

Planificación inteligente de trayectorias para un robot móvil diferencial en entornos dinámicos

Axel Herroz Herrera*, Alejandro Rodríguez-Molina*, Geovanni Flores-Caballero†, Omar Serrano-Pérez‡, Mario Aldape-Pérez§, Miguel Ángel Paredes-Rueda*

* Tecnológico Nacional de México/IT de Tlalnepantla, Research and Postgraduate Division, Tlalnepantla de Baz 54070, México.

† Universidad Aeronáutica en Querétaro, Querétaro 76278, México.

‡ Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería Campus Guanajuato, Guanajuato 36275, México.

§ Instituto Politécnico Nacional, CIDETEC, Computational Intelligence Laboratory (CIL), Ciudad de México 07700, México.

Abstract—La planificación de trayectorias es una tarea esencial para cualquier robot móvil. Esta tarea consiste en llevar a un robot móvil desde una posición inicial a una final evadiendo obstáculos con el menor costo posible. Cuando los entornos de operación son estáticos, i.e., cuando todos los obstáculos en el escenario permanecen siempre en la misma configuración, la planificación de trayectorias puede realizarse fuera de línea, antes de que el robot comience a moverse, o en línea, asumiendo que los obstáculos no cambiarán. Sin embargo, cuando estos entornos son dinámicos, i.e., cuando los obstáculos están en movimiento, la tarea de planificación se complica considerablemente y es necesario replanificar las trayectorias en línea, mientras el robot se mueve. En este sentido, se han propuesto planificadores sofisticados que, por su elevado costo o su alta complejidad, pueden resultar poco factibles de implementar en la práctica. Este trabajo describe un planificador simple pero eficiente basado en un algoritmo clásico de tipo Bug. A través de cómputo inteligente, esta propuesta realiza la optimización en línea de los parámetros de Bug cuando es necesario, para que el robot logre evadir los obstáculos dinámicos de forma efectiva, con alto rendimiento y con un costo reducido. Los resultados de la propuesta se muestran en simulación para un robot móvil diferencial.

Index Terms—Planificación de trayectorias, optimización en línea, entornos dinámicos, robot móvil diferencial, Bug0.

I. INTRODUCCIÓN

Los robots móviles conforman un tipo de sistema mecánico que puede desplazarse por medios determinados (e.g., tierra, agua o aire) para realizar tareas diversas. Sin importar su clase, los robots móviles requieren de una adecuada planificación de trayectorias para lograr sus objetivos. La planificación de trayectorias consiste en llevar a un robot móvil desde una posición inicial a una final evadiendo obstáculos con el menor costo posible, e.g., en términos de distancia recorrida, energía consumida o rapidez.

Cuando los obstáculos en el entorno de operación del robot móvil son estáticos, i.e., cuando conservan su configuración al paso del tiempo, la planificación de trayectorias puede realizarse fuera de línea antes de que el robot comience a desplazarse, o en línea, asumiendo que los obstáculos no cambiarán. La planificación de trayectorias para robots móviles en escenarios estáticos se ha estudiado ampliamente en las últimas décadas. De estos estudios se han propuesto algoritmos de planificación altamente efectivos para una variedad amplia

de consideraciones prácticas. En [1] se realiza un estudio exhaustivo de estos algoritmos y se les clasifica en cuatro grandes grupos. Los algoritmos basados en cómputo reactivo, como Bug [2] o el método del Campo Potencial Artificial [3], realizan la planificación de trayectorias continuamente utilizando la información del entorno que el robot puede adquirir en cada instante de tiempo a través de sus sensores. Los algoritmos de cómputo suave se encargan de optimizar la trayectoria entre dos puntos a partir de un criterio (e.g., la minimización de la distancia total) a través de algoritmos de cómputo suave (e.g., redes neuronales, lógica difusa o metaheurísticas). Por su parte, los algoritmos de búsqueda en espacio de configuración (en inglés, C-Space), como A* [4] o los Árboles Aleatorios de Rápida Exploración [5], realizan la discretización del entorno de operación para limitar el número de alternativas en el ajuste de trayectorias. Por último, las trayectorias optimizadas se obtienen como la solución del problema de control óptimo en los algoritmos basados en control óptimo [6].

