

Path planning for mobile robots based on optimal interaction of electrostatic fields.

Daniel Fernando Zambrano-Gutierrez *,
Emmanuel Ovalle-Magallanes *, Jesus Joaquin Yanez-Borjas *,
Horacio Rostro-Gonzalez *, Juan Gabriel Avina-Cervantes *

* *Departamento de Ingeniería Electrónica, División de Ingenierías del Campus Irapuato-Salamanca, Universidad de Guanajuato, Salamanca 36885, Mexico. Email: {df.zambranogutierrez, e.ovallemagallanes, jesus.yanez, hrostrog,avina}@ugto.mx*

Abstract: This article presents a concise review of interaction laws between electrostatic charges applied to path planning for a mobile robot. The robot is represented as a punctual charge, moving freely in a predefined workspace or chart. The proposed methodology must help to estimate the free trajectory to reach a pre-established goal. The electrostatic interactions between the multiple elements of the environment (obstacles) and robot allow to dynamically determine a smooth trajectory, using minimum energy. Besides, the study includes the performance evaluation of different robust optimization techniques to solve the electrostatic charges interaction problem, estimating a fast and stable trajectory. The robot navigation is simulated, including the position or speed control parameters to improve displacement with a smoothed trajectory. In this way, the robot can evolve in complex environments as long as it disposes of the appropriate sensory information, even in dynamic environments.

Keywords: Path planning, Electrical potential, Optimization methods, Levenberg-Marquardt, Dog-Leg Algorithm

1. INTRODUCCIÓN

En la actualidad, los robots móviles se encuentran inmersos en diferentes actividades del mundo real, tales como operaciones militares, procesos agrícolas, transporte y manipulación de materiales peligrosos, entre otros [Ha et al., 2019b; Seyyedhasani et al., 2020; Evans et al., 1989]. Un robot móvil puede llegar a ser esencial en múltiples tareas, donde la intervención humana pueda estar comprometida, limitada o se requiera una alta precisión. En un robot móvil, los procesos autónomos son altamente priorizados. En este sentido, la navegación autónoma permite al robot desplazarse de una posición inicial a una posición final sin intervención humana. Durante este proceso, el robot debe planificar sus movimientos y evadir dinámicamente los obstáculos dentro de su recorrido. Cuando se dispone de un mapa de entorno, esta tarea es normalmente sencilla, pero cuando las condiciones de este cambian o cuando la cantidad de obstáculos es grande, la complejidad en la planificación de trayectorias aumenta considerablemente.

Para mejorar los procesos de navegación, se debe disponer de algoritmos robustos de planificación de trayectorias basados en métodos de optimización rápidos y estables. Al tener un conocimiento *a priori* del ambiente y los obstáculos presentes, es posible realizar la planificación de

una trayectoria global óptima [Alajlan et al., 2013], con una evasión de obstáculos precisa y alcanzando la meta deseada. En la literatura, una de las alternativas basada en las leyes de Maxwell permite realizar la planificación de trayectorias es la que explota las leyes de interacción de cargas y campos electrostáticos [Sang et al., 2021]. Esta metodología basada en principios físicos considera que la meta ejerce una fuerza de atracción en el robot y los obstáculos fuerzas de repulsión. Si bien esta idea, ya ha sido probada con anterioridad [Lee and Park, 2003], en este trabajo se propone una revisión de los fundamentos que sustentan su efectividad y la inclusión de métodos alternativos para la optimización de las trayectorias. En este tenor, se propone utilizar el método de Levenberg-Marquardt [Marquardt, 1963] y el método de Dog-Leg [Chauhan et al., 2018], los cuales, fueron aplicados de manera exitosa en el problema de planificación de trayectorias.

2. PLANIFICACIÓN DE TRAYECTORIAS

En general el problema de generación de trayectorias para un robot móvil establece que se puede determinar una trayectoria finita entre dos puntos, si el robot no colisiona con obstáculos y satisface algunos criterios de desempeño [Choset et al., 2005]. Se pueden clasificar dos

tipos de entornos para la planificación de movimientos de un robot móvil, estos son entornos dinámicos y estáticos. En el caso dinámico, no se tiene información completa del espacio a navegar y los obstáculos se consideran móviles. Para el caso estático, se conoce en su totalidad el ambiente donde se pretende navegar. Otros métodos empleados para la planificación de trayectorias son basados en enfoques estocásticos. Gong [2021] emplea un mapa probabilístico que es utilizado para calcular la trayectoria del robot móvil. Otros métodos usan técnicas topológicas relacionadas con el control automático y modelos electrostáticos [Ha et al., 2019a].

