ISSN 2007-9737

Finding a Collision-free Trajectory Using Electric Field Analysis and L1-norm

Victor Manuel Jimenez-Fernandez[1,*], Hector Vazquez-Leal[1], Uriel Antonio Filobello-Nino[1], Mario Alberto Sandoval-Hernandez[2]

[1] Universidad Veracruzana,
Facultad de Instrumentación Electrónica, Veracruz,
Mexico

[2] Instituto Tecnológico Superior de Poza Rica,
Ingeniería Electrónica, Poza Rica,
Mexico

{vicjimenez, hvazquez, ufilobello}@uv.mx,
mario.sandoval@itspozarica.edu.mx

Abstract. This paper addresses the problem of finding a collision-free trajectory between two points with known positions using the electric field concept. Analogous to an electric dipole, two strong electric point charges with opposite signs (positive and negative) are positioned in the navigation space at the starting and target points, respectively. Additionally, static obstacles are represented as slightly negative point charges. In this scenario, a hypothetical positive charge is considered a mobile particle that is released from the starting position with a specific angle in the vector field map sketched by the field lines created by the electric dipole. The trajectory produced by this hypothetical charge constitutes a path for avoiding obstacles from the origin to the destination points. To ensure that the computed trajectory remains inside a specified area, the navigation space is bounded by a rectangular border generated by a vector field function. Furthermore, to reduce the computational cost of the field map calculation (running time and memory consumption), the L1-norm is employed to describe distances instead of the traditional L2-norm (Euclidean norm). Numerical simulations demonstrate the effectiveness of this approach, as well as the reduction in computational cost by using the L1-norm.

Keywords. Collision-free, trajectory, electric-field, L1-norm.

1 Introduction

One of the most intuitive methods for determining obstacle avoidance trajectories for autonomous mobile robots is the so-called artificial potential-based method, whose principle of operation is based on the use of vector field functions [13, 8]. Its primary advantage lies in its simplicity: it offers a straightforward approach that is easy to understand and implement. In general terms, this method works by defining attractive forces toward a target point and repulsive forces away from obstacles, allowing robots to effectively navigate through complex environments while avoiding collisions [7, 14, 11].

However, this simplicity gives rise to certain drawbacks and challenges, such as restrictions on path optimization and an intensive requirement of computational resources [12, 15]. In this paper, we address these shortcomings with the aim of providing alternative numerical strategies to achieve better performance of this method. Specifically, our interest focuses in two particular aspects: (1) reducing the demand for computing resources, and (2) providing the capability to define an enclosed navigation area.

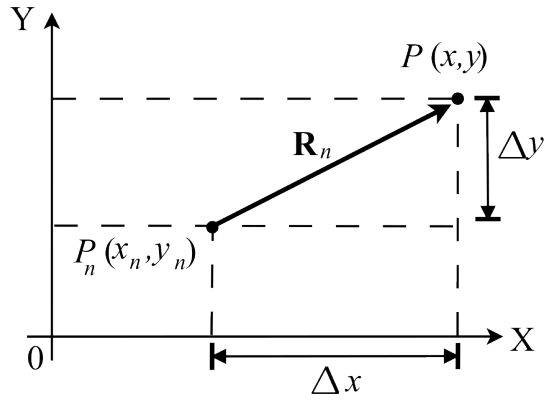

Fig. 1. Illustration of a Cartesian coordinate plane. Two points, $P_n(x_n, y_n)$ and $P(x, y)$, are marked. The directional vector between these points is $\mathbf{R}_n$, and the x and y increments are denoted as Δx and Δy, respectively

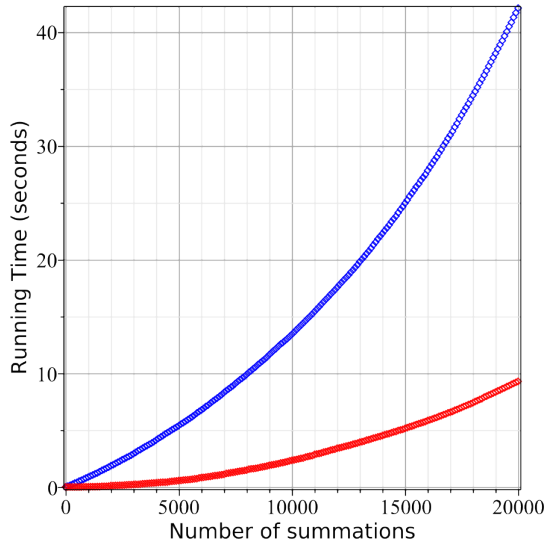

Fig. 2. Numerical simulation of vector computation. From 1 to 20,000 consecutive coordinate points using the L1-norm ($\mathbf{a}_{R_{n1}}$) and the Euclidean norm ($\mathbf{a}_{R_{n2}}$). Comparison of running times (in seconds). The results obtained using the L1 norm are depicted in red, while the results obtained using the Euclidean norm are presented in blue

In this sense, it is important to note that although many of the proposals reported for the calculation of collision-free trajectories use mathematical vector functions without any physical meaning, we have directly employed the concept of the electric field, which incorporates clarity and simplicity into the analysis [10, 9], but, as mentioned earlier, it also exhibits shortcomings in computational performance and navigation space. For example, the formulation for the electric field requires distance calculations expressed in the form of the Euclidean norm (L2-norm), which implies massive numerical operations of addition, exponentiation, and square root.

This leads to a high demand for computational resources that impacts the memory needed to store the resulting vector field function and the running time required for its evaluation.

Moreover, although it is possible to ensure the tracing of a path of field lines that emerge from the position of a source charge (positive) and converge to the position of a target charge (negative), it is not possible to guarantee that such a trajectory will remain within a defined navigation area. To overcome these shortcomings, we propose replacing the use of the Euclidean norm with the L1 norm [6, 4] and including vector field functions that act as an artificial boundary to enclose a defined spatial region.

2 Electric Field Computation

Let's consider the two-dimensional XY-coordinate plane of Fig.1 where $P_n\,(x_n, y_n)$ represents the position of any point charge, and $P(x, y)$ corresponds to any arbitrary coordinate point. We also consider the existence of an electric field, denoted as $\mathbf{E}_{P_n}$, which is created by a positive point charge Q located at $P_n\,(x_n, y_n)$.

Now, let's suppose that we are interested in determining the electric field at the point $P(x, y)$, which is represented as $\mathbf{E}_P$. Based on electrostatic principles, the magnitude of vector $\mathbf{E}_P$ can be expressed as:

$$|\mathbf{E}_P| = K\,\frac{Q}{r^2}, \tag{1}$$

where K is the Coulomb's constant ($K = 8.99 \times 10^9\ \mathrm{N\,m^2/C^2}$), and traditionally, r is the distance (expressed as Euclidean norm) between the point charge and the arbitrary point where the electric field is evaluated (in accordance with our case of study, P).

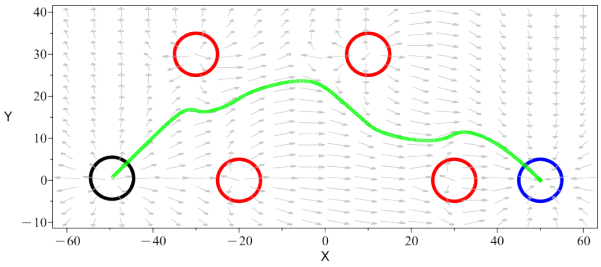

Fig. 3. Numerical simulation result when the source point (in black) is located at (-50, 0) m, the target position (in blue) at (50, 0), and obstacles at: (-30, 30) m, (-20, 0) m, (10, 30) m, and (30, 0) m. The resulting obstacle-avoidance trajectory is highlighted in green

Algorithm 1 Construct of function $\mathbf{E}_{P_U}$

Require:
Source and target coordinate points:
$S\left(x_i, y_i\right), T\left(x_f, y_f\right)$.
n-obstacle coordinate points:
$O_1\left(x_1, y_1\right), \ldots, O_n\left(x_n, y_n\right)$.
Strong charge dipole:
$+Q_S$, $-Q_T$, at the source and target points.
Weak positive charges:
$+Q_1, \ldots, +Q_n$ at the obstacle points.