Si el entorno de operación del robot móvil es dinámico, i.e., si los obstáculos cambian su configuración con el tiempo, la planificación de trayectorias con los algoritmos anteriores puede resultar poco efectiva, ya que podría generar colisiones entre el robot y los obstáculos en movimiento que *a priori* se consideraron estáticos. Por esto, se han propuesto estrategias recientes para resolver el problema de planificación de trayectorias en entornos dinámicos en línea, i.e., replanificando continuamente la trayectoria en intervalos de tiempo constantes o tan pronto como se detecta un cambio en el entorno. Estos enfoques de solución incluyen la hibridación de técnicas de planificación estáticas [7] o la propuesta de nuevos operadores en dichas técnicas [8]. A pesar de que estas alternativas son efectivas, su costo y complejidad pueden ser elevados, por lo que su implementación práctica puede no ser factible.

Por lo anterior, este trabajo propone un planificador simple pero eficiente basado en un algoritmo clásico de tipo Bug. A través de cómputo inteligente, el nuevo planificador realiza la optimización en línea de los parámetros de Bug cuando es necesario, para que el robot logre evadir los obstáculos dinámicos de forma efectiva, con alto rendimiento y con un costo reducido. Los resultados de la propuesta se muestran en

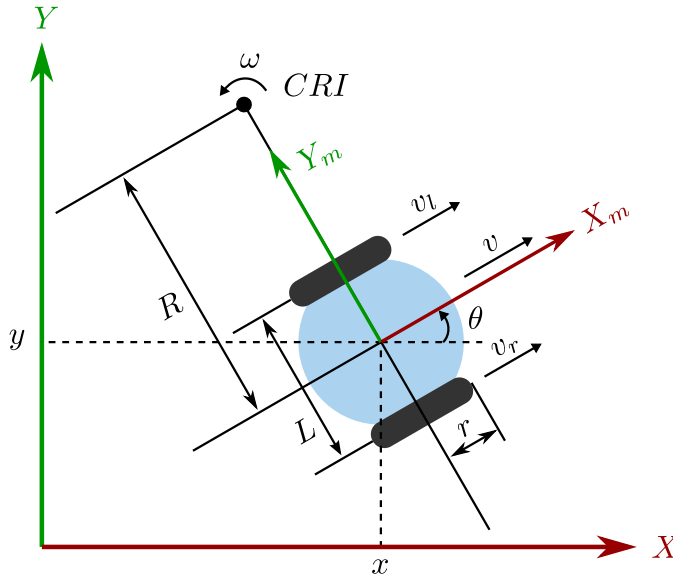


Fig. 1. Esquemático del robot móvil diferencial.

simulación para un robot móvil diferencial.

El resto del artículo se organiza de la siguiente manera. La Sección II describe al robot móvil diferencial y al algoritmo clásico Bug0 para la planificación en escenarios estáticos. La Sección III incluye la propuesta del nuevo método de planificación basado en Bug0 y cómputo inteligente para escenarios dinámicos. Los experimentos y resultados se detallan en la sección IV. Finalmente, la Sección V presenta las conclusiones.

II. PLANIFICACIÓN DE TRAYECTORIAS PARA EL ROBOT MÓVIL DIFERENCIAL

A. El robot móvil diferencial

El robot móvil diferencial consta de una plataforma con dos ruedas idénticas de radio r separadas por un eje de rotación de tamaño L , como se observa en el esquemático de la Fig. 1. La posición y orientación del robot puede describirse a través de las coordenadas x y y , y el ángulo θ medido con respecto al eje X . Estas variables se consideran como las coordenadas generalizadas del sistema. Por otro lado, el giro de cada rueda es controlado independientemente y sus velocidades angulares producen cambios en la posición y orientación del robot. Dependiendo de la diferencia en las velocidades angulares de las ruedas, el robot puede girar alrededor del Centro de Rotación Instantáneo (CRI) con radio R , velocidad tangencial v y velocidad angular ω .

