

## Navegación autónoma de robot móvil diferencial con algoritmo $A^*$ Differential mobile robot autonomous navigation with algorithm $A^*$

E. Baizabal-Montañez <sup>a,\*</sup>, N. Gonzalez-Yances <sup>1</sup>, A. Morales-Díaz <sup>1</sup>, K. Ordaz-Hernández <sup>1</sup>

<sup>a</sup>Grupo de Robótica y Manufactura Avanzada, Centro de Investigación y de Estudios Avanzados del IPN-Salttilo, Industrial, Zona Industrial, 25900 Ramos Arizpe, Coahuila, México.

### Resumen

Con la finalidad de que un robot móvil diferencial pueda navegar en entornos con obstáculos, en el presente trabajo se aplica el algoritmo  $A^*$ , uno de los algoritmos más populares para encontrar el camino más corto entre dos nodos de un grafo. Debido a la restricción no-holónoma que poseen éste tipo de robots, la trayectoria obtenida con el algoritmo  $A^*$  se post-procesa, para que posteriormente sea ejecutada por el robot mediante un controlador no-lineal aquí propuesto. También se presenta la estabilidad de dicho controlador haciendo uso del método de Lyapunov. Finalmente el desempeño del robot es probado en experimentos.

**Palabras Clave:** Robótica Móvil, Planeación de Trayectorias, Optimización, Control No Lineal.

### Abstract

In order for a differential mobile robot to navigate in environments with obstacles, this paper applies the  $A$  algorithm, one of the most popular algorithms to find the shortest path between two nodes of a graph. Due to the nonholonomic constraint of this type of robots, the path obtained with  $A$  algorithm is post-processed, to be subsequently executed by the robot by means of a nonlinear controller proposed here. The stability of this controller is also presented using the Lyapunov method. Finally, the performance of the robot is tested in experiments.

### Keywords:

Mobile Robotics, Path Planning, Optimization, Non-linear Control.

## 1. Introducción

El creciente interés en los robots móviles se debe a las nuevas aplicaciones que tienen en el área médica (Xiaofei y Gang, 2018), industrial (Yuki y Shuki, 2022), agrícola (Luis y Herrea-Díaz, 2022) y doméstica (Tatsuya y Yuki, 2022); en éstas áreas se requiere que los robots posean un nivel de autonomía para navegar en entornos obstaculizados desde una posición inicial a un objetivo. Para que los robots puedan realizar sus tareas en distintas zonas del entorno, es necesario que lo conozcan y encuentren diversos caminos entre 2 puntos a través de un entorno complejo (laberinto), para posteriormente seleccionar el más corto y seguirlo.

El problema de la planeación y optimización de trayectorias en entornos complejos se ha abordado por métodos de búsqueda en gráficos como lo son  $A^*$ , Dijkstra y  $D^*$ , aunque también existen otros métodos basados en algoritmos genéticos y cam-

pos potenciales.

En el presente trabajo se propone una estrategia de planeación de rutas basada en grafos, capaz de comparar el costo de cada camino encontrado y seleccionar el más corto (S. Gunawan y Cahyadi, 2019) en un entorno estructurado mediante el algoritmo  $A^*$ . Para que un robot diferencial siga el camino seleccionado se propone una estrategia de ajuste de ruta y un controlador no lineal.

## 2. Estrategia de planificación de ruta

La búsqueda de caminos es un problema que usualmente se resuelve por medio de algoritmos de búsqueda en grafos; cómo lo son Dijkstra y  $A^*$ . En este tipo de problemas se conoce el estado inicial, el estado final (meta) y un conjunto de reglas mediante las cuales se pueden realizar desplazamientos para que

\* Autor para correspondencia: edgar.baizabal@cinvestav.mx

**Correo electrónico:** edgar.baizabal@cinvestav.mx (Edgar Baizabal Montañez), neftali.gonzalez@cinvestav.mx (Nefthalí J. González-Yances), america.morales@cinvestav.edu.mx (América B. Morales Díaz), kenny.ordaz@cinvestav.mx (Kenny Ordaz-Hernández).