Ensure: Vector field function:
$\mathbf{E}_{P_U} = E_{P_U x}\left(x, y\right)_{\mathbf{i}} + E_{P_U y}\left(x, y\right)_{\mathbf{j}}$.

Sum $\mathbf{E}_P \leftarrow 0$
Point $\leftarrow \{(x_i, y_i), (x_1, y_1), \ldots, (x_n, y_n), (x_f, y_f)\}$
Charge $\leftarrow \{+Q_S, +Q_1, \ldots, +Q_n, -Q_T\}$
$(n+2)$ charges
for k from 1 to $(n+2)$ **do**
Substitute: the k-th coordinate Point$[k]$, and the k-th *Charge*$[k]$ in $\mathbf{E}_P$
Sum $\mathbf{E}_P \leftarrow$ Sum $\mathbf{E}_P + \mathbf{E}_P$
end for
$\mathbf{E}_{P_U} \leftarrow \left(\dfrac{\text{Sum}\,\mathbf{E}_P}{|Sum\mathbf{E}_P|_1}\right)$

Now, if we define the directional vector $\mathbf{R}_n = (x - x_n)_{\mathbf{i}} + (y - y_n)_{\mathbf{j}}$ (where $\mathbf{i}$ and $\mathbf{j}$ are unit vectors), then the distance r between the two points, $P_n(x_n, y_n)$ and $P(x, y)$, can be described by the norm or magnitude of $\mathbf{R}_n$.

With this, Equation 1 can be rewritten as:

$$|\mathbf{E}_P| = K\frac{Q}{|\mathbf{R}_n|^2}. \tag{2}$$

And the normalization of $\mathbf{R}_n$, here denoted as $\mathbf{a}_{R_n}$, is given by:

$$\mathbf{a}_{R_n} = \frac{\mathbf{R}_n}{|\mathbf{R}_n|}. \tag{3}$$

Hence, using Equation 2 and Equation 3, the electric field vector $\mathbf{E}_P = |\mathbf{E}_P|\mathbf{a}_{R_n}$ can be expressed in a general form as:

$$\mathbf{E}_P = K\,\frac{Q}{|\mathbf{R}_n|^2}\frac{\mathbf{R}_n}{|\mathbf{R}_n|} = KQ\,\frac{\mathbf{R}_n}{|\mathbf{R}_n|^3}. \tag{4}$$

3 Euclidean Norm Versus L1-norm

In the traditional scheme of electric field computation, the variable r in Equation 1 is given by the Euclidean norm (L2-norm) of $\mathbf{R}_n$. However, in accordance with our proposal, we replace r by the L1-norm of $\mathbf{R}_n$. Formally, the norm of any vector is defined as follows:

Definition 1. Given a n-dimensional vector $\mathbf{V} = [x_1,\ x_2,\ \cdots,\ x_n]^T$, its vector norm $|\mathbf{V}|_p$, for $p = 1,\ 2,\ \cdots,\ n$, is defined as:

$$|\mathbf{V}|_p = \left(\sum_i^n |x_i|^{(p)}\right)^{\frac{1}{p}}. \tag{5}$$

Based on Definition 1, we have the magnitude of $|\mathbf{R}_n|$:

– In L1-norm:

$$|\mathbf{R}_n|_1 = |x - x_n| + |y - y_n|. \tag{6}$$

– In L2-norm (Euclidean norm):

$$|\mathbf{R}_n|_2 = \sqrt{|x - x_n|^2 + |y - y_n|^2}. \tag{7}$$

And the corresponding normalization of $\mathbf{R}_n$, is now expressed as:

– In L1-norm:

Algorithm 2 Tracing of the collision free trajectory

Require:
Source and target coordinate points:
$S(x_i, y_i), T(x_f, y_f)$,
Vector field function: $\mathbf{E}_{P_U}$,
Starting angle θ: $(\tan\theta = \frac{\Delta y}{\Delta x})$,
Step size h.
Ensure: Collision free trajectory: Path
$k \leftarrow 1$
$x_0 = x_i + \Delta x \qquad \left(\Delta x = \dfrac{\Delta y}{\tan\theta}\right)$
$y_0 = y_i + \Delta y \qquad (\Delta y = (\Delta x)\tan\theta)$
while Condition $\leftarrow$ false **do**
$x \leftarrow x_{k-1}$
$y \leftarrow y_{k-1}$
$x_k \leftarrow x + hE_{P_U x}$
($E_{P_U x}$ is the **i** component of $\mathbf{E}_{P_U}$)
$y_k \leftarrow y + hE_{P_U y}$
($E_{P_U y}$ is the **j** component of $\mathbf{E}_{P_U}$)
$k \leftarrow k + 1$
(Repeat M-times until Condition $\leftarrow$ true)
end while
Path=$\{(x_0, y_0), (x_1, y_1), \ldots, (x_k, y_k), \ldots, (x_M, y_M)\}$

$$\mathbf{a}_{R_{n1}} = \frac{\mathbf{R}_n}{|\mathbf{R}_n|_1}. \tag{8}$$

That can be written explicitly as follows:

$$\mathbf{a}_{\mathbf{R_{n1}}} = \frac{\Delta x_n}{|\Delta x_n| + |\Delta y_n|}\mathbf{i} + \frac{\Delta y_n}{|\Delta x_n| + |\Delta y_n|}\mathbf{j}. \tag{9}$$

Or in L2-norm (Euclidean norm):

$$\mathbf{a}_{R_{n2}} = \frac{\mathbf{R}_n}{|\mathbf{R}_n|_2}. \tag{10}$$

Which can also be written as:

$$\mathbf{a}_{R_{n2}} = \frac{\Delta x_n}{\sqrt{|\Delta x_n|^2 + |\Delta y_n|^2}}\mathbf{i} + \frac{\Delta y_n}{\sqrt{|\Delta x_n|^2 + |\Delta y_n|^2}}\mathbf{j}. \tag{11}$$

With $\Delta x_n = x - x_n$, $\Delta y_n = y - y_n$. Clearly, a higher complexity in the numerical evaluation of Equation 10 compared to Equation 8 can be observed. This premise is supported by the numerical simulation results depicted in Fig. 2 (performed in MAPLE), where the running time necessary for the computation and addition of a set of normalized vectors $\mathbf{a}_{R_{n1}}$, from 1 to 20,000 consecutive coordinate points, is contrasted with the corresponding running time required for the computation and addition of $\mathbf{a}_{R_{n2}}$ within the same set of points. Considering the results shown in Fig. 2, we propose using the L1-norm instead of the traditional Euclidean norm (L2-norm) for calculating the electric field. Hence, the electric field vector expressed in Equation 4 can be rewritten as:

$$\mathbf{E}_P = KQ\frac{\mathbf{R}_n}{|\mathbf{R}_{n1}|^3}. \tag{12}$$

And after substituting Equation 6 we obtain:

$$\mathbf{E}_P = \frac{KQ\Delta x_n}{(|\Delta x_n| + |\Delta y_n|)^3}\mathbf{i} + \frac{kQ\Delta y_n}{(|\Delta x_n| + |\Delta y_n|)^3}\mathbf{j}. \tag{13}$$