Las relaciones entre las velocidades angulares de las ruedas y las velocidades ω y v del robot están dadas por:

$$\begin{aligned} \omega &= \frac{v_r - v_l}{L} \\ v &= \frac{v_r + v_l}{2} \end{aligned} \quad (1)$$

con $v_r = r\omega_r$ y $v_l = r\omega_l$ como las velocidades tangenciales de cada rueda y ω_r, ω_l sus correspondientes velocidades angulares.

Así, ω y v pueden considerarse como las entradas del sistema y su modelo cinemático se deriva de la siguiente forma a partir de la Fig. 1:

$$\begin{aligned} \dot{x} &= v \cos(\theta) \\ \dot{y} &= v \sin(\theta) \\ \dot{\theta} &= \omega \end{aligned} \quad (2)$$

El modelo cinemático en (2) permite simular el comportamiento del robot móvil diferencial, en términos de sus coordenadas generalizadas, para cualquier instante de tiempo cuando se le inyectan las entradas v y ω .

B. El algoritmo de planificación Bug0

Como se mencionó en la sección introductoria, los algoritmos Bug son métodos de planificación basados en cómputo reactivo, i.e., para cada instante de tiempo, utilizan la información del entorno que el robot tiene a su alcance para decidir su siguiente maniobra. Estos algoritmos se caracterizan por su alta eficacia para planificar trayectorias en escenarios estáticos, su bajo costo computacional (considerado despreciable) y su simple implementación práctica [9]. Por lo anterior, el planificador de trayectorias para entornos dinámicos que se propone en este trabajo está basado en una de las alternativas de los algoritmos Bug conocida como Bug0.

Algoritmo 1: Bug0.

Entradas: Distancia al obstáculo más cercano (d_{obs}), Orientación del obstáculo más cercano con respecto al robot (θ_{obs}), Posición final (x_f, y_f), Posición actual del robot (x, y) Orientación actual del robot (θ), Ganancias de los controladores de velocidad (g_1, g_2), Dirección de evasión (s).

Salidas: Velocidades tangencial y angular de entrada para el robot (v, ω).

```

1 if  $d_{obs} > \mu$  then
2    $\theta_{ref} \leftarrow \arctan 2(y_g - y, x_g - x)$ 
3    $e_\theta \leftarrow \theta_{ref} - \theta$ 
4    $d_g \leftarrow \sqrt{(x_g - x)^2 + (y_g - y)^2}$ 
5    $g \leftarrow \left[ \frac{d_g}{2}, g_2 \right]^T$ 
6 else
7    $\theta_{ref} \leftarrow \theta_{obs} + s \frac{\pi}{2}$ 
8    $e_\theta \leftarrow \theta_{ref} - \theta$ 
9    $g \leftarrow [g_1, g_2]^T$ 
10  $v \leftarrow g_1 |\cos(e_\theta)|$ 
11  $\omega \leftarrow g_2 \cdot e_\theta$ 
12 return  $v, \omega$ 
    
```

El Algoritmo 1 muestra el funcionamiento de Bug0 para el robot móvil diferencial. En intervalos fijos de tiempo Δt , Bug0 utiliza la información actual del robot móvil y del obstáculo más cercano para calcular las entradas de control del robot como sigue:

- 1) Si la distancia al obstáculo es mayor a un umbral determinado μ , se utiliza un control de velocidad de

ganancia variable que depende de la distancia hacia la posición final para llevar al robot en dirección hacia ésta en línea recta (líneas 2-5 y 10-12 en el Algoritmo 1).

- 2) En caso contrario, se utiliza un control de velocidad de ganancia fija para cambiar la dirección de movimiento del robot hacia la izquierda o derecha (respectivamente con $s = 1$ o $s = -1$, pero solo una elección) del obstáculo para evadirlo (líneas 7-9 y 10-12 en el Algoritmo 1).

III. PLANIFICACIÓN INTELIGENTE EN LÍNEA PARA EL ROBOT MÓVIL DIFERENCIAL

El planificador Bug0 en el Algoritmo 1 asume un entorno de operación estático para el robot móvil diferencial. En este sentido, Bug0 hará que el robot se mueva hacia la posición final con una velocidad que tendrá siempre la misma proporción a la distancia que los separa. De igual manera, al momento de evadir un obstáculo cercano, lo hará con velocidad proporcional a la orientación de éste y siempre en la misma dirección (izquierda o derecha, pero solo una elección).