**Historial del manuscrito:** recibido el 28/10/2022, última versión-revisada recibida el 28/10/2022, aceptado el 07/11/2022, publicado el 11/11/2022. **DOI:** <https://doi.org/10.29057/icbi.v10iEspecial.10145>



un agente o robot pueda trasladarse en un grafo compuesto por nodos. Eventualmente, avanzando por el grafo, se llegará al nodo que corresponde a la meta. Finalizando así, la búsqueda y logrando obtener el conjunto de desplazamientos que lleven al agente de un punto a otro. El algoritmo  $A^*$  es uno de los preferidos para este tipo de aplicaciones; debido a la versatilidad que existe en la definición de su heurística. Entendiendo como heurística, a la posibilidad de poder definir y proponer, dentro del algoritmo, diversas formas para encontrar el camino más corto y libre de obstáculos.

### 2.1. Algoritmo $A^*$

$A^*$  es un algoritmo de búsqueda inteligente y respaldado por información que busca el camino más corto desde un estado inicial al estado meta usando una heurística óptima (Takeyas, 2005).

Este algoritmo requiere saber la posición inicial del agente, así como el punto de destino y las coordenadas de los obstáculos existentes en el entorno. Su comportamiento se basa en la evaluación de la siguiente función:

$$f(n) = g(n) + h(n)$$

donde:

$n$  : conjunto de todos los nodos adyacentes a explorar con respecto al nodo actual.

$h(n)$  : función heurística.

$g(n)$  : función de costo asociado al camino entre el primer nodo y el nodo actual. Usualmente se le asigna el mismo comportamiento de la función heurística.

$f(n)$  : función del costo total del camino.

Existen diversas funciones heurísticas que se pueden implementar en  $h(n)$  tales como: la distancia Euclidiana, distancia Manhattan, distancia Chebyshev y distancia Octile (K. Karur y Siegel, 2021). El presente trabajo implementa una heurística basada en la norma  $L_1$  que permite encontrar el camino más corto desde el nodo de inicio hasta el nodo final, considerando que el agente sólo puede desplazarse en 4 direcciones con respecto a su ubicación actual, las cuales son: {atrás, enfrente, izquierda, derecha}. En el **Algoritmo 1** se muestra un pseudocódigo utilizado del Algoritmo  $A^*$ .

---

#### Algoritmo 1: Algoritmo $A^*$

---

**Resultado:** Lista cerrada de nodos que representan el camino más corto

Declaración de lista abierta y lista cerrada.

$N_{actual}$  = Nodo actual

$N_{menor}$  = Nodo con menor  $f(n)$

$N_{adyacente}$  = Nodo adyacente

Lista abierta  $\leftarrow$  Nodo inicial

**para cada**  $N_{adyacente}$  **hacer**

$N_{menor} \leftarrow$  Nodo con menor  $f(n)$   $N_{actual} \leftarrow N_{menor}$

Lista cerrada  $\leftarrow N_{actual}$ ;

**para para cada**  $N_{adyacente}$  a  $N_{actual}$  **hacer**

**si es obstáculo ó está en lista cerrada entonces**

Ignorar nodo;

**fin**

**si No está en lista abierta entonces**

Lista abierta  $\leftarrow N_{adyacente}$  Calcular:

$N_{adyacente} \leftarrow g(n), h(n), f(n)$

$N_{actual}$  se vuelve  $N_{padre}$  de  $N_{adyacente}$

**en otro caso**

Revisar si el camino con  $N_{actual}$  es mejor.

Usando como métrica a  $g(n)$  **si Verdadero**

**entonces**

$N_{actual} \leftarrow N_{padre}$

Recalcular:

$N_{actual} \leftarrow g(n), f(n)$

**fin**

**fin**

**fin**

**si Se añade  $N_{meta}$  a Lista cerrada entonces**

Camino ha sido encontrado

**Fin del programa**

**fin**

**si  $N_{meta}$  no se encontró entonces**

No es posible encontrar un camino

**Fin del programa**

**fin**

**fin**

---

### 2.2. Esquema de ajuste de ruta

El algoritmo  $A^*$  genera rutas que no son derivables, lo cual presenta un problema para un robot diferencial que pretenda seguir todos los puntos de ese camino. Esto es debido a que cada cambio en la dirección de movimiento demanda realizar un giro de  $90^\circ$ , y esto requiere de gran cantidad de energía para cambiar la orientación del robot en un período de tiempo muy corto, lo que representa un problema en la implementación debido a las limitaciones de los motores del robot.