From this result, a generalized expression for the field generated by N-charges, at any arbitrary coordinate point $P(x, y)$, can be expressed as:

$$\mathbf{E}_P = \sum_{n=1}^{N} KQ\frac{\mathbf{R}_n}{|\mathbf{R}_{n1}|^3}. \tag{14}$$

Or in the form of a vector field as follows:

$$\mathbf{E}_P = E_{P_x}(x, y)_{\mathbf{i}} + E_{P_y}(x, y)_{\mathbf{j}}. \tag{15}$$

With its respective components:

$$E_{P_x}(x, y) = \sum_{n=1}^{N} \frac{KQ_n\Delta x_n}{(|\Delta x_n| + |\Delta y_n|)^3}. \tag{16}$$

And:

$$E_{P_y}(x, y) = \sum_{n=1}^{N} \frac{KQ_n\Delta y_n}{(|\Delta x_n| + |\Delta y_n|)^3}. \tag{17}$$

Finally, it is possible to express the normalization of the vector field $\mathbf{E}_P$ (here denoted $\mathbf{E}_{P_U}$) as:

$$\mathbf{E}_{P_U} = E_{P_{Ux}}\mathbf{i} + E_{P_{Uy}}\mathbf{j}, \tag{18}$$

where

$$E_{P_{Ux}} = \frac{E_{P_x}(x, y)}{|E_{P_x}(x, y)| + |E_{P_y}(x, y)|}. \tag{19}$$

And

$$E_{P_{Uy}} = \frac{E_{P_y}(x, y)}{|E_{P_x}(x, y)| + |E_{P_y}(x, y)|}. \tag{20}$$

ISSN 2007-9737

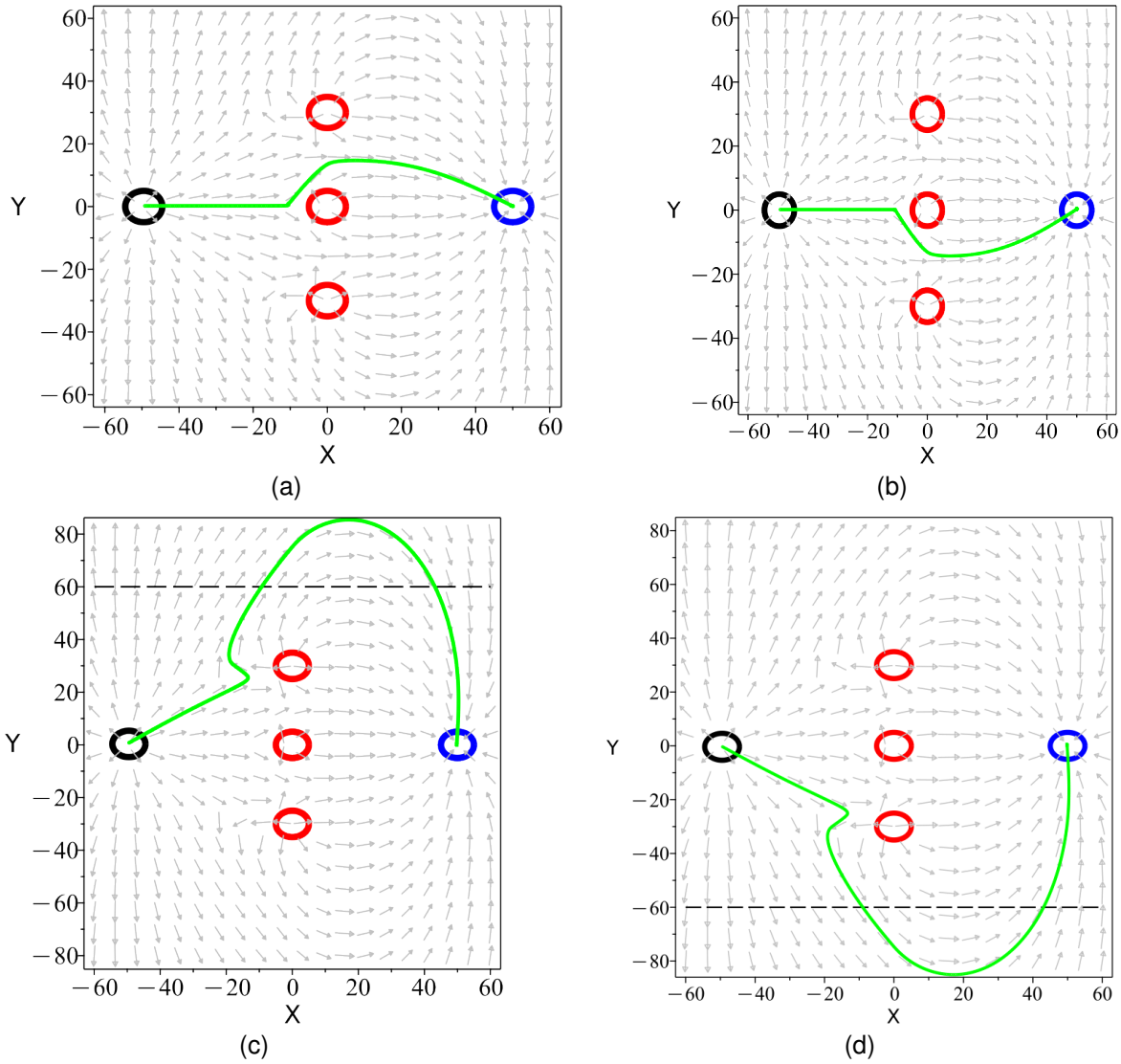

Fig. 4. Numerical simulations for the case of study with three obstacles (in red) at (0, 30) m, (0, 0) m, and (0, -30) m. The source point (in black) at (-50, 0) m and the target point (in blue) at (50, 0). The charge of the electric dipole is $\pm 10 \times 10^{-6}$ μC while each one of the obstacle charges is 1×10^{-6} μC. Four different starting angles are considered: (a) $0.057°$, (b) $-0.057°$, (c) $45°$ and (d) $-45°$. The resulting obstacle-avoidance trajectory is highlighted in green. The dashed black line indicates the upper and down spatial boundaries in the two-dimensional space

4 Path Planning Proposal

Consider a two-dimensional navigation space with a set of obstacles. Suppose we are interested in finding an obstacle-avoidance trajectory that connects two arbitrary positions defined as source and target points. To overcome this problem, we set two point charges, positive and negative, into the source and target location, respectively. These charges must be not only opposite in sign but also strong in value to form a predominant electric dipole whose vector field could be able to provide a natural path from the source to the target coordinate.

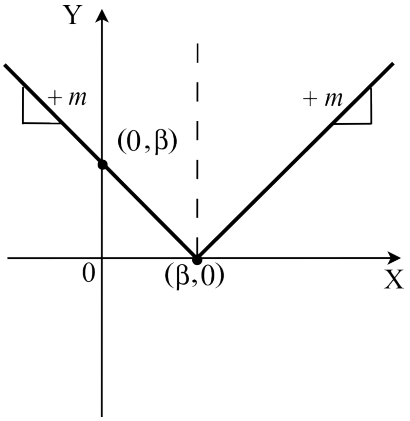

Fig. 5. Graph of function $y_a^+ = m|x - \beta|$

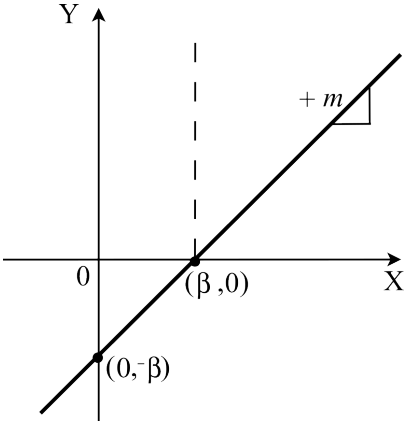

Fig. 6. Graph of function $y_l^+ = m\,(x - \beta)$

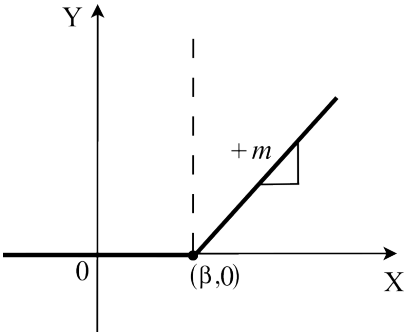

Fig. 7. Graph of function $y_p^+ = \dfrac{m}{2}\,(|x - \beta| + (x - \beta))$

Moreover, to ensure achieving a collision-free trajectory, we must locate at each obstacle position one weak positive charge. The viability of this approach can be verified by the case study shown in Fig. 3.