Debido a lo anterior, si se consideran escenarios dinámicos, pueden existir casos en donde estas proporciones no sean adecuadas para evadir a tiempo a obstáculos en movimiento (el robot podría moverse más lento o más rápido de lo necesario para librar la colisión), o en donde el robot y el obstáculo se muevan hacia la misma dirección y terminen colisionando.

El algoritmo de planificación que se propone en este trabajo busca resolver los problemas anteriores a través del ajuste en línea de los parámetros de Bug0 cuando es necesario, i.e., cuando hay un obstáculo cerca del robot con base en $d_{obs} < \mu$. Para esto se utiliza la técnica de cómputo inteligente conocida como Evolución Diferencial.

A. El problema de optimización de Bug0

El ajuste de Bug0 debe establecerse como un problema de optimización formal de la siguiente manera:

$$\min J(p) \quad (3)$$

sujeto a:

$$g_i(p) \leq 0, i = 1, 2, \dots, n_g \quad (4)$$

$$h_j(p) = 0, j = 1, 2, \dots, n_h \quad (5)$$

En el problema anterior se buscan los valores de las variables de diseño en el vector p que minimizan a la función de costo J en (3) y que además satisfacen las restricciones indicadas en (4) y (5).

En Bug0, las variables que pueden limitar su efectividad para manejar entornos dinámicos son las ganancias proporcionales Kp_1, Kp_2 y la dirección de evasión s . Entonces, el vector de variables de diseño se escoge como:

$$p = [s, Kp_1, Kp_2]^T \quad (6)$$

Por otro lado, uno de los aspectos deseables en la tarea de planificación es el costo reducido de las trayectorias. Por esta razón, se establece la siguiente función objetivo:

$$J(p) = \sqrt{(x_g - \hat{x}(t + h \Delta t))^2 + (y_g - \hat{y}(t + h \Delta t))^2} \quad (7)$$

La función objetivo anterior considera la distancia entre la posición final x_g, y_g y la posición futura del robot móvil $\hat{x}(t + h \Delta t), \hat{y}(t + h \Delta t)$ estimada con el modelo cinemático en (2) para un horizonte de tiempo futuro $h \Delta t$ al usar Bug0 con los parámetros en p , con h como el número de pasos futuros.

Una trayectoria factible puede considerarse como aquella en donde el robot no colisiona con los obstáculos. Debido a esto, únicamente se consideran restricciones de igualdad (i.e., restricciones que deben cumplirse exactamente) en el problema de optimización de Bug0. Estas restricciones se muestran a continuación:

$$h_j(p) = \begin{cases} 1, & \text{if } d_k(t + l dt) < \frac{L}{2} + \frac{L_k}{2} \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

$$j = 1, 2, \dots, n_{obs}h,$$

$$k = 1, 2, \dots, n_{obs},$$

$$m = 1, 2, \dots, h$$

en donde $d_k(t + m \Delta t)$ es la distancia entre la posición del robot (estimada con el modelo cinemático en (2) al utilizar Bug0 con los parámetros en p) y la posición del k -ésimo obstáculo (predicha o calculada) para el instante de tiempo futuro $t + m \Delta t$, con L_k como el diámetro del k -ésimo obstáculo y n_{obs} como el número de obstáculos en el escenario. Para esto, se asume que la posición de cada obstáculo en movimiento puede calcularse o estimarse. Algunos trabajos que abordan este problema pueden consultarse en [10], [11].

B. El optimizador inteligente

El problema de optimización en (6)-(8) es complejo, ya que posee términos no lineales, discontinuos y no diferenciables, por lo que no es candidato para ser resuelto por algoritmos clásicos de optimización. Sin embargo, las técnicas de optimización de cómputo inteligente pueden ayudar a encontrar soluciones adecuadas a este problema. En particular, este trabajo adopta el algoritmo de cómputo evolutivo conocido como Evolución Diferencial (ED) [12] para resolver el problema de optimización de Bug0. ED ha mostrado ser eficaz en la solución de problemas complejos de ingeniería y su variante más simple es la *rand/1/bin*.