Sin embargo, los modelos fundamentados en métodos metaheurísticos han recibido reciente atención en la solución de innumerables problemas de ingeniería por su alta versatilidad, sin ser la planificación de trayectorias la excepción. De esta forma, los algoritmos genéticos han sido probados con éxito en métodos de planificación de trayectorias [Da Graça Marcos et al., 2009]. El enjambre de partículas (PSO, Particle Swarm Optimization) ha permitido igualmente realizar una planificación de trayectoria global exitosamente [Das and Jena, 2020]. En especial, la navegación basada en el método de potenciales eléctricos sigue siendo utilizado por su robustez, incluso en ambientes multi-agente [Gayle et al., 2009]. En el presente proyecto, se estudian las leyes físicas del electromagnetismo que rigen la interacción entre cargas con el objeto de modelar un proceso de navegación y planificación de trayectorias inspirados en potenciales electrostáticos y métodos de optimización robustos. En la siguiente sección, se presenta la descripción general del sustento teórico para solución de la tarea de navegación.

3. GENERACIÓN DE TRAYECTORIAS MEDIANTE POTENCIAL ELÉCTRICO

El método de campos potenciales es una de las técnicas utilizadas para la generación de trayectorias para robots móviles, la cual, por ser eficiente, robusta y de implementación relativamente simple, permite realizar controles de locomoción por posición o desplazamiento con base en la formulación matemática completamente determinista. Khatib [1985] propone que los obstáculos y el robot tengan una carga eléctrica del mismo signo. Esto con la finalidad de generar una fuerza de repulsión, mientras que la meta tiene asociada una carga eléctrica de signo opuesto, para atraer al robot al punto destino. La tarea asignada al robot consistirá en llegar a su objetivo a partir de la planificación de trayectoria generada dinámicamente, apoyándose en el campo eléctrico conservativo sobre el cual se desplaza el robot. El proceso se inicia definiendo la posición del robot por el vector, $\mathbf{X} = [x, y]^T \in \mathbb{R}^2$, y una función potencial de valor real, continua y, por tanto, diferenciable,

$$U(\mathbf{X}) : \mathbb{R}^2 \rightarrow \mathbb{R}. \quad (1)$$

El valor de una función potencial puede verse como energía almacenada, y por lo tanto, el gradiente del potencial nos genera el vector campo eléctrico, como se

establece en las leyes de Maxwell. En seguida, se observa que en presencia de una carga externa q_o se puede generar una fuerza electrostática por efecto del campo eléctrico del entorno. Lo cual permite establecer que el campo eléctrico se obtiene como $\mathbf{E} = -\nabla U$, y la fuerza a su vez se calcula por $\mathbf{F} = q_o \cdot \mathbf{E} = -q_o \cdot \nabla U$. Note que el operador gradiente está definido por

$$\nabla U(\mathbf{X}) = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]^T U(\mathbf{X}). \quad (2)$$

Con el objeto de que el robot sea atraído hacia el objetivo y evite exitosamente los obstáculos del entorno circundante, se debe estimar un campo eléctrico resultante total E_T que incluya la acción de los campos de repulsión con los obstáculos y de atracción hacia el objetivo. La fuerza total que experimenta la carga externa (representada por el robot), inducirá por medio de la segunda Ley de Newton una aceleración, lo cual propiciará el desplazamiento por medio de las ecuaciones de cinemática tradicional, $\mathbf{x}(t) = \frac{1}{2}at^2 + \mathbf{v}_o t + \mathbf{x}_o$. El campo eléctrico resultante viene dado por

$$\mathbf{E}_T = \mathbf{E}_a + \mathbf{E}_r, \quad (3)$$

donde, \mathbf{E}_a y \mathbf{E}_r son los campos de atracción y repulsión, respectivamente. Dado que el robot está representado por una carga q_o^+ del mismo signo que los obstáculos y signo opuesto a la posición destino, se obtiene la fuerza de atracción hacia el objetivo y repulsión hacia los obstáculos, como se muestra en la Fig. 1.

