucks> IEEE SPECTRUM
- [3] Thrun S, Burgard, W., Fox, D. 2005 Probabilistic Robotics MIT Press, Cambridge.
- [4] Russell S, Norvig P, 2009. Artificial Intelligence: A Modern Approach. 3rd Edition. Pearson.
- [5] Nise N. 2015. Control Systems Engineering. 7th Edition. Wiley
- [6] Quigley M, Gerkey B, Smart W 2015 Programming Robots with ROS: A Practical Introduction to the Robot Operating System. O'Really.
- [7] Calle I. 2014 Navegación Autónoma de un Robot Móvil usando Técnicas Probabilistas de Localización y Mapeo Basadas en Métodos Monte Carlo Secuenciales. Tesis UNI-FIEE Posgrado. Lima.