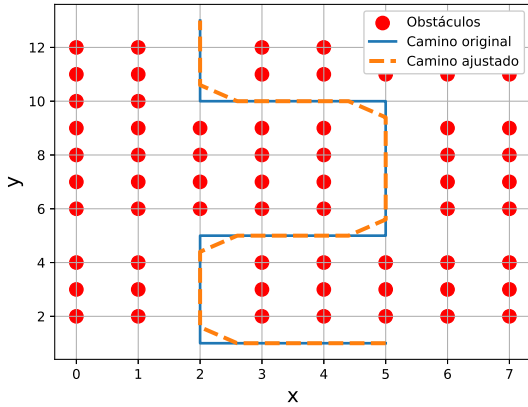


Figura 1: Comparación de ruta con A\* original y ruta ajustada.

Con la finalidad de obtener rutas que no exijan grandes cambios de orientación de forma abrupta, se propone un post-procesamiento de la ruta, que evita los giros de  $90^\circ$  al agregar un ajuste mediante un recorte de esquinas en el camino obtenido con A\*. En la Figura (1) se muestra una comparación entre la ruta original (color azul) y la ruta ajustada (color naranja), la cual se implementó para brindar al robot un comportamiento más suave en los cambios de dirección.

### 3. Modelo cinemático de robot móvil diferencial

Como se ha mencionado, la navegación se realizó con un robot diferencial (Figura 2), cuyo modelo cinemático es representado como se presenta en (1):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (1)$$

donde  $[x, y, \theta]^T$  representan la posición y orientación del centro de masas del robot en el plano. Las velocidades lineal y angular del robot están representadas por  $v$  y  $\omega$ , respectivamente, las cuales son las entradas de control. El sistema (1) es un sistema sub-actuado ya que posee tres estados y sólo dos entradas de control; por lo que representa un sistema de construcción simple y control complejo.

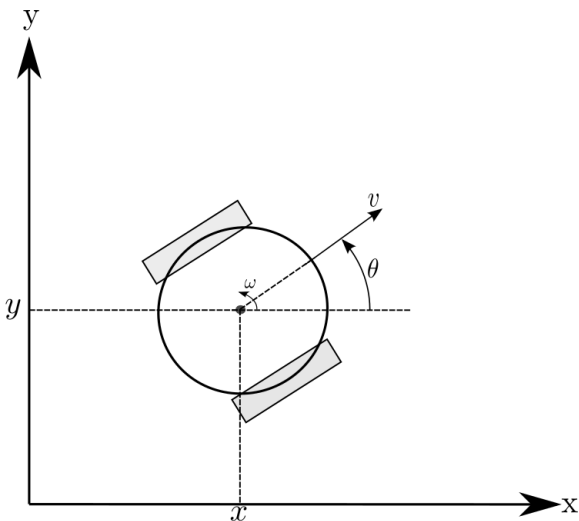


Figura 2: Diagrama de robot diferencial.

Las llantas del robot imponen una restricción al movimiento del mismo; no permitiendo el deslizamiento en dirección lateral. Dicha restricción puede ser expresada como:

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0. \quad (2)$$

### 4. Ley de control

Considerando (1) se propone una ley de control no lineal para resolver un problema de regulación a los puntos de la ruta ajustada mostrada en la Figura 1. Considerando una pose deseada  $[x_d, y_d, \theta_d]^T$  y la pose actual  $[x, y, \theta]^T$ , se definen los errores en el marco de referencia local del robot de la siguiente forma:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \mathbf{R}^T(\theta) \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}, \quad (3)$$

donde  $\mathbf{R}(\theta)$  es una matriz de rotación definida de la forma:

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

Obteniendo la dinámica de (3)

$$\begin{aligned} \dot{e}_x &= \omega e_y - v \\ \dot{e}_y &= -\omega e_x \\ \dot{e}_\theta &= -\omega \end{aligned} \quad (5)$$

Considerando (5) se propone el siguiente control no lineal:

$$\begin{aligned} v &= k_x e_x, \\ \omega &= \frac{\sin(e_\theta)}{e_\theta} k_y e_y + k_\theta e_\theta. \end{aligned} \quad (6)$$