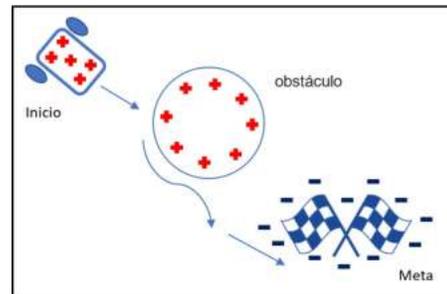


Fig. 1. La carga negativa atrae al robot y la carga positiva lo repele, lo que da como resultado una ruta óptima libre de obstáculos.

3.1 Campo de atracción electrostático

Se deben cumplir ciertos criterios a la hora de seleccionar la función del potencial $U_a(\mathbf{X})$. Dicho potencial debe ser una función convexa, diferenciable y real. Una opción es utilizar una función cuadrática,

$$U_a(\mathbf{X}) = \frac{1}{2}(\mathbf{X} - \mathbf{X}_g)^T \mathbf{Q}(\mathbf{X} - \mathbf{X}_g), \quad (4)$$

donde, \mathbf{Q} es una matriz función de cargas electrostáticas, la cual matemáticamente debe ser simétrica y definida positiva, $\mathbf{X}_g = [x_g, y_g]^T$ es la coordenada de la posición meta. La Fig. 2 muestra un potencial atractivo ilustrativo. Por lo tanto, el campo de atracción se define como

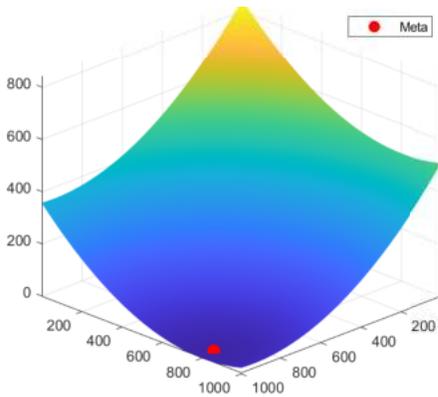


Fig. 2. Campo potencial de atracción $U_a(\mathbf{X})$.

$$\mathbf{E}_a = -\nabla U_a = -\nabla \left\{ \frac{1}{2} (\mathbf{X} - \mathbf{X}_g)^T \mathbf{Q} (\mathbf{X} - \mathbf{X}_g) \right\}, \quad (5)$$

$$\mathbf{E}_a = -\mathbf{Q} (\mathbf{X} - \mathbf{X}_g). \quad (6)$$

La fuerza de atracción entre el robot móvil y el objetivo viene dada por

$$\mathbf{F}_a(\mathbf{X}) = -q_o^+ \mathbf{E}_a. \quad (7)$$

3.2 Campo repulsivo electrostático.

La definición de potencial repulsivo consiste en crear una barrera de potencial alrededor de cada obstáculo que no pueda ser atravesada por el robot. Esto implica que el potencial repulsivo debe ser una función no negativa, continua y diferenciable. Además, es deseable que el potencial repulsivo no afecte al movimiento del robot cuando esté lo suficientemente lejos de los obstáculos. Dado que físicamente la proximidad de dos cargas llevaría a generar una fuerza electrostática muy grande, se considera en el análisis de interacción la aproximación hasta un cierto punto de seguridad entre las cargas, representadas por el robot y los obstáculos, sin provocar inconsistencias físicas por singularidades en las ecuaciones. Se pueden emplear distintas formas para el diseño de este potencial repulsivo. Usaremos una función potencial continua y suave para definir un conjunto de obstáculos. La función potencial eléctrico de referencia para n obstáculos se calcula de la siguiente manera,

