Received: 23 May 2023 | Revised: 1 August 2023 | Accepted: 4 September 2023

DOI: 10.1111/exsy.13455

ORIGINAL ARTICLE

Expert Systems WILEY

GBNN algorithm enhanced by movement planner for UV-C disinfection

D. V. Rodrigo [1] | J. E. Sierra-García [2] | M. Santos [3]

[1]Computer Science Faculty, Complutense University of Madrid, Madrid, Spain

[2]Department of Digitalization, University of Burgos, Burgos, Spain

[3]Institute of Knowledge Technology, Complutense University of Madrid, Madrid, Spain

Correspondence
J. E. Sierra-García, Department of Digitalization, University of Burgos, Burgos 09006, Spain.
Email: jesierra@ubu.es

Abstract

In order to maintain adequate levels of cleanliness and sanitation in public facilities, prevent the buildup of viruses and other harmful pathogens, and ensure health and safety, health and labor authorities have repeatedly warned of the need to adhere to proper disinfection protocols in the workplace. This is particularly important in public places where food is handled, where there are more vulnerable people, including hospitals and health care centers, or where there is a large concentration of people. One promising approach is the combination of ultraviolet-C (UV-C) light and mobile robots to automate disinfection processes. Being this technology effective for disinfection, an excessive dose of UV can damage the materials, limiting its applicability. Therefore, a major challenge for automatic disinfection is to find a route that covers the entire surface, ensures cleanliness, and provides the correct radiation dose while preventing environmental materials from being damaged. To achieve this, in this paper a novel intelligent control approach is proposed. A bio-inspired Glasius neural network with a motion planner, an UV estimation module, a speed regulator, and pure pursuit controller are combined into one intelligent system. The motion planner proposes a sequence of movements to go through the space in the most efficient way possible, avoiding obstacles of the environment. The speed controller adjusts the dose of UV-C radiation and the pure pursuit regulator ensures the following of the path. This approach has been tested in various simulation scenarios of increasing complexity and in four different areas of dosing requirements. In simulation, a 44% reduction of the maximum dose is achieved, 17% less distance travelled by the robot and, what is more important, 229% more locations with the appropriate dose.

KEYWORDS

autonomous robot, complete coverage path planning, Glasius bio-inspired neural network, ultraviolet germicidal irradiation

1 | INTRODUCTION

In recent times, disinfection has become increasingly popular for eliminating harmful pathogens from various surfaces, including hospitals, health centers, schools, and other public places. But this disinfection task can be time consuming and even dangerous, thus its automation would be a very positive technological advance (Burgos-Artizzu et al., 2007).

Disinfection by ultraviolet radiation has been widely used in the medical and pharmaceutical sector. Ultraviolet-C (UV-C) light may be very effective because, depending on the UV-C dose, some types of micro-organisms are deactivated as a result of nucleic acid breakdown (Kowalski et al., 2020).The use of this technique in combination with mobile robotics expands the capabilities of UV-C radiation. In these days, when pathogens are on the spot, the combination of robotics developments, such as the generation of routes for surface coverage, with disinfection tools that have been extensively validated in microbiology studies, represents a breakthrough in the day-to-day safety of the general population.

There are several companies that have developed autonomous robots for disinfection with UV-C light. Nevertheless, it must be ensured that the environment receives a sufficient dose of disinfection without exceeding certain limits in order to prevent material damages. To this aim, in this work we propose an algorithm, applicable to any disinfection autonomous robot, which combines complete coverage techniques with intelligent robot speed controllers that adjust the dose. Specifically, an intelligent control architecture that combines neural networks, a motion planner, a pure pursuit controller and a speed controller has been proposed.

Among different path planning methods (San Juan et al., 2018), those using bioinspired algorithms have recently attracted extensive attention (Li et al., 2019). The bio-inspired Glasius neural network (biGBNN) algorithm can be applied to the complete coverage path planning (CCPP) of any autonomous robot. But the biGBNN may lead to unsatisfactory performance in complex environments, so different modifications have been proposed (Chen & Zhu, 2019; Yao & Zhao, 2021). Indeed, in this work we propose the use of a motion planner that modifies the behaviour obtained by the neural activity of the GBNN to cover the space in the most efficient way possible. This approach is aligned with previous works that show that the combination of conventional and intelligent control techniques provide successful results in mobile robot applications (Sierra-Garcia & Santos, 2022).

The main contributions of the automatic surface disinfection strategy here proposed can be summarized as follows.

- Design of a movement planner that considers the obstacles in the environment and the motion strategy.
- Improvement of the GBNN algorithm to interact with the movement planner, reducing unnecessary robot movements.
- Incorporation of an auxiliary map to the architecture to consider the previous disinfected cells as obstacles.
- Implementation of a speed controller to adjust the dose of UV light while the robot is moving.
- Test the approach in different simulation scenarios of growing complexity.

The algorithm proposed in this article follows either boustrophedon or spiral pattern, which seek full coverage and optimal UV-C radiation. It is considered intelligent because from scarce information it is able to find a route that covers the entire space to be disinfected. Its efficiency has been tested on various scenarios of increasing complexity with very satisfactory results. Even more, areas that require different dose have been considered. The use of this algorithm in hospitals, for instance, will allow to disinfect more rooms with the same robot in a single day, helping to reduce the likelihood of nosocomial infections. It is important to remark that safety issues are at utmost importance when dealing with UV-C radiation. Thus, the solution here proposed is focused on the navigation algorithm followed by an autonomous robot in scenarios without humans beings or animals. This way, any possible harm is avoided.

In addition, when developing a robot algorithm for UV-C disinfection, it is important to prioritize the safety of the population by incorporating safety features, providing proper training to operators, and adhering to established safety guidelines and regulations, including the use of personal protective equipment. By doing so, the use of UV-C disinfection technology can be effective while also ensuring the safety of the population in its operational environment. In this case, the robot would incorporate safety features such as sensors to detect the presence of people in the area and automatically shutdowns as it is meant to work in spaces without people.

The paper structure is as follows. Related works are summarized and discussed in Section 2. Section 3 describes the kinematic model of the differential robot and the ultraviolet distribution model. The control architecture for disinfection is detailed in Section 4. The movement planner and its interaction with the enhanced GBNN algorithm are explained in Section 5. Section 6 discusses the simulation results for different scenarios. The paper ends with the conclusions and future works.

2 | RELATED WORKS

Bio-inspired intelligence has aroused great interest, especially in the last decades. There are many applications, and robotics in general stands out among them. In particular, it has been proved to be very useful in the field of mobile robotics with different objectives. In (Li et al., n.d.) a comprehensive survey of bio-inspired intelligence, with a focus on neurodynamics approaches, is presented. It also describes various robotic applications, particularly path planning and control of autonomous robotic systems.

Complete coverage path planning with Glasius bio-inspired neural network has been investigated for different autonomous vehicles. To mention some papers, Sun et al. (2018) propose a multi-AUVs (Autonomous Underwater Vehicles) full coverage discrete and centralized programming

based on GBNN algorithm (Sun et al., 2018). The new strategy is based on grid map and neural networks. Simulation results confirm that with the proposed algorithm multi-AUVs can plan reasonable and collision-free coverage path and reach full coverage.