Donde  $k_x, k_y, k_\theta$  son las ganancias de control y son definidas positivas. El término  $\frac{\sin(e_\theta)}{e_\theta}$  es usado para suavizar el giro del robot cuando ocurre un cambio de dirección. Esto quiere decir que una vez que el robot se encuentre en la orientación indicada, la entrada de control  $\omega$  sólo dependerá de  $e_y$ ; esto es debido a que por regla de L'Hopital se tiene que:

$$\lim_{e_\theta \rightarrow 0} \frac{\sin(e_\theta)}{e_\theta} = 1. \quad (7)$$

para cuestiones prácticas se establece que cuando  $|e_\theta| \leq \delta$ ,  $\frac{\sin(e_\theta)}{e_\theta} = 1$

#### Prueba de estabilidad

Para probar la estabilidad del controlador (6) en lazo cerrado con el sistema (1) se propone la siguiente función de la Lyapunov:

$$V(e_x, e_y, e_\theta) = \frac{1}{2} e_x^2 + \frac{1}{2} e_y^2 + \frac{1}{2} e_\theta^2, \quad (8)$$

derivando (8) se obtiene:

$$\dot{V}(e_x, e_y, e_\theta) = e_x \dot{e}_x + e_y \dot{e}_y + e_\theta \dot{e}_\theta. \quad (9)$$

sustituyendo (5) en (9) se obtiene:

$$\dot{V}(e_x, e_y, e_\theta) = -v e_x - e_\theta \omega. \quad (10)$$

sustituyendo las entradas de control (6) en (10):

$$\dot{V}(e_x, e_y, e_\theta) = -k_x e_x^2 - \sin(e_\theta) k_y e_y - k_\theta e_\theta^2 \quad (11)$$

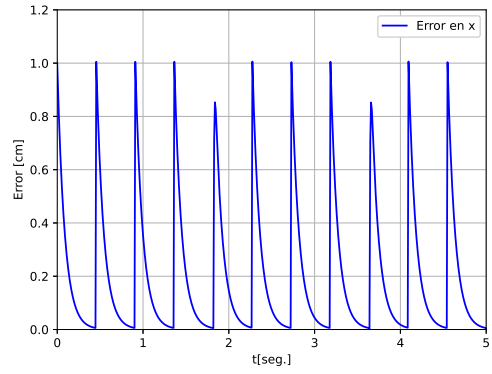
reescribiendo (11):

$$\begin{aligned} \dot{V}(e_x, e_y, e_\theta) = & -k_x e_x^2 - \gamma k_y e_y^2 - k_\theta e_\theta^2 + \gamma k_y e_y^2 \\ & - \sin(e_\theta) k_y e_y \end{aligned} \quad (12)$$

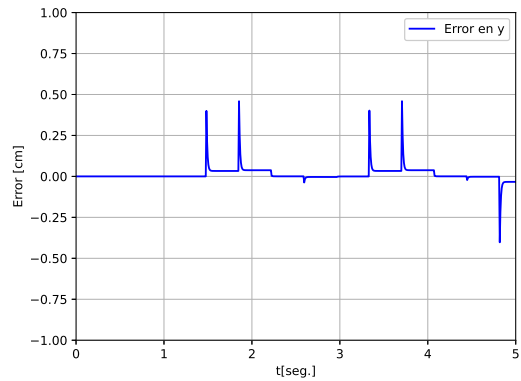
acotando (12):

$$\dot{V}(e_x, e_y, e_\theta) \leq -k_x e_x^2 - \gamma k_y e_y^2 - k_\theta e_\theta^2 + \gamma k_y e_y^2 - k_y e_y. \quad (13)$$

En (13) se cumple que  $-k_x e_x^2 - \gamma k_y e_y^2 - k_\theta e_\theta^2 \leq 0$ ; sin embargo, no se puede garantizar que  $\gamma k_y e_y^2 - k_y e_y \leq 0$ . Por lo cual, éste no es un control que tenga estabilidad asintótica global; sin embargo se cumple que  $V(e_x, e_y, e_\theta) \leq 0$  sí y sólo sí  $e_y \geq \frac{1}{\gamma}$ , donde  $0 < \gamma$ ; Si  $\gamma \rightarrow \infty$  Entonces  $e_y \rightarrow 0$ , Por lo que el control es últimamente acotado de acuerdo con (Khalil, 2015).



(a) Error en x.



(b) Error en y.

Figura 4: Errores.