$$U_r(\mathbf{X}) = \eta \sum_{i=1}^n \frac{Q_r}{R(\mathbf{X})}, \quad (8)$$

donde, $R(\mathbf{X}) = \|\mathbf{X} - \mathbf{X}_r\|_2$ es la distancia Euclidiana o radial entre dos cargas y Q_r son las cargas puntuales asociadas a cada obstáculo. El gradiente de este potencial de repulsión, $\mathbf{E}_r(\mathbf{X}) = -\nabla U_r(\mathbf{X})$, es quien en consecuencia genera la fuerza resultante de repulsión para n obstáculos que afectan a la carga prueba (robot). Esta resultante de repulsión es definida por

$$\mathbf{F}_r(\mathbf{X}) = K \sum_{i=1}^n \frac{q_o^- Q_r}{R^2} \left(\frac{x - x_{r_i}}{R} \hat{a}_x + \frac{y - y_{r_i}}{R} \hat{a}_y \right), \quad (9)$$

donde \hat{a}_x y \hat{a}_y son vectores unitarios en los ejes x y y , respectivamente. K es la constante de Coulomb y se define

como $K = \frac{1}{4\pi\epsilon_o}$, siendo $\epsilon_o = 8.85418 \times 10^{-12}$ [F/m] la permitividad en el vacío. Un campo de repulsión puede representarse tal como se muestra en la Fig. 3, donde cada obstáculo genera sus propias fuerzas de repulsión.

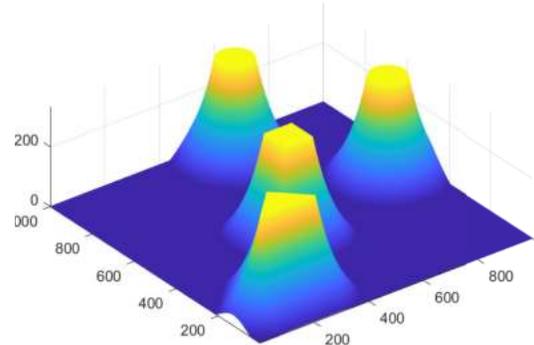


Fig. 3. Campo de Potencial Repulsivo $U_r(\mathbf{X})$.

3.3 Fuerza resultante y función de costo

El robot, al evolucionar en su entorno, estará afectado por la fuerza repulsiva de los obstáculos $\mathbf{F}_r(\mathbf{x})$ y la fuerza de atracción $\mathbf{F}_a(\mathbf{X})$ hacia el objetivo. De esta manera, la fuerza resultante se calcula por medio de

$$\mathbf{F}_T(\mathbf{X}) = -q_o^+ \nabla U_a(\mathbf{X}) + \eta \sum_{i=1}^n \left(\frac{x - x_{r_i}}{R^{3/2}} \hat{a}_x + \frac{y - y_{r_i}}{R^{3/2}} \hat{a}_y \right), \quad (10)$$

donde $\eta = \frac{q_o^- Q_r}{4\pi\epsilon_o}$ es una constante de regularización que permite controlar la proximidad del robot hacia los obstáculos.

La función resultante es la función de costo que se desea maximizar (o minimizar) según los signos asignados a la Ec. 10. Para valores muy cercanos de cargas, la fuerza crecería enormemente, por lo cual, se puede agregar una función Heaviside $u_h(\mathbf{X})$ en el término de $\mathbf{F}_r(\mathbf{X})$ para saturar la respuesta a distancias cortas. El campo eléctrico global para este estudio se muestra en la Fig. 4.

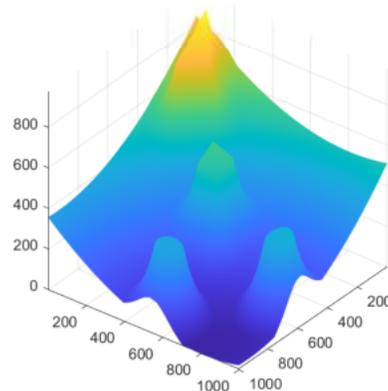


Fig. 4. Campo de Potencial Electrostático Total.