Cooperative path planning for unmanned surface vehicles (USVs) is presented in (Yao et al., 2022). The Glasius bioinspired neural network is used to calculate the neural activity for the discretized working space of USV. The ocean current is considered in the definition of neuron connection weight and thus the standard path of single USV for obstacle avoidance is obtained. The Hungarian algorithm is then applied for the task assignment among multi-USVs.

The work by (Xing et al., 2023) proposes a deep reinforcement learning method to achieve complete coverage path planning for an unmanned surface vehicle (USV). The deep Q network (DQN) is used to train the complete coverage path planning strategy of the agent. Considering the task of complete coverage path planning, this paper replaces random actions with a set of actions towards the nearest uncovered grid.

Samarakoon et al. propose a novel online CCPP method for a hinged reconfigurable robot (Samarakoon et al., 2022). It is designed with two sub-methods, global and local coverage path planning that have been implemented adapting a Glasius Bio-inspired Neural Network that performs online path planning considering a fixed shape for the robot. The parameters are configured by genetic algorithms to assure collision-free coverage and access of obstacle regions. Experimental results validate the proposed online CCPP method in heterogeneous environments, including dynamic workspace. The same authors propose in (Muthugala et al., 2022) a novel energy-efficient CCPP method based on GBNN for a ship hull inspection robot. The energy model considers the energy usage of a ship hull maintenance robot due to changes in direction, distance, and vertical position. It has been proved effective for dynamic workplaces since it performs online path planning.

The paper by (Yi et al., 2023) proposes a CCPP for omni-directional robots of varying width. The coverage algorithm is a modified GBNN with n-reconfiguration states. It generates the global path autonomously, which determines the self-reconfigurable robot width as per the *n*-th reconfiguration states so as to increase area coverage in open areas and reduce robot footprint in tight spaces.

To address the many constraints for a multi-robot system to perform a region coverage search task in an unknown environment, (Chen et al., 2022) propose a novel multi-robot distributed collaborative region coverage search algorithm based on GBNN. The model converts the environmental information detected by the robot into dynamic neural activity landscape of GBNN. Besides, they work with several dynamic search sub-teams that collaborate to optimize the solution and obtain the next movement path of each robot. Somehow similar, in (Tan et al., 2023) The global environment information is introduced by Q-learning reinforcement learning method in a complete coverage path planning biologically inspired neural network algorithm. Besides, the Q-learning method is used for path planning at the positions where the accessible way points are changed, which optimizes the path planning strategy near these obstacles.

Closer to our work, (Han et al., 2023) presents a new CCPP strategy with biologically inspired neural network for cleaning robots. The planned path of cleaning robot takes into account the dynamic neural activities and the distribution of obstacles in the environmental map. In addition, an improved dynamic deadlock escape algorithm is presented to select the optimal escape target point. The simulation results show that the proposed CCPP algorithm is able to generate an orderly path in both known and unknown environment. Similarly, (Rodrigo et al., 2023) introduces a preventive deadlock processing algorithm and an escape route generator algorithm in the design of a complete coverage path planning algorithm based on a GBNN. It is applied to disinfection using mobile robotics with UV-C light.

Unlike these works, our algorithms uses obstacle templates to complement the path strategies provided by the EGBNN, and a velocity control to adjust the radiation dosage. This approach reduces unnecessary robot movements, which is translated into a less travelled distance and less rotations to disinfect the same area, and what is more important, less cells with an excess of radiation dosage.

3 | SYSTEM MODELLING

The development of a reliable simulation model plays an important role in in the mobile robot navigation and control (Sánchez et al., 2022). The proposed system combines autonomous mobile robotics for the exploration of environments with the use of UV-C light for disinfection.

3.1 | Kinematic model of a differential robot

A differential robot is represented in Figure 1, where the position $p_I\{x,y,\theta\}$ is expressed in Cartesian coordinates of the inertial frame $\{X_I,Y_I\}$. It is a common a practice to assume that there is not lateral slippage, thus the speed y-component in the robot frame is 0. The position of the center of the real axle (x,y), and the orientation of the robot, θ, are the state variables of the system. The longitudinal speed, V, is defined as the average speed of the two wheels. The angular velocity, W, is the speed difference of each wheel divided by the distance between them.

Then, the kinematic model is defined by the combined action of longitudinal and angular velocities (1), where V_L and V_R are the left and right wheel linear velocity, respectively.

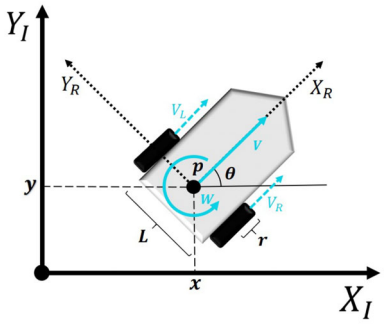


FIGURE 1 Kinematic model of a differential robot in the Cartesian space.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V\cos\theta \\ V\sin\theta \\ W \end{bmatrix} = \begin{bmatrix} \frac{V_L + V_R}{2}\cos\theta \\ \frac{V_L + V_R}{2}\sin\theta \\ \frac{V_R - V_L}{L} \end{bmatrix} \tag{1}$$

3.2 | Ultraviolet distribution model

The factors that most impact on disinfection are humidity and distance from the source of radiation. The germicidal dose is the product of time (t) and intensity (I). The law of the inverse square applies to germicidal ultraviolet light. Therefore, the intensity of the light decreases as the distance (d) from the lamp increases.

$$I \propto 1/d^2 \tag{2}$$

When a microorganism is exposed to UV-C radiation, cell nuclei are modified due to photolytic processes. As a result, cell division and reproduction are prevented. The relationship between dose and deactivation of a target microorganism can be expressed as (Arguelles, n.d.):

$$\frac{N(t)}{N_0} = e^{-klt} = e^{-kD} \tag{3}$$

where N is the number of microorganisms; k is a constant value dependent on the type of microorganism; I is the UV-C intensity; t is time; and D is the received UV-C dose. The UV-C dose (energy per unit area) is the product of UV-C intensity (power per unit area) by exposure time (s):

$$D\left(\frac{mJ}{cm^2}\right) = I\left(mW/cm^2\right) \cdot t(s) \tag{4}$$

As the UV-C may be harmful for human health, all safety regulations must be strictly complied when these technologies are applied. All workers exposed to risks from radiation receive all necessary information and training relating to the outcome of the risk assessment (including the use of personal protective equipment in accordance with UNE-EN 170). (Spanish Standardization Association, 2020). Even more, it is recommended to use a specific electronic safety device in order to increase the safety protection by switching off the light if a human enters the room while the disinfection is in progress.

4 | CONTROL ARCHITECTURE FOR UV-C DISINFECTION

The proposed control architecture is shown in Figure 2. It is composed by the EGBNN, the movement planner, a module to estimate the UV-C, a speed controller, the pure pursuit controller, a map, and an auxiliary map. The EGBNN generates the list of way-points where the robot should go, *goals*. These way-points are tracked by the pure pursuit controller. On the other hand, the UV-C estimation module uses the radiation diagram to estimate the radiation dose will be received by the environment, it updates the map and compute the UV-C percentage that feeds the speed controller, $UV_{\%}$.

