

Solución de un laberinto (conexo) con un robot móvil terrestre.

1st Héctor Aguilar Galván
Ingeniería en Mecatrónica.
DICIS
h.aguilargalvan@ugto.mx

2nd Víctor M. Castellano Ornelas
Ingeniería en Mecatrónica.
DICIS
vm.castellanoornelas@ugto.mx

3rd Zayra D. García Chávez
Ingeniería en Mecatrónica.
DICIS
zd.garciachavez@ugto.mx

4th Iván A. Navarro Munguía
Ingeniería en Mecatrónica.
DICIS
ia.navarromunguia@ugto.mx

5th Edgar A. Valdés Zamilpa
Ingeniería en Mecatrónica.
DICIS
ea.valdeszamilpa@ugto.mx

Resumen—Se pretende describir por medio del presente documento la programación (algoritmo para evasión y avance del robot), tanto como la simulación del mismo, mencionando las consideraciones que se tomaron para cumplir con el propósito del proyecto; el cual consiste en el escape del robot de un laberinto conexo a partir de cualquier punto inicial del mismo.

Index Terms—

I. INTRODUCCIÓN.

El presente reporte describirá una rutina de programación realizada en el lenguaje Python para la solución de un laberinto conexo, por medio de un robot móvil terrestre Pioneer 3dx [1], ambientado en el entorno de simulación de CoppeliaSim.

El reto será que el robot en cuestión logre salir eventualmente de cualquier laberinto, a partir de cualquier punto en el que se le coloque; apoyado por un gama de sensores de proximidad que serán útiles para detectar y medir los muros que tiene a su alrededor.

La solución para este planteamiento será por medio del algoritmo conocido como “De la mano derecha” [2], el cual básicamente pretende que el robot siga los muros que estén siendo detectados por los sensores del lado derecho, con lo cual, como resultado lógico, hará que el Pioneer 3dx encuentre una salida del laberinto conexo tarde o temprano.

II. METODOLOGÍA.

II-A. Algoritmo de solución.

Como se puede observar en la figura 1 el proceso comienza inicializando los sensores y obteniendo la posición y la rotación del Pioneer 3dx respecto del mundo, posteriormente se miden los sensores y se cuestiona si los sensores posicionados a la derecha del robot detectan algún objeto, en caso de que NO se detecte ningún objeto gira 90° en el sentido de las manecillas del reloj y avanza recto en la dirección en la que este “mirando” el robot.

Por otro lado, si los sensores detectan algún objeto a la derecha el algoritmo revisará si en los sensores ubicados al frente existe algún objeto, en caso de que estos sensores no detecten nada el algoritmo le ordenará al robot avanzar en

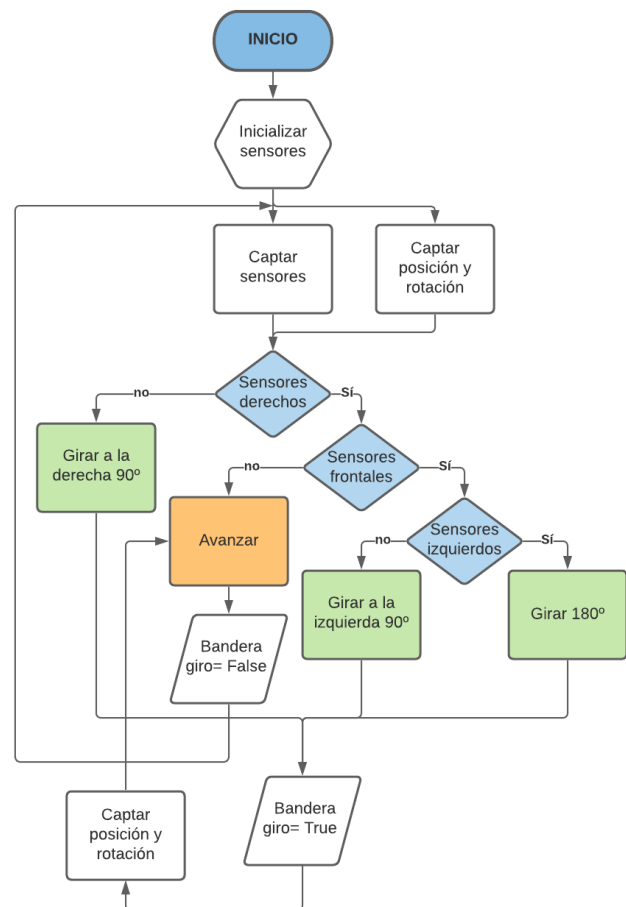


Figura 1. Diagrama de flujo

línea recta en la misma dirección en la que venía.