3.4 Método de Levenberg-Marquardt

El algoritmo de Levenberg-Marquardt genera una solución al problema de minimización de una función generalmente de naturaleza no lineal [Marquardt, 1963]. El citado algoritmo es una técnica iterativa que encuentra el mínimo de una función, con ciertos parámetros de inicialización. En este algoritmo no se necesita calcular una matriz Hessiana exacta, sino una aproximación la cual se define como,

$$\nabla^2 U(\mathbf{X}) \approx J(\mathbf{X})^\top J(\mathbf{X}), \quad (11)$$

donde, $J(\mathbf{X})$ es el Jacobiano del potencial. Considerando la siguiente aproximación,

$$H(\mathbf{X}) = \nabla^2 U(\mathbf{X}) \approx J(\mathbf{X})^\top J(\mathbf{X}) + \lambda \mathbf{I}, \quad (12)$$

donde λ es un parámetro que se escoge dinámicamente para que $H(\mathbf{X})$ sea definida positiva y \mathbf{I} es una matriz identidad, se obtiene finalmente el algoritmo de Levenberg-Marquardt (LM) (13),

$$\mathbf{X}_{k+1} = \mathbf{X}_k - (J(\mathbf{X}_k)^\top J(\mathbf{X}_k) + \lambda \mathbf{I})^{-1} J(\mathbf{X}_k)^\top. \quad (13)$$

Finalmente, el método de Levenberg-Marquardt en su forma general está descrito por el Algoritmo 1.

Algorithm 1 Método de Levenberg Marquardt.

Input : $U(\mathbf{X})$, \mathbf{X}_g , $iter$

Output: Coordenadas de la meta $[x^*, y^*]$

$\mathbf{X}_k \leftarrow [x, y]$ \triangleright Posición inicial del robot

$\lambda \leftarrow 0.001$ \triangleright Constante de amortiguamiento

while $Error > tolerancia$ **do**

$J(\mathbf{X}_{k+1}) = \nabla U(\mathbf{X}_k)$

$H = J(\mathbf{X}_k)^T J(\mathbf{X}_k) + \lambda \mathbf{I}$

$\alpha = H^{-1} J(\mathbf{X}_k)^T$

$\mathbf{X}_{k+1} = \mathbf{X}_k - \alpha$

$Error = \|\mathbf{X}_{k+1} - \mathbf{X}_g\|_2^2$

$iter \leftarrow iter + 1$

$\mathbf{X}_k \leftarrow \mathbf{X}_{k+1}$

end

3.5 Método de Dog-Leg

El método de Dog-Leg es un algoritmo iterativo basado en regiones de confianza que busca optimizar una función e incluyendo mejoras a los métodos tradicionales como el gradiente descendente y Gauss-Newton. En cada iteración se genera una nueva región de confianza. Dependiendo de los resultados obtenidos en esta región se toma las decisiones de aceptar o no el nuevo punto obtenido o de aumentar, reducir o mantener el radio de la región de confianza [Chauhan et al., 2018]. En su forma general el paso de actualización del algoritmo está descrito por,

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{P}_k, \quad (14)$$

donde \mathbf{P}_K es el incremento o tamaño de paso de la k -ésima iteración. Para conseguir \mathbf{P}_K que aproxime a la solución

óptima, se trabaja con un subproblema más simple que consiste en encontrar un \mathbf{P}_K^*

$$\mathbf{P}_K^* = \arg \min [m_k(\mathbf{P}), \mathbf{P} \in Q_k], \quad (15)$$

donde $m_k(\mathbf{P})$ es la función de costo y está definida como

$$\min_{\mathbf{P} \in \mathbb{R}^n} m_k(\mathbf{P}) = f_k + \nabla f_k^T \mathbf{P} + \frac{1}{2} \mathbf{P}^T b_k \mathbf{P} \quad (16)$$

f_k es el valor de la función de costo en la iteración k y $b_k \approx \nabla^2 f_k$, Q_k es la región de confianza y está definida como, $Q_k = \{\mathbf{P} \mid \|\mathbf{P}\| < \Delta_k\}$, donde Δ_k es el radio de la región de confianza en k -ésima iteración. Para ajustar el parámetro Δ_{k+1} en cada iteración, es necesario el cálculo de los vectores P_U y P_B tal que

$$\Delta_{k+1} = \|P_B + P_U\|_2, \quad (17)$$

$$P_U = -\frac{J(\mathbf{X}_k)^T J(\mathbf{X}_k)}{J(\mathbf{X}_k)^T H(\mathbf{X}_k) J(\mathbf{X}_k)}, \quad (18)$$

$$P_B = -H(\mathbf{X}_k)^{-1} J(\mathbf{X}_k), \quad (19)$$

donde $H(\mathbf{X})$ es la matriz Hessiana y $J(\mathbf{X})$ es el gradiente del potencial. Formalmente, el método de Dog-Leg se describe tal como se muestra en el Algoritmo 2.