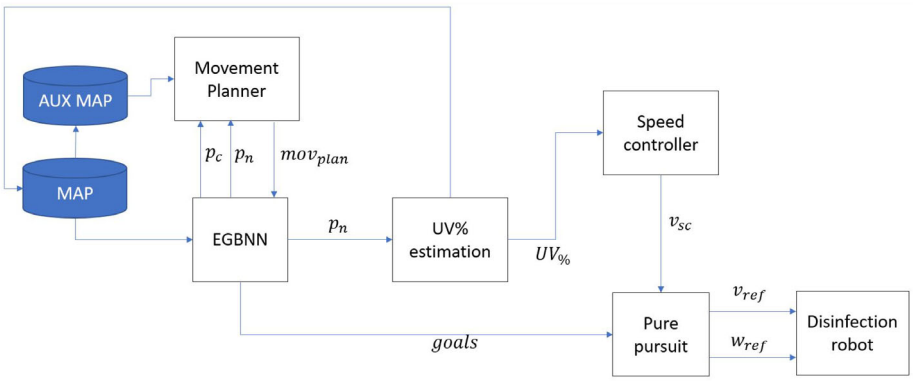


FIGURE 2 System architecture diagram.

The speed controller adjusts the speed of the robot to ensure the required UV-C dosage is administered. It generates the speed target v_{sc} that is followed by the pure pursuit controller. In turn, the pure pursuit controller governs the movement of the robot providing the references for the longitudinal speed v_{ref}, and angular speed w_{ref}.

The EGBNN uses the neural activity to determine the different weights between each of the neuronal connections. These weights are crucial because they influence the route the robot will follow. In addition, the orientation of the robot is also considered. However the neural activity is not the only information considered to decide the next cell. A movement planner has been included in the architecture to improve the performance of the algorithm.

The EGBNN, considering the current position p_c and the surrounding neural activity, estimates the next position p_n. This information is sent to the movement planner which studies the obstacles in the environment and the destination, and it provides a sequence of future movements to avoid the obstacles in an organized way, mov_{plan}.

To improve the performance of the algorithm, the movement planner does not receive the map but the auxiliary map. This auxiliary map contains the obstacles in the map and the cells previously disinfected, the latter are also considered as obstacles by the movement planner. This approach helps the algorithm to avoid cells previously disinfected.

4.1 | Disinfection approximation

Different microorganisms can be deactivated by adjusting the radiation dose. We have tested the approach with SARS-COV-2 as an example, due to its worldwide relevance. Experiments have found that the required dose to make inactive the SARS-CoV-2 virus is between 108 and 250J/m^2 (Biasin et al., 2021; Derraik et al., 2020; Sabino et al., 2020; Sellera et al., 2021). In this work the applied dose is 500J/m^2 to ensure a higher reduction. In addition, we successfully tested this UV-C lamp to disinfect a hospital room in a previous work (Sierra Garcia et al., 2021).

A circular radiation profile has been determined following the inverse square law previously described in (2). The radiation emitted by each lamp in this scenario is approximately $I_{UV} = 550\mu W/cm^2$ at a distance of 1 m. The retention time varies with the power of the UV-C lamp used; in this case, the retention time necessary is around 90 s. However, if the power of the lamp were smaller, a larger retention time would be needed.

It must be noted that the area under the robot does not receive radiation, so its disinfection is not considered. Anything between the light source and the target will block UV-C rays. Similarly, there may be specks of dust that cause shadows, which may mean that not all microorganisms are receiving the expected dose. However, these effects are considered out of the scope of the proposed model.

The UV-C radiation percentage estimation is obtained during the execution of the algorithm. A grid map is created to store the dosage at each cell while the robot moves. At the end of the algorithm, it is possible to know what cells receive the highest dose. This information is used in the speed controller to ensure that the final dose is within the appropriate range.

The disinfection process is simulated using the cell matrix with the proposed UV-C distribution and the real characteristics of the disinfection lamp. Each cell is 0.1×0.1 m, to have better precision for posterior analysis. It should be noted that this approach has certain limitations:

- The proposed radiation profile has no distance limitation.
- The existence of obstacles is not considered. Therefore, according to this model, the radiation goes through the existing walls in the map.
- The absorption coefficients of each material are not considered.

Figure 3 shows the dose received by using the proposed UV distribution model and the specifications of the disinfection lamp.

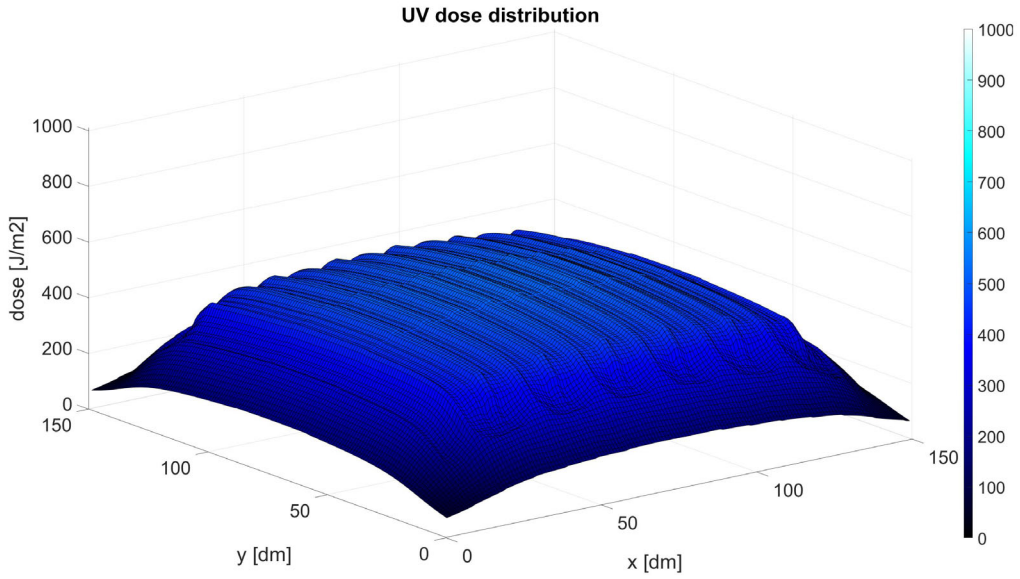


FIGURE 3 Dose received in an empty map, at the end of the disinfection process, with the UV distribution model proposed and lamp power $I_{UV} = 550\mu W/cm^2$.

4.2 | Speed and pure pursuit controller

There are two robot controllers, as shown in Figure 2. On the one hand, a path-tracking controller (called pure pursuit controller), and a speed controller that calculates the linear velocity in order to constantly follow a point in front of the robot.

Based on the principles of path tracking algorithms, the Pure Pursuit technique calculates the correct angular velocity for tracking, allowing a robotic agent to move from its current position to a predefined point ahead of the robot. On the other hand, the linear speed is governed by the speed control mechanism, which makes it easy to continuously search for the designated point in front of the robot.

The Pure Pursuit approach involves a geometry-based methodology to determine the necessary arch that will guide the vehicle towards a chosen reference point. In particular, this way-point is spatially located at a distance in front of the current position of the vehicle. The process consists of establishing an arc that connects the current position of the vehicle and the destination point; the arc length defines the look ahead distance and serves as the third constraint, along with the origin and destination points, creating a single path (Coulter, 1992).