In this example, an electric dipole of charges of $\pm 10 \times 10^{-6}$ μC is considered. The positive charge is placed at the source coordinate (-50, 0) m (in black), and the negative one at the target point (50, 0) m (in blue). Four weak positive charges (each one of 1×10^{-6} μC) are also placed at the obstacle positions (in red): (-30, 30) m, (-20, 0) m, (10, 30) m, and (30, 0) m.

Under this scenario, and considering all of these charges, our approach is based on determining the vector function of the total electric field that results from the sum of the single electric field contributions produced by each charge (in accordance with Equation 18). Hence, a collision-free trajectory is directly obtained from the vector function $\mathbf{E}_{P_U}$.

In this regard, it is important to mention that there are multiple paths that can be selected by considering the graph of the total electric vector field map. The chosen path will depend on the starting angle of the trajectory. In this case of study, a $45°$ angle was used. In Fig. 3, the graph of the vector field function is plotted in gray color.

As it can be observed, the electric field lines always points away from the positive dipole charge (source) toward the negative dipole charge (target), and in accordance with the chosen starting angle, the expected obstacle avoidance trajectory was successfully achieved (highlighted in green).

The numerical procedure to trace the collision-free trajectory from the source to target points is divided into two algorithms: Algorithm 1, which addresses the problem of constructing the vector field function $\mathbf{E}_{P_U}$, and Algorithm 2, which evaluates the constructed vector field function to trace the collision-free trajectory.

However, while this approach consistently allows us to find a collision-free trajectory from the source positive charge to the target negative charge, there is a shortcoming that needs to be addressed. This problem involves the uncertainty of ensuring that all the generated paths will remain within a two-dimensional bounded space.

To exemplify this problem, consider the four trajectory scenarios shown in Fig. 4 where the following positions are considered: source (-50, 0) m, target (50, 0) m, and obstacles (0, -30) m, (0, 0) m, and (0, 30) m.

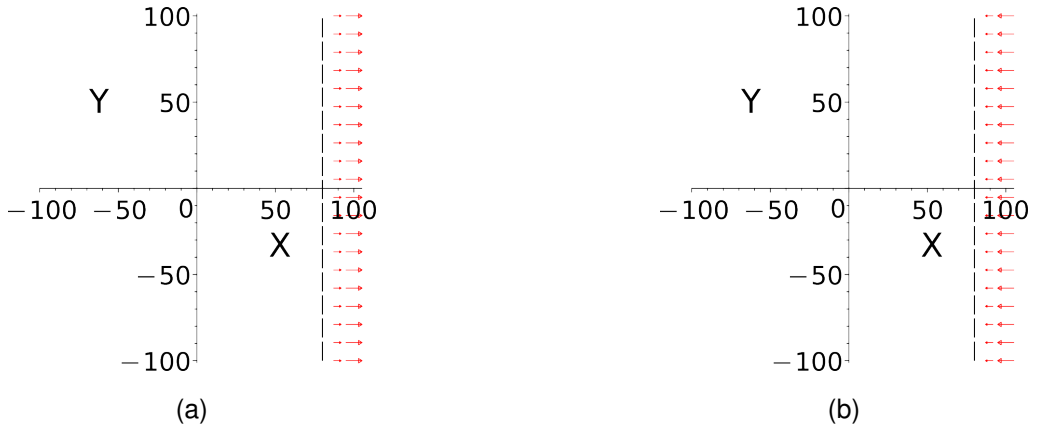

Fig. 8. Graph of vector functions used to define a virtual border boundary at $\beta = 80$, (a) $\mathbf{E}_{y_p^+}$ where $y_p^+ = -40 + \frac{x}{2} + \frac{|x-80|}{2}$, (b) -$\mathbf{E}_{y_p^+}$ (using $-y_p^+$)

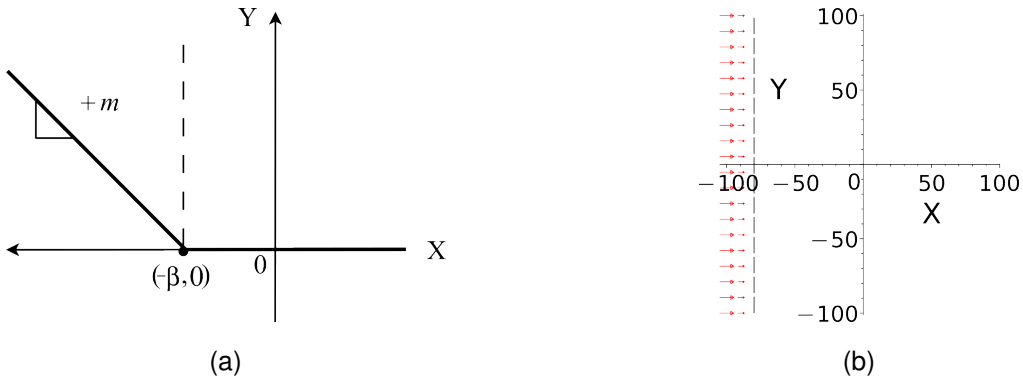

Fig. 9. Graphs of functions: (a) The general graph of y_p^-, (b) Graph of the vector function $\mathbf{E}_{y_p^-}$ used to define the virtual border at $\beta = -80$. The corresponding y_p^- function in this example is $y_p^- = -40 - \frac{x}{2} + \frac{|x+80|}{2}$

The charge of the electric dipole is $\pm 10 \times 10^{-6}$ μC while each one of the obstacle charges is 1×10^{-6} μC. As it was already mentioned, the obtained trajectory is highly dependent on the starting angle. The scenarios shown in Fig. 4 illustrates the following cases:

a) Slight positive angle $(0.057°)$ to force avoiding the central obstacle by passing over it.
b) Slight negative angle $(-0.057°)$ to force avoiding the central obstacle by passing bellow it.
c) A $45°$ angle in order to achieve a trajectory passing over the obstacle in (0, 30) m.
d) A $-45°$ angle in order to achieve a trajectory passing bellow the obstacle in (0, -30) m.

Now let us assume that we are restricted to the region that is bounded by two borders at $Y = \pm 60$ m. Clearly it can be observed that the scenarios illustrated in Fig. 4a and Fig. 4b fulfill this spatial restriction, in contrast, the trajectories achieved in the cases reported in Fig. 4c and Fig. 4d, do not satisfy it because both trajectories cross out the set boundaries. In this sense, the inability of restricting the trajectory into a defined region of interest, is a limitation that must be overcome in order to achieve a practical application:

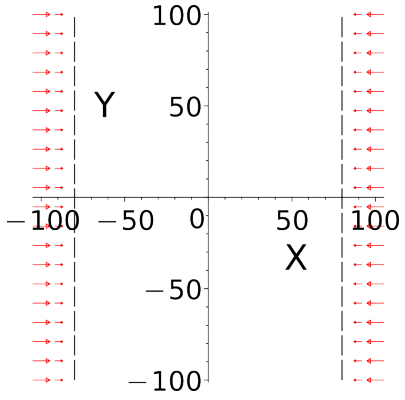

Fig. 10. Construction of vertical field border walls. The graph of the vertical field border function expressed by the sum: $-\mathbf{E}_{y_p^+} + \mathbf{E}_{y_p^-}$ is highlighted in red.