Algorithm 2 Método de Optimización Dog-Leg.

Input : $U(\mathbf{X})$, \mathbf{X}_g , $\Delta > 0$, $\Delta_0 \in (0, \Delta)$, η , $iter$

Output: Coordenadas de la meta $[x^*, y^*]$

$\mathbf{X}_k \leftarrow [x, y]$ \triangleright Posición inicial del robot

Δ_0 \triangleright Radio inicial de la región de confianza

while $Error > tolerancia$ **do**

$\mathbf{P}_K = \arg \min [m_k(\mathbf{P}), \mathbf{P} \in Q_k]$

$\Delta_{k+1} = \|P_B(X_k) + P_U(X_k)\|$

if $\mathbf{P}_K < 1/4$ **then**

$\Delta_{k+1} = \Delta_k/4$

else

if $\mathbf{P}_K > 3/4$ **and** $|\mathbf{P}_K| = \Delta_k$ **then**

$\Delta_{k+1} = \min(2\Delta_k, \Delta)$

else

$\Delta_k + 1 = \Delta_k$

end

end

if $\mathbf{P}_K > \eta$ **then**

$\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{P}_k$

else

$\mathbf{X}_{k+1} = \mathbf{X}_k$

end

$Error = \|\mathbf{X}_{k+1} - \mathbf{X}_g\|_2^2$

$iter \leftarrow iter + 1$

end

4. RESULTADOS NUMÉRICOS

Las simulaciones numéricas se realizaron con Matlab-2020 en una PC con un Procesador Intel(R) Core(TM) i7 8700 en Windows 10. Con el fin de observar el comportamiento de las trayectorias empleando el método del Levenberg-Marquardt y el método de Dog-Leg, se utilizan los siguientes parámetros de optimización resumidos en

la Tabla 1. Donde V_0 y τ son coeficientes de velocidad inicial y tiempo en un paso del robot, quien inicia su trayectoria en $\mathbf{X}_0 \in \mathbb{R}^2$. Estos coeficientes permiten dar un paso mayor al generado por el gradiente manteniendo su dirección. La Tabla 2 muestra los resultados obtenidos

Tabla 1. Parámetros usados para el ajuste de los algoritmos de optimización

Levenberg Marquardt			Dog-Leg			
λ_0	V_0	τ	η	Δ_0	V_0	τ
0.001	2	1	0.2	1	2	1

a partir de diferentes posiciones iniciales del robot dado los obstáculos y la meta como se muestra en la Fig. 5. De manera práctica y para asignar unidades métricas,

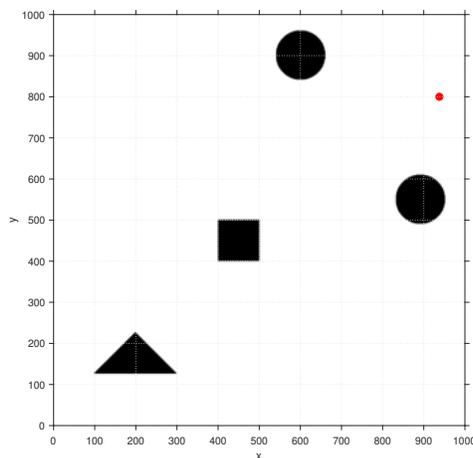


Fig. 5. Mapa generado con cuatro obstáculos fijos. Con un punto rojo se muestra la meta la cual el robot debe alcanzar.

se define un entorno de $1000 \times 1000 \text{ cm}^2$, con la meta en $\mathbf{X}_g = [940, 800]$. Es importante señalar que, por construcción del método basado en potenciales eléctricos, físicamente, la proximidad de dos cargas de signo opuesto (el robot y la meta) podría generar una singularidad por lo cual se tiende a una región equipotencial muy cercana. De esta manera, es posible definir una función de error que nos determine qué tan alejado queda el robot de la meta. La función de error se define como la distancia euclidiana entre ambas posiciones. De acuerdo con los

Tabla 2. Resultados numéricos para llegar al vector objetivo dado por $\mathbf{X}_g = [940, 800]$.