Way-points are defined by coordinates $[x\ y\ \theta]$, which are used to obtain the robot velocity. The so called anticipation distance measures the distance from the current location at which the angular velocity is to be calculated. A small anticipation distance will cause the robot to move quickly towards the next point of the path, but the robot may surpasses that way-point causing oscillations. On the other hand, a large anticipation distance might result in larger curvature near the corners. In principle there is no theoretical limitation regarding the number of points. The limits are only determined by the memory of the hardware used to run the algorithm.

The information provided by the proposed coverage algorithm is used to estimate the speed of the robot at each section of a cell. Based on the UV-C percentage matrix, the lamp model, and the dose required for disinfection, a speed value is obtained using (5). The speed controller adjusts the speed of the robot to ensure the total required dosage D_{req} is given, considering the power of the lamp and the surrounding obstacles. If the robot finds many obstacles in the way, the effective power would decrease, so the controller reduces the robot speed to guarantee the required dosage by enlarging the retention time.

$$v_{sc}(x,y) = \rho \cdot UV_{\%}\,(x,y)\left(\frac{I_{UV}}{D_{req}(x,y)}\right) \tag{5}$$

where v_{sc} is the reference for the linear velocity of the robot in m/s; ρ is a positive parameter; $UV_{\%}$ is the estimated maximum percentage of the dose received, considering the position of the robot and its eight neighbouring cells; I_{UV} is the lamp radiation power in mW/cm^2; and D_{req} is the required radiation dose in mJ/cm^2. This dose is a function that depends on the location and can be set to different values to deal with other agents and different textures.

The I_{UV} value is based on the specifications of the UV-C distribution model. Specifically, a value of 0.55 mJ/cm^2 for the lamp is used, given by the tests performed in (Fischer et al., 2020).

4.3 | Enhanced GBNN CCPP

The GBNN algorithm has been modified to include the UV-C distribution information and the interaction with the movement planner, to make it more efficient. In this section we present the implementation of the enhanced CCPP strategy. Detailed information on the GBNN can be found in (Luo & Yang, 2008; Sun et al., 2018). The variables that define the problem are listed in Table 1.

Keeping in mind that the main task of the robot is to guarantee the disinfection of all regions, the path for complete coverage is built dynamically. Regarding the GBNN, a grid is drawn in the space to be covered, forming cells. A series of flags $f(k,l)$ are defined for each neural position (k,l). The value of the flag indicates the current state of the cell according to whether it has been visited or not, regardless the disinfection status of it. The value 0 means non-visited, 1 indicates visited, 2 means that it is an obstacle, and 3 refers to a deadlock.

It is assumed that the positions of the obstacles are previously known by the robot. To initialize the algorithm, the flag state of each neuron is initialized to 0 if the cell is empty, and to 2 if there is an obstacle. Similarly, the external input of each neuron is updated with a value of 100 or -100, and the neuronal activity with 1 or -1, following the previous criteria. This way, negative values are associated to cells with obstacles, rewarding movement towards areas of high neuronal activity.

The coverage phase is described in Algorithm 1. The external inputs I_i are the source of information about the environment. These external inputs I_i are defined as:

- $I_i = +E$, if the cell is not visited.
- $I_i = 0$, if the cell is visited.
- $I_i = -E$, if the cell is an obstacle.

The time evolution of the neural activity is given by (6), defined in (Sun et al., 2018).

$$x_i(t) = f\left(\sum_{j=1}^{k} w_{ij} \cdot max\left[x_j(t-1), 0\right] + I_i\right) \tag{6}$$

where $x_i(t)$ is the activity of the i-neuron at t; $x_j(t-1)$ is the activity of the neighbour neuron j at $t-1$, which is laterally connected with the i-neuron, and the index k corresponds to the connections between the neuron i and its neighbouring neurons.

The function $f(a) : \mathbb{R} \to \mathbb{R}$ is defined in (7), being $max[a,0] : \mathbb{R} \to \mathbb{R}$ the biggest value between a and 0.

$$f(a) = \begin{cases} -1, & a < 0 \\ \beta a, & 0 \le a < 1,\ \beta > 0 \\ 1, & a \ge 1 \end{cases} \tag{7}$$

TABLE 1 Variables and parameters of the disinfection problem.

Variable	Definition
(k_c, l_c)	Current position
(k_p, l_p)	Previous position
(k_n, l_n)	Next position
N_x	Map size on the x-axis
N_y	Map size on the y-axis
$\mathcal{N}(k,l)$	Set of the discretized workspace $\{(k,l), 1 \le k \le N_x, 1 \le l \le N_y\}$
$\mathcal{N}_r(k,l)$	Neighbouring neurons $\{\mathcal{N}(m,n) \mid m \in (k-1,k,k+1),\ n \in (l-1,l,l+1)\ , (m,n) \neq (k,l)\}$
$\mathcal{N}_{rnd}(k,l)$	Non-diagonal Neighbouring neurons $\{\mathcal{N}(m,n) \mid (m,n) \in \{(k-1,l),(k+1,l),(k,l-1),(k,l+1)\}\}$
$x(k,l)$	Neural activity at position (k,l)
$\Delta\theta(k,l)$	Angular difference between the central cell and the cell at position (k,l)
$I(k,l)$	External input to neuron $\mathcal{N}(k,l)$
$f(k,l)$	State flag in neuron $\mathcal{N}(k,l)$
r_0	Receptive field radius of a 2D neural network
E	Positive constant for the external inputs
c	Positive constant for the path selection strategy
$UV_{\%}(k,l)$	Approximate disinfection percentage in neuron $\mathcal{N}(k,l)$

ALGORITHM 1 Enhanced GBNN Algorithm

1. Compute neural activity by Equation (6).
2. $\mathcal{N}_r(k_c,l_c) \leftarrow \{\mathcal{N}(m,n) \mid m \in (k_c-1,k_c,k_c+1) \wedge n \in (l_c-1,l_c,l_c+1) \wedge (m,n) \neq (k_c,l_c)\}$
   $\mathcal{N}_{rnd}(k_c,l_c) \leftarrow \{\mathcal{N}(m,n) \mid (m,n) \in \{(k_c-1,l_c),(k_c+1,l_c),(k_c,l_c-1),(k_c,l_c+1)\}\}$
3. **if** $mov_{plan} == \emptyset$ **then**
   if *deadlock* $==$ *true* **then**
   $(k_n,l_n) \leftarrow \underset{(m,n) \in \mathcal{N}_r(p_c)}{\operatorname{argmax}} \left\{x(m,n) + c\left(1 - \frac{\Delta\theta(m,n)}{\pi}\right)\right\}$
   else
   $(k_n,l_n) \leftarrow \underset{(m,n) \in \mathcal{N}_{rnd}(p_c)}{\operatorname{argmax}} \left\{x(m,n) + c\left(1 - \frac{\Delta\theta(m,n)}{\pi}\right)\right\}$
   end-if
   $mov_{plan} \leftarrow checkMovPlanner(k_c,l_c,k_n,l_n)$
   end-if
4. **if** $mov_{plan} != \emptyset$ **then**
   $\left[(k_n,l_n),mov_{plan}\right] \leftarrow execMovPlanner\left(k_c,l_c,mov_{plan}\right)$
   end-if
5. $k_p \leftarrow k_c$; $l_p \leftarrow l_c$
   $k_c \leftarrow k_n$; $l_c \leftarrow l_n$.
6. **if** $\exists(k,l), \forall g \in \{k-1,k,k+1\} \wedge \forall h \in \{l-1,l,l+1\}$,
   s.t $f(g,h) = 1$
   then
   $l(k,l) \leftarrow 0$.
   end-if
7. $f(k_p,l_p) \leftarrow 1$.
   Update $UV_{\%}$ matrix
8. **if** $\exists(k,l), \forall m \in \{k-1,k,k+1\} \wedge n \in \{l-1,l,l+1\}$,
   $m \neq k, n \neq l$, $1 \le m \le N_x, 1 \le n \le N_y$, s.t.
   $1 \le f(m,n) \le 2$, **or**
   $x(m,n) < x(k,l)$
   then
   $f(k,l) \leftarrow 3$.
   $deadlock \leftarrow true$
   else
   $deadlock \leftarrow false$
   end-if
9. Go to 1 (Until disinfection is complete)