El Algoritmo 2 muestra el funcionamiento general de ED *rand/1/bin*. Al inicio, ED genera una población P_G con NP individuos (soluciones candidatas al problema de optimización) aleatorios (líneas 1 y 2). Durante un número determinado de generaciones G_{max} y para cada uno de los individuos base en la población (líneas 3 y 4), ED selecciona tres individuos de la población aleatoriamente (línea 5). Estos tres individuos se utilizan para formar un mutante con ayuda del operador de mutación *rand/1* (línea 6) y el factor de escala $F \in [0, 1]$. Utilizando la información del individuo base y del mutante, la operación de cruce *bin* (línea 7) genera un individuo descendiente, con la probabilidad de cruce $CR \in [0, 1]$.

El descendiente compite con el individuo base para determinar cuál prevalecerá en la población de la siguiente generación (línea 8). Al paso de las generaciones (línea 9), los individuos más aptos permanecerán en la población y, al final, el mejor de ellos se toma como la solución al problema de optimización (líneas 10 y 11).

Algoritmo 2: Evolución Diferencial

Entradas: Número de generaciones (G_{max}), Número de individuos (NP), Probabilidad de cruce (CR), Factor de escala (F).

Salidas: El mejor vector de variables de diseño (p_{best}).

```

1  $G \leftarrow 0$ 
2 Genera una población inicial  $P_G$  con  $NP$  individuos aleatorios.
3 while  $G < G_{max}$  do
4   foreach  $p_i \in P_G$  do
5     Selecciona tres individuos  $p_{r1}$ ,  $p_{r2}$ , y  $p_{r3}$  aleatoriamente de la población  $P_G$  tal que  $r1 \neq r2 \neq r3 \neq i$ .
6     Genera un individuo mutante  $v_i = p_{r1} + F(p_{r2} - p_{r3})$ .
7     Genera un individuo descendiente  $u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rnd(0,1) < CR \text{ or } j = j_{rnd} \\ p_{i,j}, & \text{otherwise} \end{cases}$ , con  $j_{rnd}$  como una variable de diseño aleatoria y  $rnd(0,1)$  un número aleatorio en  $[0, 1]$ .
8     Elige entre  $p_i$  y  $u_i$  al individuo más apto que prevalecerá en la población  $P_{G+1}$ .
9    $G \leftarrow G + 1$ 
10 Selecciona al mejor individuo  $p_{best}$  de la población  $P_G$ .
11 return  $p_{best}$ 
    
```

IV. EXPERIMENTOS Y RESULTADOS

A. Descripción de los experimentos

Para los experimentos en simulación se considera un robot móvil diferencial con $L = 0.15$ (m) y $r = 0.02$ (m). La posición inicial del robot se establece en $x_0 = 0$ (m) y $y_0 = 0$ (m), mientras que su orientación inicial en $\theta_0 = 0$ (rad). Asimismo, se busca que el robot móvil llegue a la posición final dada por $x_g = 5$ (m) y $y_g = 0$ (m). La simulación del robot móvil se realiza a partir del modelo cinemático en (2), utilizando el método de integración de Euler con la condición inicial $x^0 = 0 \in R^3$, con un intervalo de muestreo $\Delta t = 0.03$ (s) y un tiempo final de quince segundos.

En el espacio comprendido por la posición inicial y final del robot móvil se incluyen cinco obstáculos dinámicos que realizan las diferentes trayectorias descritas en la Tabla I. Cada obstáculo se asume del mismo tamaño que el robot móvil, i.e., $L_k = L$, $k = 1, \dots, 5$.

En cuanto al planificador propuesto, se elige $h = 10$ como el horizonte de predicción para la evaluación del problema de optimización en (6)-(8). El umbral de distancia al obstáculo más cercano en Bug0, que también se utiliza para determinar cuándo se realiza la optimización de sus parámetros con ED, se establece en $\mu = 3L$. Los parámetros de ED se eligen como $G_{max} = 100$, $NP = 20$, $CR = 0.5$ y $F = 0.5$, con el

TABLA I
TRAYECTORIAS DE LOS OBSTÁCULOS DINÁMICOS.