Levenberg Marquardt			Dog-Leg		
\mathbf{X}_0	Iter	Error [cm]	\mathbf{X}_0	Iter	Error [cm]
[36, 220]	166	3.6	[36, 220]	143	2.6
[428, 27]	150	3.2	[428, 27]	116	2.2
[150, 850]	100	3.4	[150, 850]	63	1.2

resultados de la Tabla 2, el método de Dog-Leg requiere un menor número de iteraciones y alcanza un menor error

en comparación con el método de Levenberg-Marquardt. En la Fig. 6 se observa la trayectoria obtenida con el algoritmo Levenberg-Marquardt. El robot logra llegar

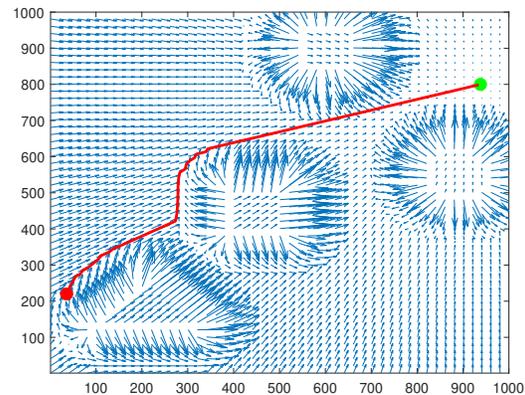


Fig. 6. Ruta generada mediante Potenciales Eléctrico y Levenberg-Marquardt.

de $\mathbf{X}_0 = [36, 220]$ a $\mathbf{X}_g = [938, 800]$ en 166 iteraciones, con un error de 3.6 cm. Por su parte, en la Fig. 7 se observa la trayectoria obtenida con el algoritmo Dog-Leg para la misma posición inicial del robot, y la misma meta. En cada iteración se dibuja una región de confianza y

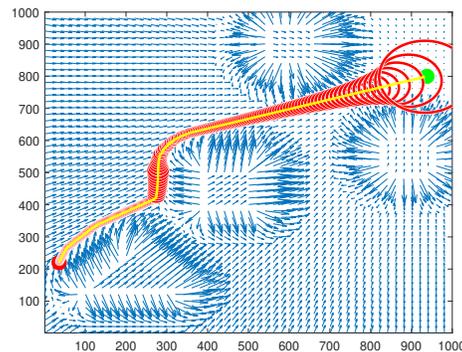


Fig. 7. Ruta generada mediante Potenciales Eléctrico y Dog-Leg. Las regiones de confianza son resaltadas en color rojo.

cumpliendo el objetivo en 143 iteraciones con un error de 2.3 cm.

5. CONCLUSIÓN

En este trabajo, se presenta un análisis detallado de los fundamentos matemáticos de las leyes de interacción de cargas electrostáticas para aplicaciones en robótica móvil, en específico en la planificación de trayectorias y la navegación reactiva. Esta metodología plantea la solución de la planificación de trayectorias de un robot que evoluciona en un ambiente conocido representado por un mapa de potenciales. Ciertas depresiones pueden ocurrir por efecto de la interacción global de fuerzas y el robot puede ser atraído hacia un mínimo local, lo cual puede ser solventado variando dinámicamente los parámetros del optimizador. Para encontrar la mejor solución, se compararon

dos técnicas robustas de optimización numérica, las cuales son la técnica de Levenberg-Marquardt y el método de Dog-Leg para la minimización de la función objetivo que permite una evolución suave hacia la meta. Los resultados numéricos demuestran que la planificación de trayectorias mediante potenciales eléctricos y técnicas de optimización numérica obtienen rutas continuas, y permiten la evasión dinámica de obstáculos. Además, se pueden aplicar un sin número de obstáculos como de potenciales se disponga, y el método aun en casos extremos se desempeña adecuadamente. Así mismo, mediante la formulación planteada es posible acelerar la convergencia mediante los coeficientes introducidos de velocidad inicial y tiempos en el paso del robot. Aunque estos algoritmos presentan una mayor exploración y explotación de la función a minimizar, la solución podestás estas comprometida al caer en mínimos locales. Sin embargo, se pueden desarrollar estrategias que detecten pérdida de convergencia o inestabilidad, para producir perturbación en los parámetros de las meta-heurísticas y poder garantizar la obtención de la solución.