The global propagation of the neural activity is only produced by positive neural activities, as they are the only ones that can affect other neurons. On the other hand, the negative neural activities only have a local effect. Regarding the calculation of the weight connection w_{ij}, the further the neighbouring neuron is from the central neuron, the smaller the influence it will have on it. The weight w_{ij} of neurons i and j is defined in (8), where $|q_i - q_j|$ is the Euclidean distance between vectors q_i and q_j in the state space and α is a positive constant.

$$w_{ij} = \begin{cases} e^{-\alpha|q_i-q_j|^2}, & 0 < |q_i - q_j| \le r_0 \\ 0, & |q_i - q_j| > r_0 \end{cases} \tag{8}$$

Accordingly, for non-visited regions, the external excitation input ($I_i = +E$), where $\sum_{j=1}^{k} w_{ij} \cdot max\left[x_j(t-1),0\right] + E \gg 1$, makes the neural activity to be 1. On the contrary, for regions with obstacles, the external inhibitory input ($I_i = -E$), where $\sum_{j=1}^{k} w_{ij} \cdot max\left[x_j(t-1),0\right] - E < 0$, guarantees that the neural activity is -1. When the neuron is covered, its external input I_i is equal to 0.

As the algorithm selects the cells with largest neural activity, non-visited regions attract the robot, and it moves to non-visited areas until all regions have been visited. In each algorithm iteration the next cell is computed from the neural activity or from the sequence of movements

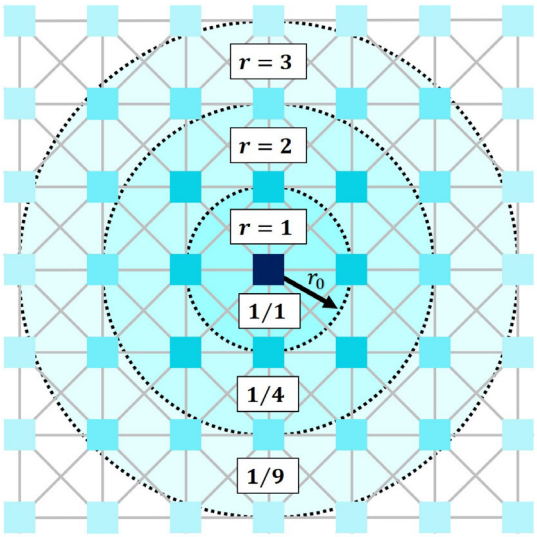


FIGURE 4 2-D neural network architecture.

provided by the movement planner. If in a matrix of 3 × 3 cells all its cells have been visited then the I of the center neural in the matrix is reset. After this check, the $UV_{\%}$ matrix and the map are updated and the existence of deadlocks is evaluated.

To execute the algorithm a 2D Cartesian space discretized into square or rectangular cells is used (Figure 4). The decay factor of the radiation intensity as a function of distance from the source is also shown.

In this case, according to the specifications of the model used to approximate the UV distribution, the cells have a square shape of 1×1 m. This cell size has been considered because the computational complexity grows with the number of cells in the map (Luo & Yang, 2008).

In a map, there are $M = N_x \times N_y$ neurons. Then, if $N_x = N$ and $N_y = N$, the map has N^2 neurons. All neurons are connected to 8 neurons. Thus, there are $8 \times N^2$ neuron connections. Therefore, the algorithm complexity is $O\left(N^2\right)$.

5 | INTERACTION BETWEEN THE GBNN AND THE MOVEMENT PLANNER

The coverage task should perform the shortest path and turn as little as possible to prevent excessive energy consumption. Thereby, the next position denoted by p_n is calculated from the current position, p_c, by (9).

$$p_n = \underset{(m,n) \in \mathcal{N}_r(p_c)}{\operatorname{argmax}} \left\{ x(m,n) + c\left(1 - \frac{\Delta\theta(m,n)}{\pi}\right) \right\} \tag{9}$$

Where $x(m,n)$ is the neural activity of neuron in the cell (m,n), c is a positive constant, $\mathcal{N}_r(p_c)$ denotes the neighbourhood of the current position p_c, and the term $\Delta\theta(m,n) \in [0, 2\pi]$ corresponds to the angle of rotation between the current and the possible next moving direction to the cell (m,n). In this work a value $c = [0,1]$ is considered.

The procedure to compute $\Delta\theta$ is shown in Figure 5 and formalized by (10), where *atan*2 denotes the four-quadrant inverse tangent function.

$$\Delta\theta(p_n) = \Delta\theta_n = \theta_c - \theta_n = \left| atan2\left(y_{p_c} - y_{p_p}, x_{p_c} - x_{p_p}\right) - atan2\left(y_{p_n} - y_{p_c}, x_{p_n} - x_{p_c}\right) \right| \tag{10}$$

It should be noted that diagonal movements are disabled to achieve an organized path according to the selected motion strategy. Their use is only allowed for getting out from deadlock situations. When performing diagonal movements the robot tends to leave spaces without disinfecting. Then, the robot should return to these areas with the consequent waste of time and energy that would suppose. Thus, these movements are disabled for standard movements. This is formalized in the step 3 of the Algorithm 1, if the deadlock is false the diagonal cells are not included in the neighbourhood set $\mathcal{N}_{rnd}$.

Once the next cell is computed, the movement is not directly executed but the GBNN requests to the movement planner whether there exists a prerecorded optimum movement sequence considering the state of the map. This is formalized in the Algorithm 1 in the step 3 by

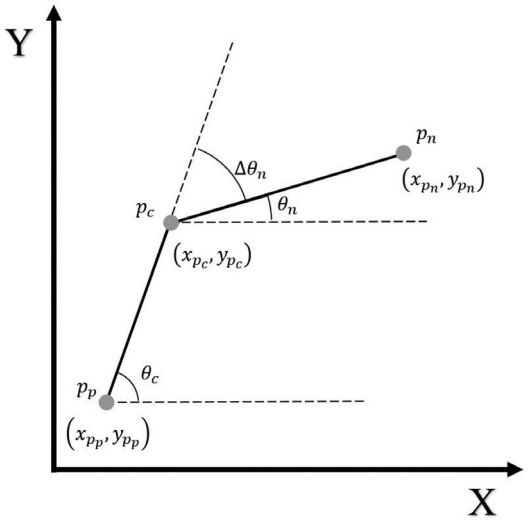


FIGURE 5 Path selection strategy - next position P_n calculation.