## 5. Resultados

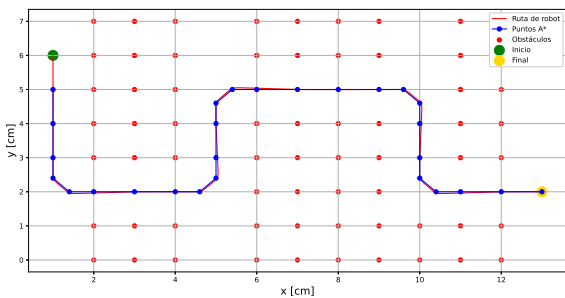


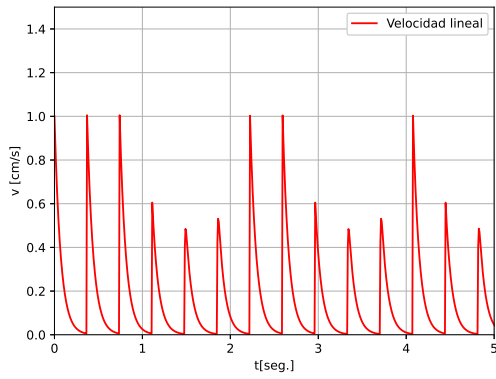
Figura 3: Ruta A\* recortada y ruta seguida por el robot

Se obtuvieron resultados de la implementación de algoritmo A\* con el control no lineal (6) tanto a nivel simulación y experimental.

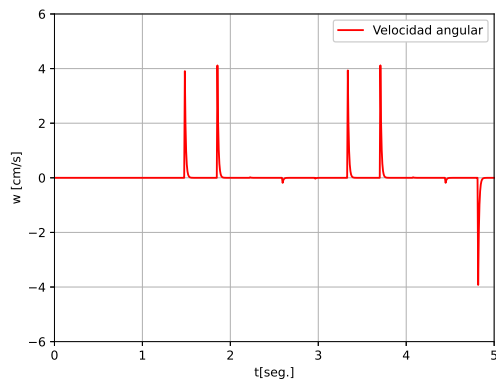
### Simulaciones

La simulación se realizó en un espacio representado por una matriz de 8 renglones y 15 columnas (ver Figura 3); donde se ubicaron 66 obstáculos, es decir 66 nodos por los cuales el robot no puede transitar. Las ganancias del controlador (6) fueron:  $k_x = 1$ ,  $k_y = 0,001$   $k_\theta = 5$ . En la Figura 3 se observan (en azul) los puntos obtenidos de la ruta A\* recortada, la ruta seguida por el robot (en rojo) así como los puntos de inicio y final (verde y amarillo, respectivamente); Nótese la cercanía entre la ruta seguida por el robot y la obtenida mediante A\*. En la Figura (4a) se observa  $e_x$ , el cual es un error que converge a cero de forma periódica conforme el robot se va aproximando a cada uno de los puntos calculados con A\*. Por otro lado,  $e_y$  (Figura 4b); el cual depende directamente de la entrada de velocidad angular. Éste un error diferente de cero cuando el agente realiza un cambio de dirección.

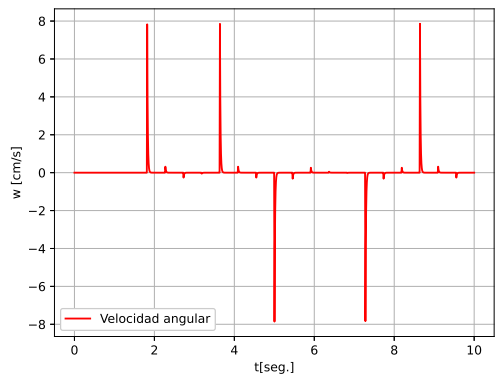
Las entradas de control de velocidad lineal y angular mostradas en las Figuras (5a) y (5b) mantienen un comportamiento similar a los errores mostrados en la Figura (4). Esto es debido a que las velocidades tienden a disminuir conforme el agente se acerca a cada uno de los puntos que definen la ruta obtenida con A\*.



(a) Velocidad lineal.



(b) Velocidad angular.



(c) Velocidad angular sin ajuste de ruta.

Figura 5: Entrada de control.