Obstáculo (k)	$x_k(t)$	$y_k(t)$
1	1.0	$0.1 \sin(\frac{1}{2}t)$
2	2.0	$0.3 \cos(t)$
3	$2.5 + 0.25 \cos(t)$	$0.25 \sin(t)$
4	3.0	$0.35 \sin(\frac{1}{2}t)$
5	4.0	$0.3 \cos(t)$

objetivo de obtener resultados en un tiempo asequible para la propuesta (menor a Δt).

El planificador de trayectorias propuesto se implementó en el lenguaje de programación C++ y las simulaciones se ejecutaron en una computadora con Intel(R) Core(TM) i3-10105F CPU @ 3.70GHz y 32GB de RAM.

B. Resultados y discusión

Debido a que el planificador propuesto tiene un comportamiento estocástico inducido por el uso de ED, se realizan 30 ejecuciones independientes del experimento descrito en la sección anterior para obtener información relevante de su funcionamiento.

La Tabla II muestra los resultados de las 30 ejecuciones independientes del experimento. Esta tabla incluye información sobre el número de ejecución, la distancia recorrida por el robot móvil, las veces que colisionó con los obstáculos en movimiento, el tiempo que le tomó llegar a la posición final, y el tiempo total de ejecución. Por su parte, la Tabla III muestra el resumen estadístico de los resultados de la Tabla II.

Con base en la información de las Tablas II y III, se puede observar que la propuesta es capaz de generar trayectorias libres de colisiones, una característica fundamental para la planificación, pues evita daños en el robot y costos de reparación adicionales. Por otro lado, se puede notar que la distancia recorrida presenta una desviación pequeña con respecto al promedio (alrededor de 4cm), lo que denota una alta repetibilidad de las trayectorias obtenidas con el planificador a pesar de ser una propuesta estocástica. Algo similar sucede con la duración del recorrido y el tiempo de ejecución, cuya desviación está alrededor de una décima de segundo. En particular, se debe notar que el tiempo de ejecución es menor a la mitad del tiempo que le demora al robot completar el trayecto. Lo anterior resalta la factibilidad para implementar la propuesta en la práctica.

La Fig 2 muestra una ejecución arbitraria del planificador de trayectorias propuesto. Cada imagen incluye la información de la posición inicial (circulo azul), la posición final (circulo azul), el robot móvil diferencial y su orientación (circulo negro con segmento de recta), el recorrido realizado por el robot (línea segmentada en negro), los obstáculos (círculos rojos) y sus recorridos (líneas punteadas en rojo), para cada segundo del recorrido. Esta figura muestra como el planificador propuesto ayuda al robot móvil a evadir a tiempo los obstáculos móviles, mientras que lo lleva a la posición final con una distancia de recorrido minimizada.