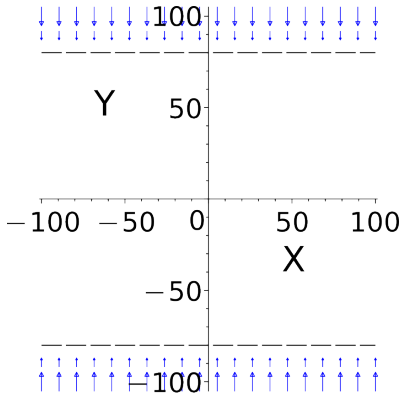

Fig. 11. Construction of horizontal field border walls. The graph of the horizontal field border function expressed by the sum: $-\mathbf{E}_{x_p^+} + \mathbf{E}_{x_p^-}$ is highlighted in blue

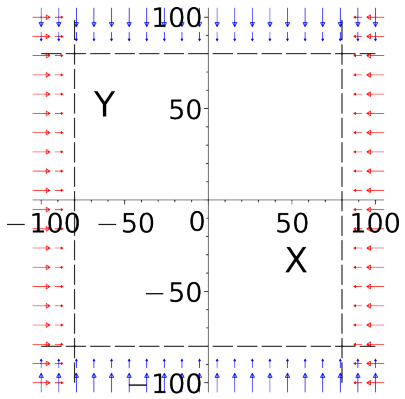

Fig. 12. Graph of the vector function $\mathbf{E}_b$ that models the four sides of the bounded rectangular space. In this example the delimiting functions are (21)-(24)

$$y_p^+ = -40 + \frac{x}{2} + \frac{|x-80|}{2}, \tag{21}$$

$$y_p^- = -40 - \frac{x}{2} + \frac{|x+80|}{2}, \tag{22}$$

$$x_p^+ = 40 - \frac{y}{2} - \frac{|y-80|}{2}, \tag{23}$$

$$x_p^- = -40 - \frac{y}{2} + \frac{|y+80|}{2}. \tag{24}$$

5 Defining a Spatial Area of Interest for the Generated Trajectories

In this section, we address the problem of defining a rectangular area of interest for all the source-to-target trajectories derived from the computation of the total electric field.

To achieve this objective, we propose to model the rectangular space with a vector field produced by piecewise-linear functions [5, 3, 1, 2], here denoted as y_p^+, y_p^-, x_p^+ and x_p^-.

To clarify this issue, we present the procedure to obtain these functions, their algebraical form, and their graphical characteristics. Firstly, we consider the functions y_p^+ and y_p^-. In this regard, let us introduce the shifted form of the absolute value function y_a^+ as follows:

$$y_a^+ = m|x-\beta|. \tag{25}$$

Fig. 5 shows that the graph of Equation 25 is composed of two symmetrical straight lines: one with a negative slope $-m$ and one with a positive slope $+m$. This graph is shifted to the breakpoint $(\beta, 0)$. We also consider the form of a shifted linear equation y_l^+ given by:

$$y_l^+ = m\,(x-\beta)\,. \tag{26}$$

Fig. 6 shows the graph of Equation 26 whose shape consists of a horizontal shift to the right straight line with a vertical intersection at the point $(0, -\beta)$. Hence, we create a new function y_p^+ by adding Equation 25 and Equation 26, and dividing by 2 as follows:

$$y_p^+ = \frac{y_a^+ + y_l^+}{2} \text{ or } y_p^+ = \frac{m}{2}\left(|x-\beta| + (x-\beta)\right). \tag{27}$$

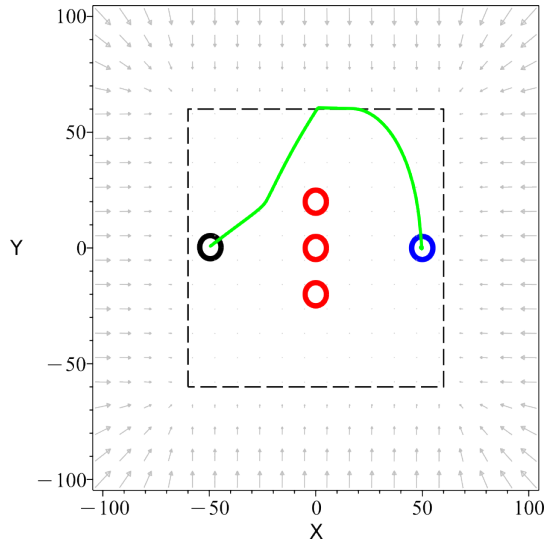

Fig. 13. The same situation shown in Fig. 4c, but in these new simulation, a square region enclosed by a vector field function (in the form of Equation 41) for $\beta = 60$, has been included

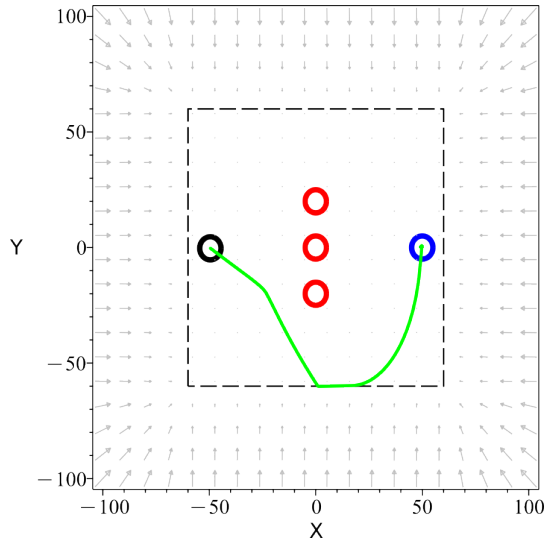

Fig. 14. The same situation shown in Fig. 4d, but in these new simulation, a square region enclosed by a vector field function (in the form of Equation 41) for $\beta = 60$, has been included

The purpose of creating y_p^+ is to construct a function whose value is equal to zero for a domain defined from $-\infty$ to the breakpoint $(\beta, 0)$, and once this point is reached, it monotonically increases its value by a constant ratio.

Based on this premise, the graph of y_p^+ (shown in Fig.7) can be obtained by the sum of each one of the slopes of the straight line segments given in the graphs of y_a^+ and y_l^+. As expected, it can be verified that along to the X-axis (in the interval $x \leq \beta$) there is an horizontal segment due to the slope cancellation $(+m - m = 0)$, while in the interval $x \geq \beta$ there is a linear segment with a slope of $+m$ (obtained by $\frac{1}{2}(m + m) = m$). From this result, we use Equation 27 to construct the vector field function $\mathbf{E}_{y_p^+}$ as:

$$\mathbf{E}_{y_p^+} = \left(y_p^+\right)_{\mathbf{i}}. \tag{28}$$

To illustrate this, consider the problem of defining a rectangular region defined by the vertices (-80, -80) m, (80, -80) m, (80, 80) m, and (-80, 80) m. In accordance with Equation 27, for this example we have:

$$y_p^+ = -40 + \frac{x}{2} + \frac{|x - 80|}{2}. \tag{29}$$

With $m = 1$ and $\beta = 80$. Figure 8a shows the field lines produced by the vector function $\mathbf{E}_{y_p^+}$. Clearly, it can be seen that this graph describes a border which can be used to model one of the sides of the required rectangular bounded space.