Para el caso en el que se detecta algún objeto a la derecha y al frente del robot se medirán los sensores dispuestos a la izquierda de este mismo, si los sensores no se activan esto

nos indica que el robot puede girar en esa dirección y como previamente se comprobó que no se puede ir a ningún otro lado se le da la orden de que gire 90° en dirección contraria a las manecillas de reloj y que avance una cierta distancia ignorando posibles giros.

Finalmente, si el robot no puede girar a ningún lado o avanzar, esto indicaría que llegó a un callejón sin salida, se indica que efectúe un giro de 180° y avance una determinada distancia para evitar ese callejón.

Como ya se menciona cada vez que se efectúa un giro el robot avanza ignorando giros posibles, esto se hace para evitar que el programa realice dos giros en la misma vuelta, esto es posible dado que existe una bandera que ignora los giros si previamente se realizó uno.

II-B. Implementación del algoritmo.

El entorno de simulación CoppeliaSim fue donde se llevaron a cabo las pruebas necesarias para comprobar el funcionamiento de funcionamiento del algoritmo, desde observar el comportamiento del robot a través de pseudocódigo probado, hasta hacer cálculos y pruebas para implementar correctamente el algoritmo. Este entorno es bastante útil pues permite la manipulación de los recursos para la realización de este proyecto; robot Pioneer 3dx y diferentes bloques necesarios para formar el laberinto.

II-B1. Laberinto.: Se construyó el siguiente laberinto para realizar las pruebas necesarias y comprobaciones, pensando en que el robot hiciera un recorrido largo y pasara el mayor tiempo y cantidad de casos posibles antes de lograr abandonar el laberinto, como se puede observar, se utilizaron bloques pequeños y algunos más grandes (de contención) para formar en su totalidad el laberinto, además de consideró una anchura de los pasillos pensada en que el robot Pioneer 3dx pueda avanzar y rotar sin problemas de espacio.

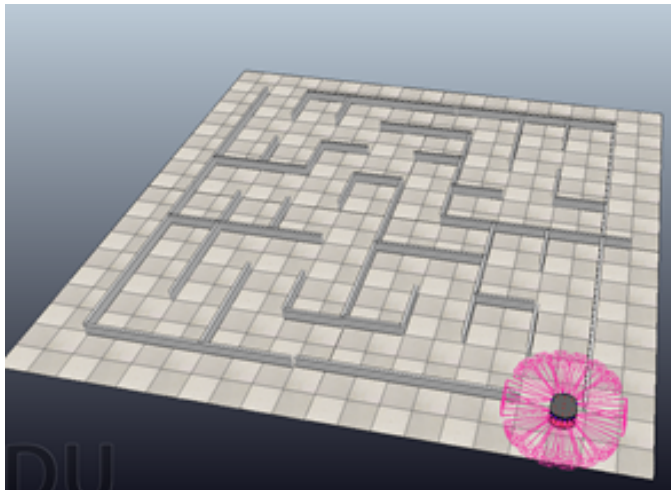


Figura 2. Laberinto de pruebas.

II-B2. Robot: El robot Pioneer DX-2 y su sucesor Pioneer 3dx son plataformas populares para educación, investigación, creación de prototipos, exhibiciones entre otros proyectos. El

Pioneer 3dx consta de dos llantas en sus laterales las cuales pueden accionar sus ejes de manera independiente esto con el fin de que el robot pueda rotar a cualquier dirección y sobre su propio eje, dicho robot cuenta con sensores a su alrededor que son anillos de sonares los cuales sirven para detectar a cierta distancia un objeto. En general el Pioneer 3dx es un robot móvil terrestre el cual puede ser útil para censar el entorno en el que se rodea además de poder trasladarse libremente en cualquier plano, por lo cual es el candidato perfecto para la realización de este proyecto.



Figura 3. Robot para simulación P-3dx

entonces bien, el algoritmo presentado en la subsección anterior es implementado en los recursos aquí mencionados, para posteriormente presentar a continuación los resultados obtenidos.

III. RESULTADOS

Este caso es el más sencillo de explicar ya que como se puede apreciar en la captura de la consola, cuando el robot se encuentra un pasillo a la derecha este gira en dicha dirección cabe destacar que el giro se efectuara aunque el robot aunque sea capaz de seguir avanzando.