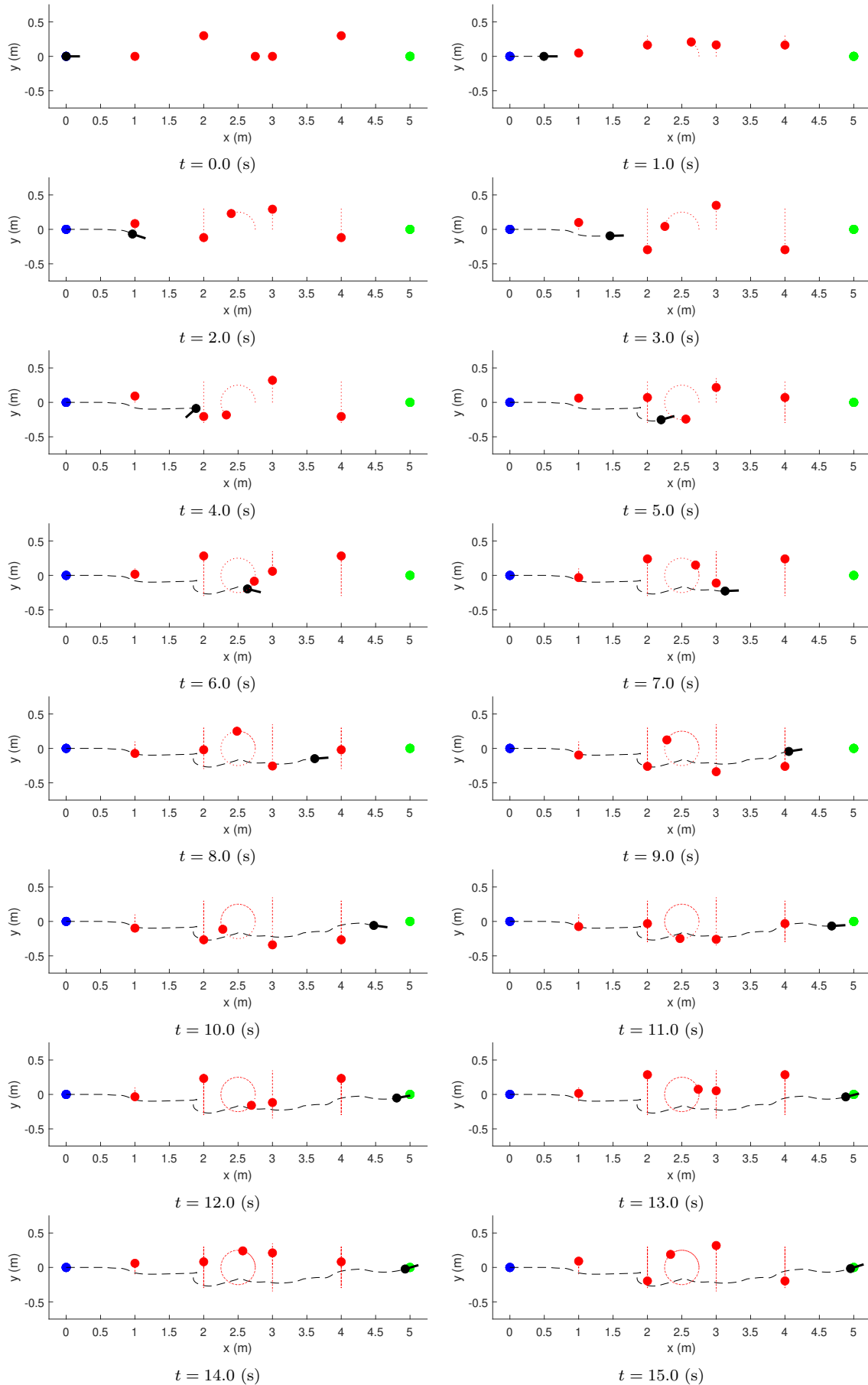


Fig. 2. Información de una ejecución arbitraria del planificador de trayectorias propuesto. Las imágenes muestran el escenario en cada segundo del recorrido.

TABLA II
RESULTADOS DE LAS 30 EJECUCIONES INDEPENDIENTES CON EL PLANIFICADOR PROPUESTO.

Ejecución	Distancia recorrida (m)	Colisiones con obstáculos	Duración del recorrido (s)	Tiempo de ejecución (s)
1	5.2244	0	14.5800	7.0210
2	5.2237	0	14.5800	6.7310
3	5.1907	0	14.2500	6.5490
4	5.2406	0	14.6100	6.7820
5	5.1720	0	14.3700	6.3780
6	5.2282	0	14.5800	6.8710
7	5.2316	0	14.5800	6.7360
8	5.2357	0	14.6100	6.7440
9	5.2484	0	14.6100	6.7380
10	5.2384	0	14.7600	6.6330
11	5.2339	0	14.6100	6.7310
12	5.2597	0	14.6400	6.7680
13	5.2444	0	14.6400	6.7670
14	5.2355	0	14.6100	6.7710
15	5.2201	0	14.5500	6.8060
16	5.2349	0	14.6400	6.8230
17	5.2549	0	14.6400	6.7650
18	5.2460	0	14.6100	6.7750
19	5.2411	0	14.6100	6.9430
20	5.1963	0	14.2800	7.0000
21	5.1644	0	14.3700	6.9500
22	5.2397	0	14.6100	6.7930
23	5.3846	0	14.8200	7.1100
24	5.2274	0	14.6100	6.7620
25	5.2232	0	14.6400	6.7600
26	5.2376	0	14.6100	6.7640
27	5.3453	0	14.7600	6.9140
28	5.2276	0	14.5500	6.7130
29	5.1803	0	14.3700	6.5330
30	5.2469	0	14.6100	6.9430

TABLA III
RESUMEN ESTADÍSTICO DE LOS RESULTADOS.