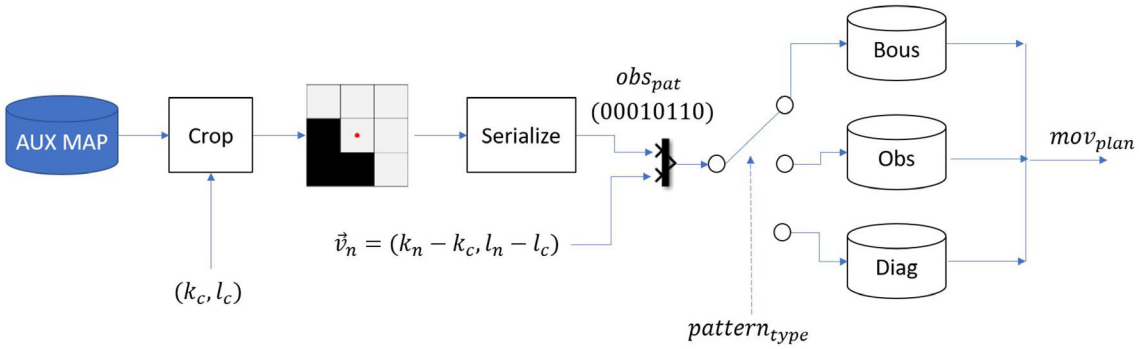


FIGURE 6 Movement planner architecture.

$mov_{plan} \leftarrow checkMovPlanner(k_c, l_c, k_n, l_n)$. This prerecorded movement sequence mov_{plan} is obtained by the movement planner whose architecture is shown in Figure 6.

First from the auxiliary map the neighbourhood cells to the current are extracted. This 9×9 matrix is binarized and serialized. If there is an obstacle in a cell it receives an 1 and 0 in other case. After the binarization, the content of the matrix is serialized from top to bottom and from left to right, without considering the central cell to obtain a byte which univocally identifies the obstacle pattern in the environment. In the Figure 6 an example of the serialization can be observed.

The obstacle pattern obs_{pat} together with the vector to the next cell $v_n = (k_n - k_c, l_n - l_c)$ are used as index to access to 3 different lookup tables which stores the optimum sequences of movements. Only one lookup table is accessed at the same time, this is managed by the signal $pattern_{type}$.

To follow the motion technique used, be it spiral or boustrophedon, the overall coverage trend is considered to move from top to bottom and from left to right, thus optimizing the full coverage path, reducing repetition and clutter. When selecting the next point to move, it is important to consider the need to perform specific movements on certain occasions. These events are:

- The need for specific turns when following the boustrophedon strategy;
- the need to avoid corners without making a diagonal movement during a deadlock situation;
- the need to get around obstacles in a certain way.

For each one of these 3 event types, the relation between the obstacle pattern, the vector to the next cell, and the optimum sequence of movements, $\left(obs_{pat}, \vec{v}_n\right) \rightarrow mov_{plan}$, is stored in a different lookup table: *Bous* denotes the table for the boustrophedon motion, *Obs* denotes the

table to avoid obstacles, and *Diag* refers to the table to perform diagonal movements. First the *Obs* table is checked, if there is not match then the *Bous* table only if the motion type is boustrophedon, and finally the *Diag* table only if there is a deadlock.

If the movement planner provides a sequence of movement, then this sequence is executed in the step 4 Algorithm 1. In each iteration of algorithm in the step 4 a movement is extracted from the sequence. This is formalized by $[(k_n, l_n), mov_{plan}] \leftarrow execMovPlanner(k_c, l_c, mov_{plan})$. When the sequence is empty $mov_{plan} == \emptyset$ and the step 3 is again executed.

The obstacle patterns can be seen as a series of templates used to check the environment. These templates are used to decide the sequence of movements necessary to continue. Figure 7.a shows one of the many possible cases where the robot has to abandon the general movement trend and go around an obstacle.

Something similar happens when generating the boustrophedon movement since it is not a turn that occurs naturally using neural network behaviour. This situation can be observed in Figure 7b.

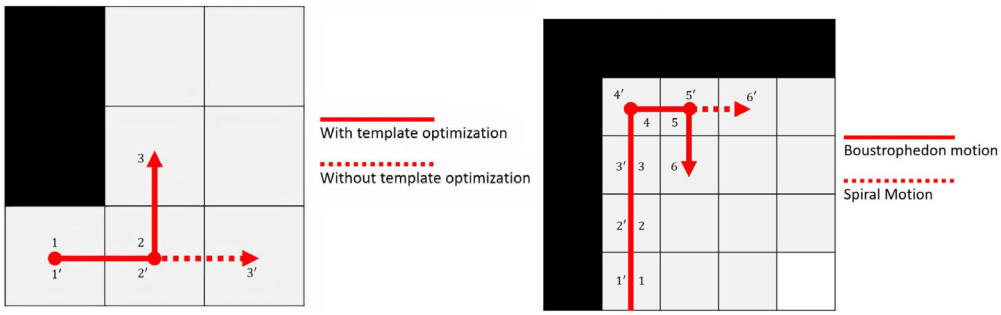
In the case of boustrophedon motion, the robot has to follow a specific motion pattern. To do it, different motion templates have been designed as shown in Figure 8. They are stored in the *Bous* table.

To facilitate following a path efficiently in environments with different obstacle patterns, obstacle templates are used. These templates force the robot to adopt a specific path without considering the neural network behaviour. In Figure 9 the most common situations that have occurred in the maps used for testing are shown. They are stored in the *Obs* table.

In cases where these diagonal movements are needed and there are obstacles, considering the size of the robot, an L-shaped movement is executed to go around the obstacle instead of the diagonal movement. In Figure 10, the most common cases appear. They are stored in the *Diag* table.

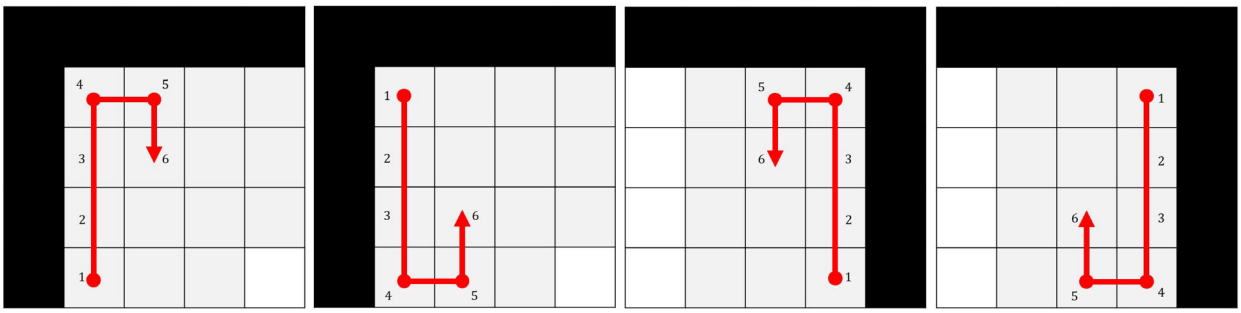
These templates force the robot to adopt a specific path without considering the neural network behaviour during the GBNN algorithm process. The coverage is considered to be achieved moving top to bottom and left to right, thus optimizing the route, reducing repetition and clutter. The templates are defined as a matrix of 3×3 cells that surrounds the robot; thus, there are $2^8 = 256$ different templates. This number of templates, that can be easily handled by the algorithm, covers all possible cases, even for more complicated environments and different obstacle shapes.