However, with the aim of using this field as a barrier of trajectories, it must be opposite in sign (-$\mathbf{E}_{y_p^+}$) as shown in Fig. 8b. Following the same procedure, and knowing that $|x - \beta| = |-x + \beta|$, we replace x by $-x$ in Equation 27 to obtain the horizontal-reflected function of y_p^+, here denoted as y_p^-:

$$y_p^- = \frac{m}{2}\left(|x + \beta| - (x + \beta)\right), \tag{30}$$

where:

$$y_a^- = m|x + \beta|, \tag{31}$$

$$y_l^- = -m\left(x + \beta\right). \tag{32}$$

Similarly, we use Equation 30) to construct the vector function $\mathbf{E}_{y_p^-}$ as:

$$\mathbf{E}_{y_p^-} = \left(y_p^-\right)_{\mathbf{i}}. \tag{33}$$

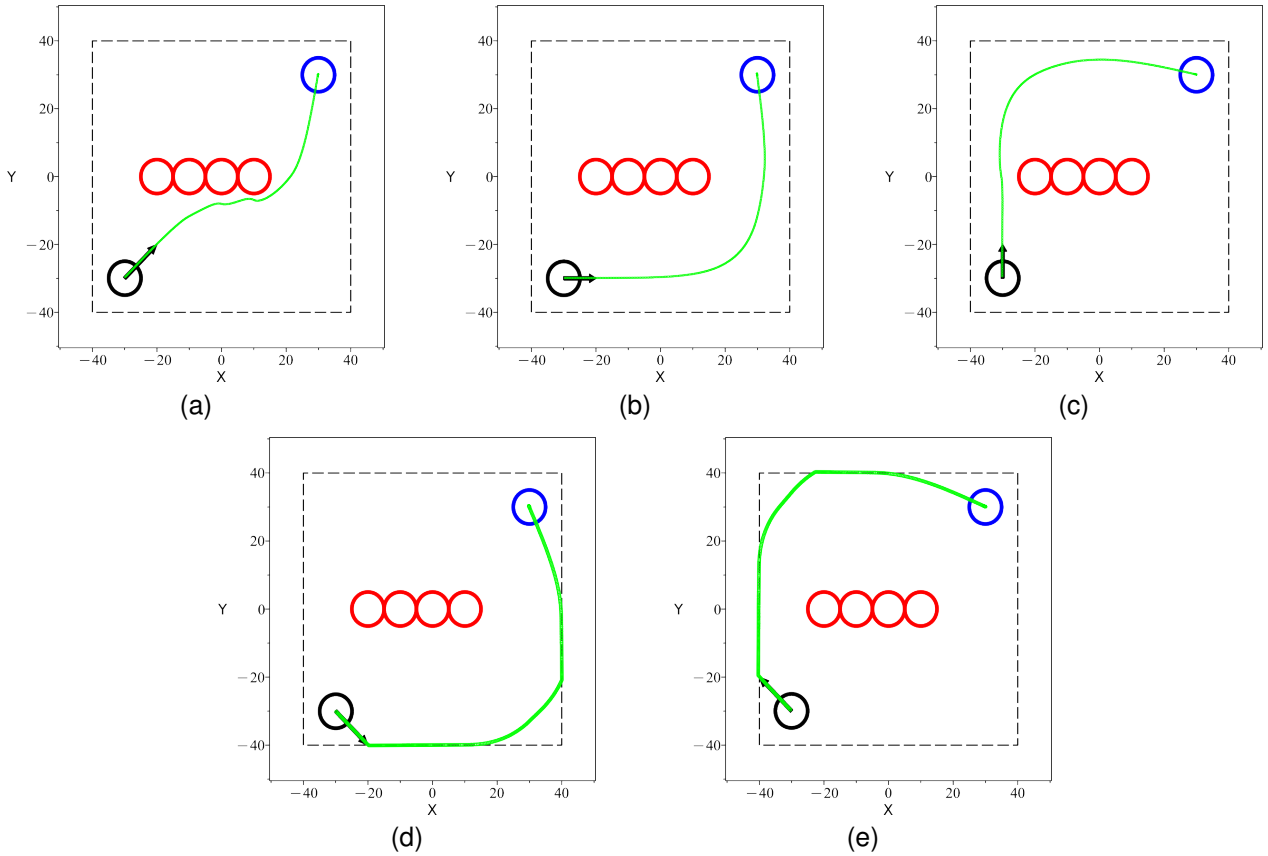

Fig. 15. Simulations performed in order to evaluate the effectiveness of the border vector field function $\mathbf{E}_b$ by different starting angles: (a) $45°$, (b) $0°$, (c) $90°$, (d) $315°$, and (e) $135°$

The general graph of Equation 30, and the specific graph of Equation 33 for the example here considered ($\beta = 80$), are depicted in Fig.9(a) and Fig.9(b), respectively. In this case, it is important to highlight that Equation 33 does not need to change its sign to provide a suitable field barrier of trajectories, and the corresponding y_p^- function, for our case of study, is given by:

$$y_p^- = -40 - \frac{x}{2} + \frac{|x+80|}{2}. \tag{34}$$

With $m = 1$ and $\beta = -80$. With all these elements, we are able to construct the required rectangular bounded space. To do this, we firstly add the two vector fields (the negative of Equation 28 and Equation 33) in order to construct the two vertical field borders shown in Fig.10.

Then, in a similar way, we perform the change of variable $x = y$ with the purpose of constructing two horizontal field borders $\mathbf{E}_{x_p^-}$ and $\mathbf{E}_{x_p^-}$ described as follows:

$$\mathbf{E}_{x_p^+} = \left(x_p^+\right)_{\mathbf{j}}. \tag{35}$$

And:

$$\mathbf{E}_{x_p^-} = \left(x_p^-\right)_{\mathbf{j}}, \tag{36}$$

where x_p^+ and x_p^- have the form:

$$x_p^+ = \frac{m}{2}\left(|y-\beta| + (y-\beta)\right), \tag{37}$$

$$x_p^- = \frac{m}{2}\left(|y+\beta| - (y+\beta)\right). \tag{38}$$

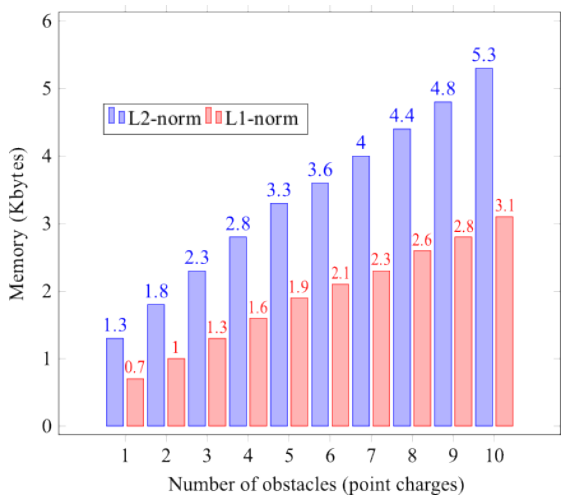

Fig. 16. Memory used by the symbolic expression (text file size expressed in Kbytes) of the vector function, depending on the number of obstacles considered $(1, 2, \ldots, 10)$