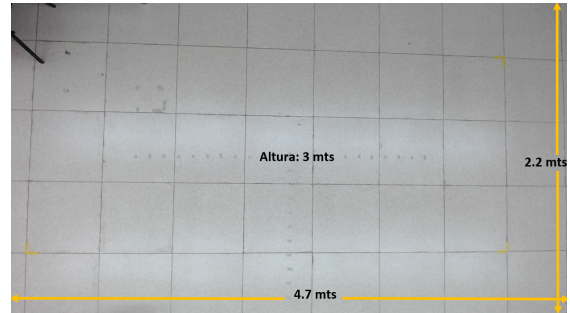
Por motivos de comparación en la Figura 5c se presenta la velocidad angular del robot al recorrer la ruta obtenida mediante  $A^*$  sin el ajuste que recorta las esquinas, mostrando que las velocidades requeridas son 2 veces mayores en magnitud con respecto a las velocidades angulares requeridas al seguir la ruta recortada (Figura 5b)

*Experimentos*

Los experimentos se reliazaron con una cámara Logitech C920 que posee una resolución de 1280 x 720 píxeles a una tasa de refresco de 30 cuadros por segundo, ver Figura (6a). La cámara está situada en el techo a 3 metros de altura, proporcionando un área de trabajo de  $10.34 m^2$  como se observa en la Figura (6b).



(a) Cámara usada.



(b) Dimensiones de set experimental.

Figura 6: Set experimental.

La Figura (7) muestra el robot móvil diferencial utilizado. En la parte superior del robot se colocó un marcador en blanco negro tipo Aruco (Muñoz y Garrido, 2018) para medir la pose del robot y establecer el lazo cerrado de control que permite el seguimiento de los puntos ajustados obtenidos con el algoritmo  $A^*$ . Las salidas del controlador (6) se encuentran relacionadas con las velocidades de las ruedas izquierda  $\omega_l$  y derecha  $\omega_r$  por la siguiente función:

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \frac{1}{2R} \begin{bmatrix} 2 & L \\ 2 & -L \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (14)$$

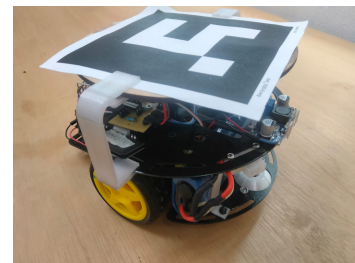


Figura 7: Robot móvil diferencial.

Se exportó el entorno mostrado en la Figura (3) al área de trabajo del set experimental, véase Figura 8 y se dibujaron todos los nodos (cuadros azules), obstáculos virtuales (cuadros rojos) y el camino calculado previamente con  $A^*$  (Puntos verdes). El entorno se dibujó con las librerías de OpenCv de python.

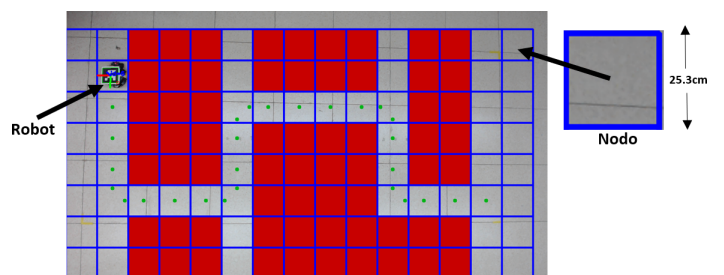


Figura 8: Ambiente estructurado virtual.

El experimento de la navegación representada en la Figura (8) dió como resultado la regulación a los puntos de  $A^*$  que se muestra en la Figura (9). El algoritmo de control y envío de datos al robot se programaron en python. Las ganancias utilizadas en la implementación del controlador (6) fueron:  $k_x = 1,5$ ,  $k_y = 0,3$  y  $k_\theta = 0,6$ . Para realizar las velocidades de rueda derecha e izquierda (14), se cuenta con un control PID que actúa sobre la velocidad angular de las ruedas, para esto se utilizaron las ganancias:  $k_p = 341,25$ ,  $k_d = 1,5$  y  $k_i = 1,5$ . La Figura (9) presenta la ruta que siguió el robot con respecto a cada uno de los puntos previamente ajustados y calculados con  $A^*$ . El resultado muestra que el robot pasa por la mayoría de los puntos presentando algunos errores.

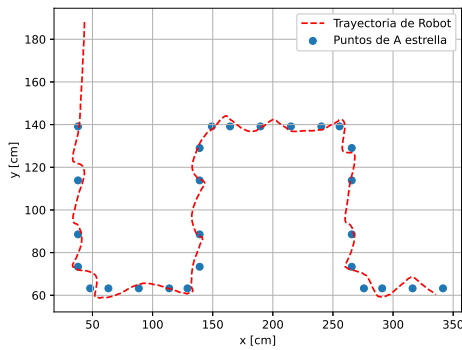


Figura 9: Secuencia de puntos obtenidos por el robot durante el experimento.