Another important feature to perform the complete coverage is the auxiliary obstacle map. This map is not part of the neural network but it contributes to increase the use of templates. In this map, the cells that have been previously disinfected are stored as if they were obstacles. A delay of 3 samples is used to analyse the map. This way the previously disinfected cells are managed as if they were obstacles by the obstacle avoidance strategy. In Figure 11 the auxiliary obstacle map is shown.



(a) Example of template (b) Spiral vs boustrophedon

FIGURE 7 Movement with templates.



(a) Top-Right (b) Bottom-Right (c) Top-Left (d) Bottom-Left

FIGURE 8 Boustrophedon motion templates.

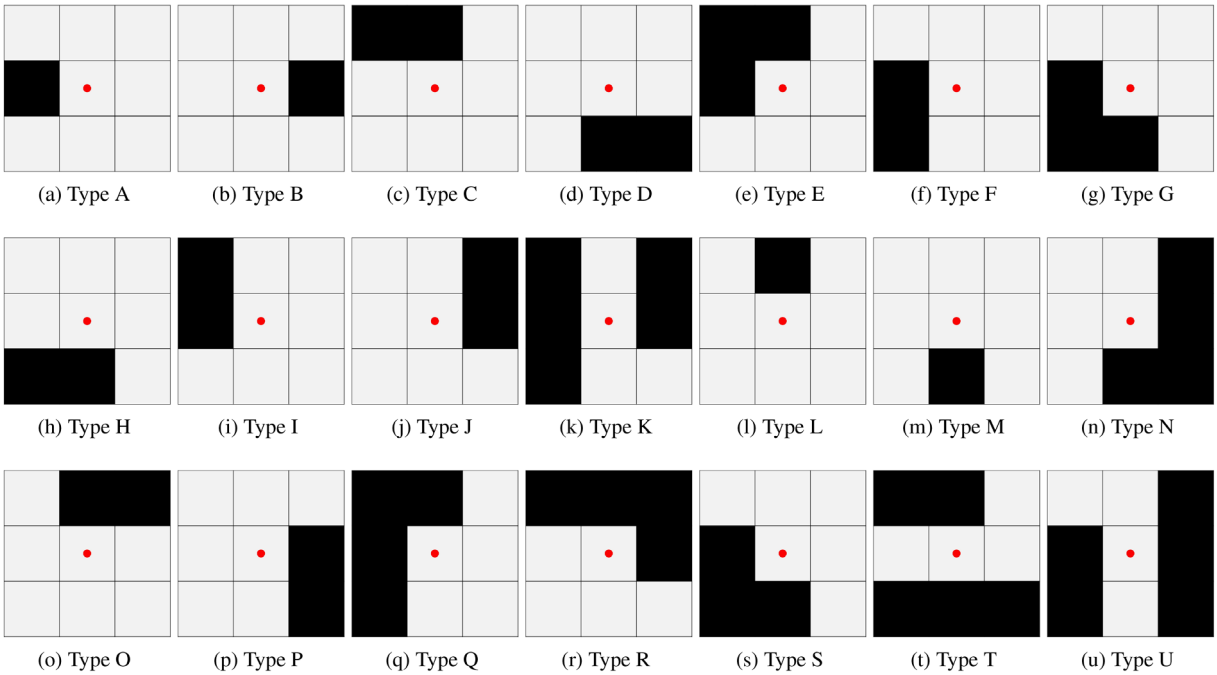


FIGURE 9 Obstacle template.

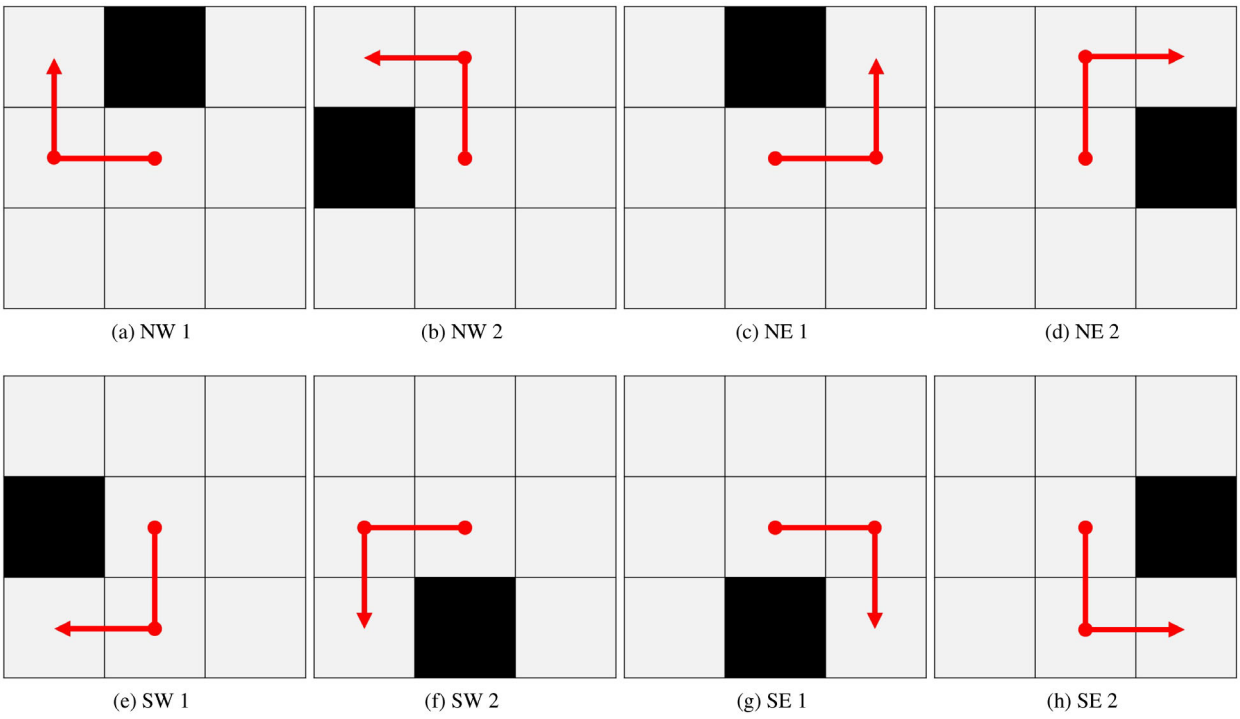
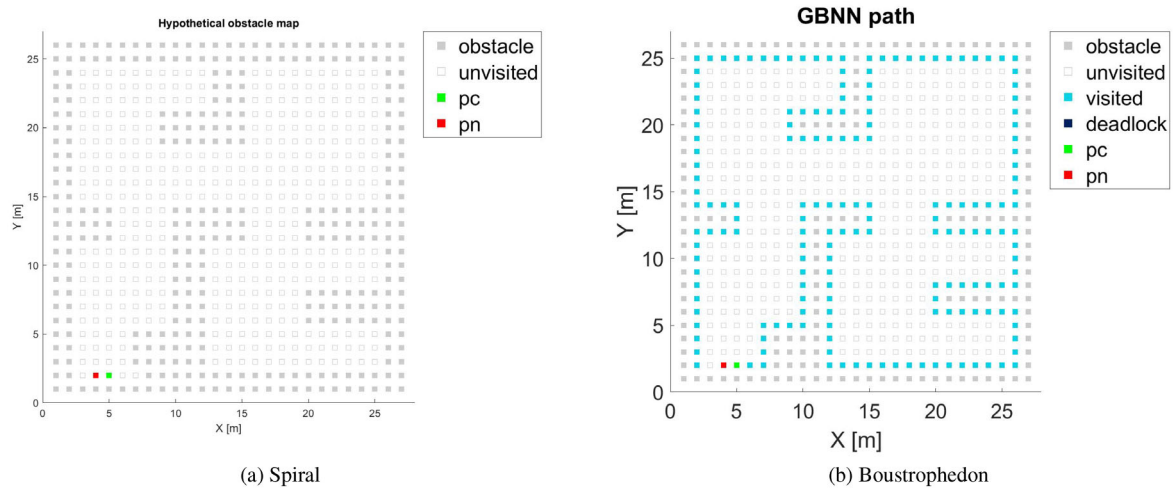


FIGURE 10 Diagonal movement templates.