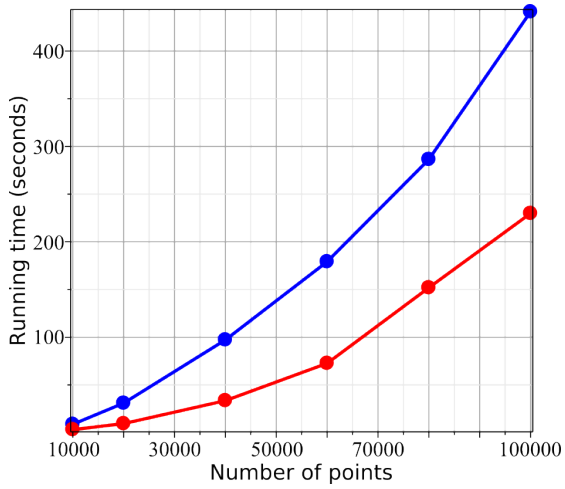

Fig. 17. Running time consumed by the simulation for the evaluation of the vector field function (expressed in seconds). Various scenarios with different numbers of points to evaluate were taken into account (10×10^3, 20×10^3, 40×10^3, 60×10^3, 80×10^3, 100×10^3). The results obtained using the L1 norm are depicted in red, and the results obtained by the Euclidean norm are highlighted in blue

Which, in accordance with our example, it can be described as:

$$x_p^+ = 40 - \frac{y}{2} - \frac{|y-80|}{2}. \tag{39}$$

With $m = 1$ and $\beta = 80$, and:

$$x_p^- = -40 - \frac{y}{2} + \frac{|y+80|}{2}. \tag{40}$$

With $m = 1$ and $\beta = -80$. The graph of the horizontal field border function given by the sum: $-\mathbf{E}_{x_p^+} + \mathbf{E}_{x_p^-}$ is shown in Fig. 11.

Finally, we merge both border fields, vertical and horizontal, to construct the rectangular bounded space shown in Fig. 12, where the vector field function whose graph models the four sides of the bounded rectangular space shown in Fig. 12, is given by:

$$\mathbf{E}_b = (E_v)_{\mathbf{i}} + (E_h)_{\mathbf{j}}. \tag{41}$$

With:

$$E_v = -y_p^+ + y_p^-. \tag{42}$$

And:

$$E_h = -x_p^+ + x_p^-. \tag{43}$$

And in our case study, it is written as:

$$E_v = -x - \frac{|x-80|}{2} + \frac{|x+80|}{2}, \tag{44}$$

$$E_h = -y - \frac{|y-80|}{2} + \frac{|y+80|}{2}. \tag{45}$$

6 Simulation Results

In Fig.13 and Fig.14, the same situations shown in Fig. 4(c) and Fig. 4(d) are simulated. However, in these new simulations, the area of interest is delimited by a square region bordered by vector field function (in the form of Equation 41) whose vertices lie on the coordinate points:

– (-60, -60) m,
– (60, -60) m,
– (60, 60) m,
– (-60, 60) m.

With the inclusion of this border region, we show that the source to target path always keeps inside the enclosed area.

ISSN 2007-9737

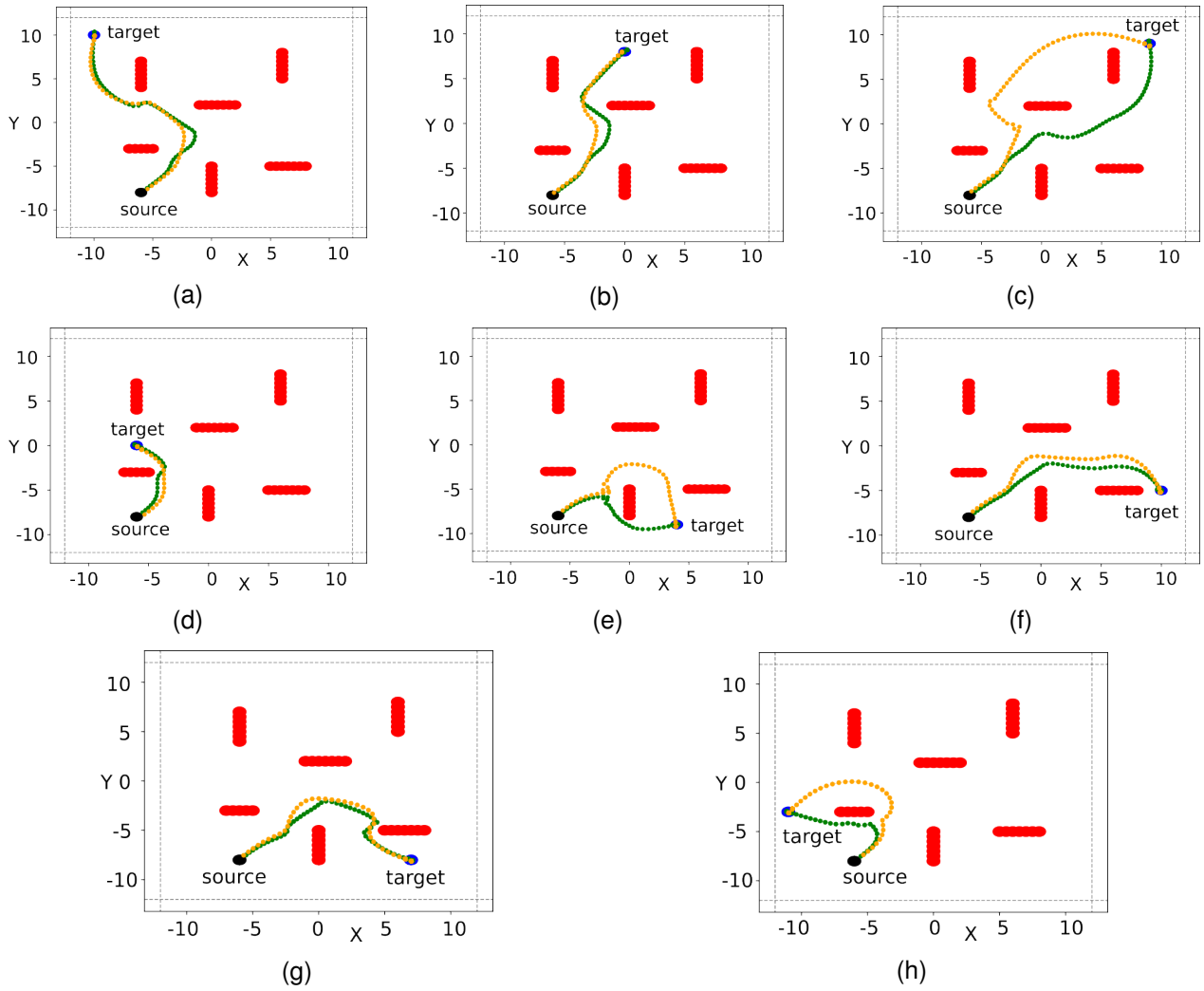

Fig. 18. Python simulations. Case of study: 40 obstacles arranged as 6 fixed linear obstacles (8 different scenarios). The trajectories resulting from Euclidean norm computations are distinguished by an orange color, while the trajectory derived from the L1-norm is represented in green color

6.1 Example 1: Different Starting Angles

To illustrate the effectiveness of the bounded region produced by the vector field function $\mathbf{E}_b$, we consider five new cases of study. In these cases, a situation with four obstacles at positions (-20, 0) m, (-10, 0) m, (0, 0) m, and (10, 0) m, located in a horizontal line as shown in Fig. 15, is considered. Moreover, a border field with the vertices (-40, -40) m, (40, -40) m, (40, 40) m, and (-40, 40) m, is also considered.

The charge of the electric dipole is $\pm 100 \times 10^{-6}$ μC while each one of the obstacle charges is 2×10^{-6} μC. The cases shown in Fig. 15a, Fig. 15b, and Fig. 15c correspond with the starting angles: $45°$, $0°$, and $90°$, respectively.