Las Figuras (10) y (11) muestran  $e_x$  y  $e_y$  respectivamente de la ruta que se obtuvo en la Figura (9). La Figura (10) presenta cómo  $e_x$  tiende a aproximarse a cero una vez que el robot llegó a cada uno de los puntos. Debido a que al asignar un nuevo punto a regular, el error tiende a incrementar y nuevamente a disminuir conforme el robot llega a este nuevo punto deseado de la secuencia. En  $e_y$  se obtuvieron valores negativos ya que este error está asociado con la orientación del robot de acuerdo al algoritmo de control presentado en la ecuación (6). Las oscilaciones de este error, indican que se requiere una mejora en la sintonización del controlador PID a bordo del robot

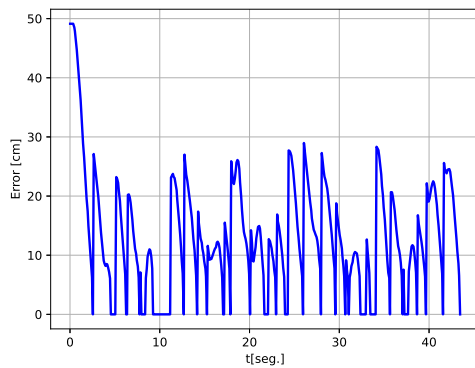


Figura 10: Error en x.

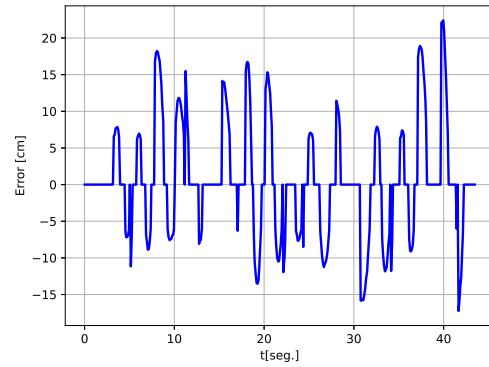


Figura 11: Error en y.

## 6. Conclusiones

En este artículo, se propone un esquema de navegación autónoma de un robot móvil diferencial implementando el algoritmo  $A^*$  con ajuste de esquinas como planificador de rutas libres de obstáculos en un entorno virtual. Se observó que el control de regulación propuesto es capaz de seguir el conjunto de puntos ajustado y obtenidos mediante  $A^*$  de forma satisfactoria en simulación y aceptable en el set experimental.

Como trabajo futuro se propone mejorar el set experimental; transformar el problema de regulación a varios puntos, a un problema de seguimiento de una trayectoria continua y agregar restricciones que consideren evasión de obstáculos. Aunado a esto, se encuentra la propuesta de trabajar con entornos más realistas compuestos por un mayor número de nodos; que den pauta a probar la estrategia de navegación con otras funciones de costo.

## Referencias

- K. Karur, N. Sharma, C. D. y Siegel, J. (2021). A survey of path planning algorithms for mobile robots. *vehicles*.
- Khalil, H. (2015). *Nonlinear Control*. Pearson.
- Luis, E. y Herrea-Diaz, J. (2022). Toward autonomous mobile robot navigation in early-stage crop growth. *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.
- Muñoz, R. y Garrido, S. (2018). Detection o aruco markers. *OpenCV Documentation*.
- S. Gunawan, G. P. y Cahyadi, A. (2019). Smoothed a-star algorithm for nonholonomic mobile robot path planning. " *International Conference on Information and Communications Technology*.
- Takeyas, B. (2005). Algoritmo  $a^*$ . *Instituto tecnológico de Nuevo Laredo*.
- Tatsuya, M. y Yuki, N. (2022). World robot challenge 2020-partner robot: a data-driven approach for room tidying with mobile manipulator. *Advanced Robotics*.
- Xiaofei, C. y Gang, Z. (2018). Wsns node localization algorithm based on mobile robot assistance. *3rd IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*.
- Yuki, N. y Shuki, T. (2022). Automated hammering inspection system with multi-copter type mobile robot for concrete structures. *IEEE Robotics and Automation Letters*.