AGRADECIMIENTOS

Este estudio fue apoyado por el Departamento de Ingeniería Electrónica de la División de Ingenierías (DICIS), de la Universidad de Guanajuato, NUA 147445, 147347, 145276, y el Consejo Nacional de Ciencia y Tecnología CONACYT (Mexico), con los números de beca 1046000/766748, 626154/755609, 688045/481368.

REFERENCIAS

- Alajlan, M., Koubaa, A., Chaari, I., Bennaceur, H., and Ammar, A. (2013). Global path planning for mobile robots in large-scale grid environments using genetic algorithms. In *2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR)*, 1–8. IEEE, Sousse, Tunisia. doi: 10.1109/ICBR.2013.6729271.
- Chauhan, V.K., Dahiya, K., and Sharma, A. (2018). Trust Region Levenberg-Marquardt Method for Linear SVM. In *2017 9th International Conference on Advances in Pattern Recognition, ICAPR 2017*, 380–385. IEEE, India. doi:10.1109/ICAPR.2017.8593090.
- Choset, H.M., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S., and Arkin, R.C. (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.
- Da Graça Marcos, M., Tenreiro Machado, J.A., and Azevedo-Perdicoulis, T.P. (2009). Trajectory planning of redundant manipulators using genetic algorithms. *Communications in Nonlinear Science and Numerical Simulation*, 14(7), 2858–2869. doi: 10.1016/j.cnsns.2008.10.014.
- Das, P.K. and Jena, P.K. (2020). Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Applied Soft Computing*, 92, 106312. doi: 10.1016/j.asoc.2020.106312.
- Evans, J., Krishnamurthy, B., Pong, W., Croston, R., Weiman, C., and Engelberger, G. (1989). HelpMate™: A robotic materials transport system. *Robotics and Autonomous Systems*, 5(3), 251–256. doi:10.1016/0921-8890(89)90049-3.
- Gayle, R., Moss, W., Lin, M.C., and Manocha, D. (2009). Multi-robot coordination using generalized social potential fields. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, 106–113. IEEE. doi: 10.1109/ROBOT.2009.5152765.
- Gong, W. (2021). Probabilistic model based path planning. *Physica A: Statistical Mechanics and its Applications*, 568, 125718. doi:10.1016/j.physa.2020.125718.
- Ha, J.S., Park, S.S., and Choi, H.L. (2019a). Topology-guided path integral approach for stochastic optimal control in cluttered environment. *Robotics and Autonomous Systems*, 113, 81–93. doi: 10.1016/j.robot.2019.01.001.
- Ha, Q.P., Yen, L., and Balaguer, C. (2019b). Robotic autonomous systems for earthmoving in military applications. *Automation in Construction*, 107, 102934. doi:10.1016/j.autcon.2019.102934.
- Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, 500–505. China. doi: 10.1109/ROBOT.1985.1087247.
- Lee, M.C. and Park, M.G. (2003). Artificial potential field based path planning for mobile robots using a virtual obstacle concept. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 2, 735–740. Delft, Delft, The Netherlands. doi:10.1109/AIM.2003.1225434.
- Marquardt, D.W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431–441. doi:10.1137/0111030. Publisher: Society for Industrial and Applied Mathematics.
- Sang, H., You, Y., Sun, X., Zhou, Y., and Liu, F. (2021). The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations. *Ocean Engineering*, 223, 108709. doi:10.1016/j.oceaneng.2021.108709.
- Seyyedhasani, H., Peng, C., Jang, W.J., and Vougioukas, S.G. (2020). Collaboration of human pickers and crop-transporting robots during harvesting – Part I: Model and simulator development. *Computers and Electronics in Agriculture*, 172, 105324. doi: 10.1016/j.compag.2020.105324.