Ejecución	Distancia recorrida (m)	Colisiones con obstáculos	Duración del recorrido (s)	Tiempo de ejecución (s)
Mínimo	5.1644	0	14.2500	6.3780
Máximo	5.3846	0	14.8200	7.1100
Promedio	5.2359	0	14.5770	6.7858
Desviación estándar	0.0424	0	0.1290	0.1484

V. CONCLUSIONES

La planificación de trayectorias para robots móviles es un problema complejo, pero actualmente existen métodos que son capaces de resolverlo de forma efectiva y, en algunos casos, de manera eficiente cuando los entornos de operación son estáticos. Cuando los escenarios incluyen obstáculos en movimiento, el problema se complica y dichos métodos no suelen ser suficientemente eficaces. En estos casos, las alternativas de solución en la literatura especializada son sofisticadas, complejas y de alto costo, por lo que pueden ser poco factibles de implementar en la práctica.

El método de planificación que se presenta en este trabajo es una alternativa simple y eficiente. La propuesta aprovecha la simplicidad, bajo costo y alta efectividad del algoritmo Bug0, utilizado para la planificación en escenarios estáticos, dotándola de la habilidad para extender su funcionamiento al caso dinámico. Lo anterior, a través de la optimización en línea de sus parámetros con Evolución Diferencial, una técnica de cómputo evolutivo ampliamente utilizada para la solución de problemas complejos en ingeniería por su alta efectividad y cómoda implementación.

A través de pruebas en simulación con un robot móvil diferencial, la propuesta mostró ser efectiva para generar trayectorias libres de colisiones con obstáculos dinámicos y con longitud reducida. Asimismo, se observó que el planificador propuesto tiene un costo computacional asequible y las trayectorias que genera son repetibles a pesar de ser una alternativa estocástica.

REFERENCIAS

- [1] J. R. Sánchez-Ibáñez, C. J. Pérez-del Pulgar, and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, no. 23, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/23/7898>
- [2] R. A. Langer, L. S. Coelho, and G. H. C. Oliveira, "K-bug, a new bug approach for mobile robot's path planning," in *2007 IEEE International Conference on Control Applications*, 2007, pp. 403–408.
- [3] J.-F. Duhé, S. Victor, and P. Melchior, "Contributions on artificial potential field method for effective obstacle avoidance," *Fractional Calculus and Applied Analysis*, vol. 24, no. 2, pp. 421–446, Apr 2021. [Online]. Available: <https://doi.org/10.1515/fca-2021-0019>
- [4] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014, modelling of Mechanical and Mechatronic Systems. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187770581403149X>

- [5] C. Ronco and R. Bellomo, "Acute renal failure and multiple organ dysfunction in the icu: from renal replacement therapy (rrt) to multiple organ support therapy (most)," *The International journal of artificial organs*, vol. 25, no. 8, pp. 733–747, 2002.
- [6] D. Ma, S. Hao, W. Ma, H. Zheng, and X. Xu, "An optimal control-based path planning method for unmanned surface vehicles in complex environments," *Ocean Engineering*, vol. 245, p. 110532, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801822000129>
- [7] D. Lin, B. Shen, Y. Liu, F. E. Alsaadi, and A. Alsaedi, "Genetic algorithm-based compliant robot path planning: an improved bi-rrt-based initialization method," *Assembly Automation*, 2017.
- [8] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790612001255>
- [9] G. Klancar, A. Zdesar, S. Blazic, and I. Skrjanc, *Wheeled mobile robotics: from fundamentals towards autonomous systems*. Butterworth-Heinemann, 2017.
- [10] M. Gao, R. Kohlhaas, and J. M. Zöllner, "Contextual learning and sharing autonomy to assist mobile robot by trajectory prediction," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 274–275.
- [11] L. Xin, P. Hailong, and L. Jianqiang, "Trajectory prediction based on particle filter application in mobile robot system," in *2008 27th Chinese Control Conference*, 2008, pp. 389–393.
- [12] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.