In these specific cases, based on the direction provided by their starting angles, it can be clearly expected that the resulting path reaches the target position. In contrast, in the cases shown in Fig. 15d and Fig. 15e (starting angles of $315°$ and $135°$), the

trace of a successful source to target path can not be guaranteed. However, in all of these cases, it can be observed that the field border not only helps to establish limits or boundaries in a given region of interest, but also aids to achieve a successful source to target trajectory.

6.2 Running Time and Memory

In order to analyze the performance of the proposed methodology, two parameters were taken into consideration: the running time required to trace the collision-free trajectory and the amount of memory necessary to store the symbolic expression of the vector field function.

Regarding the running time, it is important to note that it is highly sensitive to the number of evaluation points (step size), whereas the amount of memory will primarily depend on the number of obstacles (number of charges) considered.

To observe the behavior of these parameters, simulations were performed by MAPLE where the number of obstacles was gradually increased from 1 to 10. From these simulations, it was observed that the symbolic expressions (saved in text format) of the field vector functions derived from the L1-norm required less memory than those obtained from Euclidean norm.

Regarding the running time, simulations with 10 obstacles were conducted, considering various numbers of points (10×10^3, 20×10^3, 40×10^3, 60×10^3, 80×10^3, 100×10^3). From these simulations, an exponential increase in the time required to evaluate the vector functions and trace the avoidance trajectory derived from the Euclidean norm, was observed. These simulation results can be observed in Fig. 16 and Fig. 17.

7 Example 2: More Obstacles and Different Target Positions

In Fig.18, we consider a more sophisticated simulation case of study (performed in Python) in which 40 obstacles (charges) have been arranged as 6 fixed line obstacles. In this case, the source is located at a fixed coordinate, while the target point is assigned to 8 arbitrary positions.

Moreover, in all of these cases, we have restricted the trajectories into a region constrained by the border field whose vertices are given by the points: (-12, -12) m, (12, -12) m, (12, 12) m, and (-12, 12) m (depicted by dashed lines). In order to ensure the attractiveness of the target position, in this case, instead of an electric dipole, an electric charge of -400×10^{-6} μC has been assigned to the target position while 200×10^{-6} μC to the source position.

The line obstacles contain arrays composed of a set of point charges; each of the obstacle charges is 10×10^{-6} μC. With the purpose of being able to compare the results obtained by the two types of norms addressed in this study, the trajectories generated by the L2-norm or Euclidean norm (orange color) and L1-norm (green color), have been depicted in Fig. 18.

Additionally, it is important to emphasize that both norms let to achieve successful trajectories between the source and target points. Moreover, it was observed that the running time of L1-norm (4.14×10^{-3} seconds) was shorter than the running time of the Euclidean norm (4.76×10^{-3} seconds). Similarly, the memory required to store the symbolic vector field function $\mathbf{E}_{P_U}$ for the L1-norm (11844 bytes) is lower than for the Euclidean norm (20503 bytes).

8 Conclusion and Future Work

In this paper, we introduce a novel approach based on the physics concept of the electric field for determining collision-free trajectories between source and target points in a two-dimensional space.

This approach considers two noteworthy features: the substitution of the conventional Euclidean norm with the L1-norm and the integration of border field functions to delineate the navigation area.

Through numerical simulations, we have validated the efficacy of our proposal showing that when compared to the traditional electric field analysis based on the Euclidean norm, our proposal significantly enhances computing performance.

For instance, employing the L1-norm requires less memory for storing symbolic expressions of the computed field than in the case of using the Euclidean norm, and similarly, the evaluation time for such expressions is also lower in the case of L1-norm. Now, the future work of our research focuses on finding criteria for selecting the value of the positive and negative charges (at the source and target positions) and obstacles. We are also exploring the possibility of new vector functions to delineate arbitrary navigation areas, not limited to rectangular shapes.

References

1. **Chua, L., Ying, R. (1983).** Canonical piecewise-linear analysis. IEEE Transactions on Circuits and Systems, Vol. 30, No. 3, pp. 125–140. DOI: 10.1109/TCS.1983.1085342.
2. **Chua, L. O., Deng, A. C. (1985).** Canonical piecewise-linear analysis-II: Tracing driving-point and transfer characteristics. IEEE Transactions on Circuits and Systems, Vol. 32, No. 5, pp. 417–444. DOI: 10.1109/TCS.1985.1085744.
3. **Chua, L. O., Deng, A. C. (1986).** Canonical piecewise-linear modeling. IEEE Transactions on Circuits and Systems, Vol. 33, No. 5, pp. 511–525. DOI: 10.1109/TCS.1986.1085952.
4. **Haciomeroglu, E. S., Andreasen, J. B. (2017).** Exploring euclidean and taxicab geometry with dynamic mathematics software. North American GeoGebra Journal, Vol. 6, No. 1, pp. 32–41.
5. **Kahlert, C., Chua, L. O. (1990).** A generalized canonical piecewise-linear representation. IEEE Transactions on Circuits and Systems, Vol. 37, No. 3, pp. 373–383. DOI: 10.1109/31.52731.
6. **Krause, E. (1987).** Taxicab geometry: An adventure in non-euclidean geometry. Dover Publications, New York USA.
7. **Liu, S., Zhang, Q., Zhou, D. (2014).** Obstacle avoidance path planning of space manipulator based on improved artificial potential field method. Journal of The Institution of Engineers: Series C, Vol. 95, No. 1, pp. 31–39. DOI: 10.1007/s40032-014-0099-z.
8. **Melchiorre, M., Salamina, L., Scimmi, L. S., Mauro, S., Pastorelli, S. (2023).** Experiments on the artificial potential field with local attractors for mobile robot navigation. Robotics MDPI, Vol. 12, No. 81, pp. 1–15. DOI: 10.3390/robotics12030081.
9. **Pimenta, L. C. A., Fonseca, A. R., Pereyra, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., Campos, M. F. M. (2005).** Robots navigation based on electromagnetic fields. Proceedings of the 15th Conference on the Computation of Electromagnetic Fields, pp. 154–155.
10. **Pimenta, L. C. A., Fonseca, A. R., Pereyra, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., Campos, M. F. M. (2006).** Robot navigation based on electrostatic field computation. IEEE Transactions on Magnetics, Vol. 42, No. 4, pp. 1459–1462. DOI: 10.1109/tmag.2006.870931.
11. **Pradhan, S. K., Parhi, D. R., Panda, A. K., Behera, R. K. (2006).** Potential field method to navigate several mobile robots. Applied Intelligence, Vol. 25, No. 3, pp. 321–333. DOI: 10.1007/s10489-006-0110-3.
12. **Rafai-Atiyah, A. N., Adzhar, N., Jaini-Izzati, N. (2022).** A review on path planning and obstacle avoidance algorithms for autonomous mobile robots. Journal of Robotics, Vol. 2022, pp. 1–14. DOI: 10.1155/2022/2538220.
13. **Rimon, E., Koditschek, D. (1992).** Exact robot navigation using artificial potential functions. IEEE Transactions on Robototics and Automation, Vol. 8, No. 5, pp. 501–518. DOI: 10.1109/70.163777.
14. **Rostami, S. M. H., Sangaiah, A. K., Wang, J., Liu, X. (2019).** Obstacle avoidance of mobile robots using modified artificial potential field algorithm. EURASIP Journal of Wireless

Communications and Networking, Vol. 70, pp. 1–9. DOI: 10.1186/s13638-019-1396-2.

15. **Zhu, Q., Yan, Y., Xing, Z. (2006).** Robot path planning based on artificial potential field approach with simulated annealing. Proceedings of the 6th International Conference on Intelligent Systems Design and Applications, pp. 622–627. DOI: 10.1109/isda.2006.253908.

Article received on 08/01/2024; accepted on 17/10/2024.
**Corresponding author is Victor Manuel Jimenez-Fernandez.*