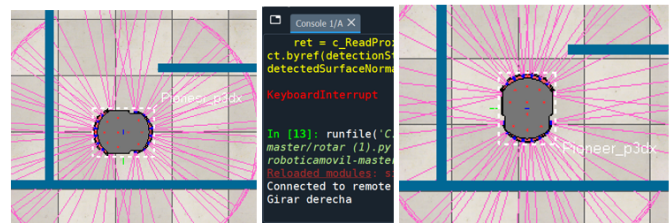


Figura 4. Giro a la derecha.

El caso presentado en la figura 5, puede considerarse un caso conflictivo ya que el robot debe ser capaz de elegir a qué lugar girar de manera coherente ya que si gira a la izquierda

rompería la lógica del programa, pero como se puede apreciar el robot elige de manera correcta girar a la derecha para posteriormente girar a la izquierda para poder continuar con el recorrido cumpliendo a condición de avanzar si el muro está a la derecha.

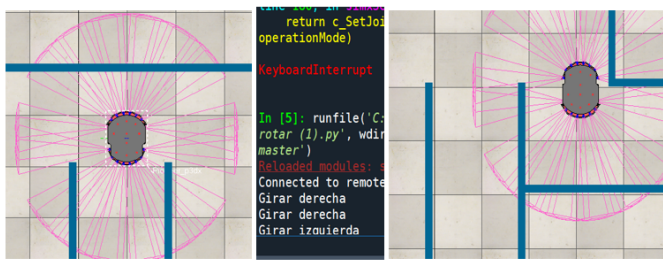


Figura 5. Giro a la derecha en T.

Igualmente, como ocurre en el caso anterior se puede observar como el robot primeramente ignora el posible giro a la izquierda debido a que este tiene como prioridad avanzar si existe un muro a su derecha, aunque también es posible ver que una vez que se cumple la condición en la que no es posible avanzar ni girar a la derecha el robot opta por girar a la izquierda.

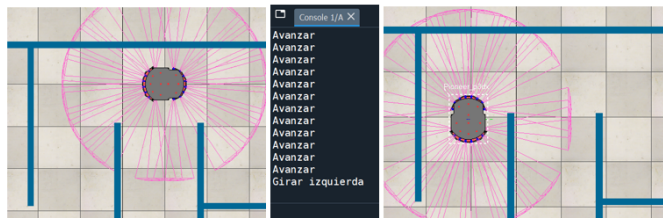


Figura 6. Giro a la izquierda en T.

Finalmente, el caso más extremo es en el cual el robot se encuentra con un callejón sin salida, esto significa que no puede avanzar ni efectuar ningún de 90° , por esta razón se le indica al Pioneer 3dx realizar un giro de 180° para poder continuar avanzando, cabe destacar que siempre que realiza dicho giro siempre esta apuntando a la salida del callejón y tendrá en todo momento una pared del lado derecho.



Figura 7. Giro de 180° para salir de un callejón.

IV. DISCUSIÓN.

De acuerdo con lo mostrado en la simulación en el entorno Coppeliasim, se revisó que el algoritmo programado en el

lenguaje Python para lograr el escape de un robot Pioneer 3dx cumple con su objetivo; evade de forma eficiente, avanzando siempre según su posterior derecho gracias a la jerarquía de posiciones programada, manteniéndose en la medida de lo posible en un avance recto para evitar perder la ruta o posteriormente colisionar con alguno de los muros del laberinto. Pues bien, al paso de aproximadamente 9 minutos logra solucionar por completo (atravesar de inicio a fin) un laberinto conexo relativamente complejo, el cual se podría plantear como uno de los casos más difíciles que se le pueden presentar al robot previamente programado, pues así abarca prácticamente todos los posibles casos de evasión que pueda haber en un laberinto conexo cuadrado. Entonces bien, se concluye que se cumplió satisfactoriamente con el objetivo del proyecto planteado.

REFERENCIAS

- [1] M. Inc. (2021) Worlds most popular intelligent wheeled robot. [Online]. Available: <http://www.activrobots.com/ROBOTS/p2dx.html>
- [2] Z. S. Niemczyk R. (2020) Review of maze solving algorithms for 2d maze and their visualisation. [Online]. Available: https://doi.org/10.1007/978-3-030-13321-4_2