6 | SIMULATION RESULTS AND DISCUSSION

In the proposed testing scenarios, the initial position is $p(2,2,\frac{\pi}{2})$, and the parameters of the model $[\alpha,\beta,\rho_0,c,E]$ are set to [2,0.6,7,0.1,100]. The speed controller uses the tuning parameter $\rho \approx 1.54$; $I_{UV} = 5.5\,W/m^2$ is for the lamp used; $D_{req} = 500\,J/m^2$ is the reference value to achieve disinfection; and $d_{LA} = 0.2$ is the anticipation distance for the pure pursuit controller. The sample time for simulation is $dt = 0.1\,ms$.



(a) Spiral (b) Boustrophedon

FIGURE 11 Updating the auxiliary obstacle map while the robot is moving [Medium complexity map, $n_{GBNN} = 172$].

The computer used for the tests is an Intel(R) Core(TM) i7-8750H CPU @ 2.20*GHz* with a 2201*MHz* clock; 1536*KB* cache; and 6 processors. The operating system is Microsoft Windows 10 Education N.

6.1 | Improvement obtained by the movement planner and the speed controller

Different metrics have been used to asses the effectiveness of the navigation algorithm of the mobile robot (Munoz-Ceballos & Suarez-Rivera, 2022). Table 2 shows the metrics used to analyse the performance algorithm, and the values of the variables once the simulation of the disinfection process has finished. A comparison is done with the GBNN algorithm and three cases: results obtained without using the movement planner (WMP), without the speed controller (WVC), or without considering any of both previous options (WMP & WVC). These results are summarized in Table 2. The ratio between travelled area versus available area is shown in the last row, called excess travel area (11). Nevertheless, the ratio scanned area versus available area has not been included because all cells are visited, so this value would be 100% in all cases.

$$ta_{ex} = \frac{Area_{Traveled} - Area_{Available}}{Area_{Available}} \times 100 \tag{11}$$

The comparative analysis confirms how the actions implemented by the Enhanced GBNN algorithm lead to an improvement in the disinfection with UV-C radiation. It can be seen how the speed controller is able to maximize the dose within the desired threshold. Besides, it is also shown how, through the use of the movement planner, a more efficient path can be obtained, avoiding unnecessary movements, as well as a better distribution of the radiation received by the environment.

It would be desirable to maximize the percentage of cells with a UV-C radiation dose in the expected range, that is, (12).

$$D_N = 0.9D_{req} < D < 1.1D_{req} \tag{12}$$

where D_N is the desired dose range and D is the final UV-C dose received. Therefore, D_H is the percentage of dose over this threshold, and D_L is the percentage of dose below it.

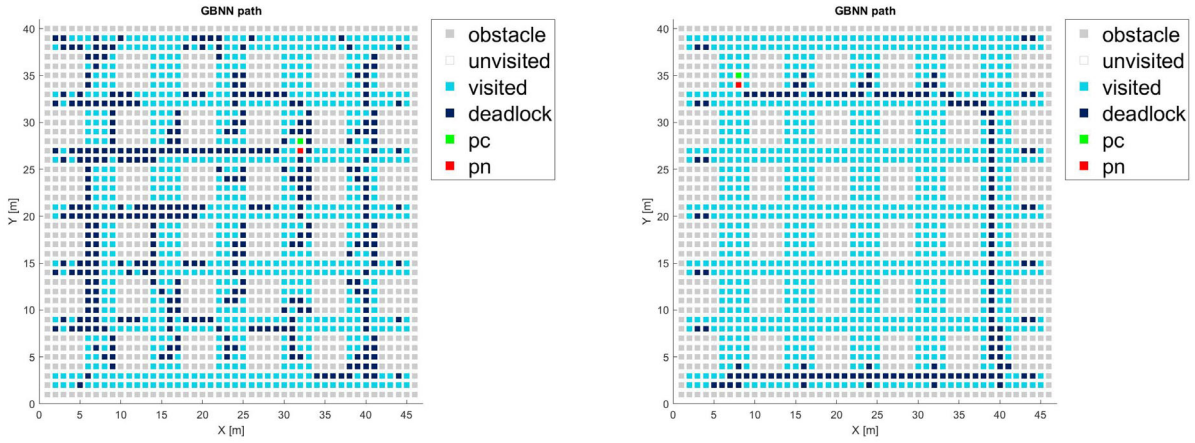
It should be noted that maximizing the areas that are within the desired range will avoid degradation of the surrounding materials and unnecessary energy consumption, ensuring a more reliable environment disinfection.

Most of the surface is found to be within the expected dosage range when using the enhanced GBNN algorithm. However, the areas closer to the walls are below the expected value. Besides, areas near blocking situations exceed this threshold, resulting in wasted energy and unnecessary deterioration of the materials. This negative effect is amplified when neither the movement planner nor the speed controller are used, as can be seen in Table 2.

Figure 12 shows how the robot manages to disinfect each of the zones by using a boustrophedon movement. Indeed, the number of deadlock situations is reduced when the enhanced algorithm is used, Figure 12b.

TABLE 2 Metrics used to analyse the disinfection process.

Metric	Definition	Enhanced GBNN	WMP	WVC	WMP & WVC
t_{robot}	Disinfection process time	02 : 17 : 47	02 : 23 : 38	02 : 14 : 14	02 : 20 : 13
n_{GBNN}	Algorithm iterations	1334	1578	1334	1578
n_{goals}	Number of goals	405	652	405	652
d_{robot}	Distance covered	1371.75 m	1708.51 m	1336.52 m	1667.98 m
r_{robot}	Radians rotated	933.93 rad	1480.35 rad	875.93 rad	1412.64 rad
v_{av}	Average speed	0.17 m/s	0.20 m/s	0.17 m/s	0.2 m/s
D_{max}	Maximum dose	1021.20 J/m^2	1242.60 J/m^2	1664.16 J/m^2	1834.13 J/m^2
D_{min}	Minimum dose	292.24 J/m^2	310.18 J/m^2	153.21 J/m^2	132.64 J/m^2
D_N	Desired dose range	73.24%	51.81%	38.39%	22.20%
D_H	Excess dose range	16.27%	40.23%	40.02%	54.30%
D_L	Shortage dose range	10.49%	7.96%	21.59%	23.50%
ta_{ex}	Excess travel area	25.16%	55.89%	21.95%	52.19%



(a) WMP/WMP & WVC (b) Enhanced GBNN/WVC

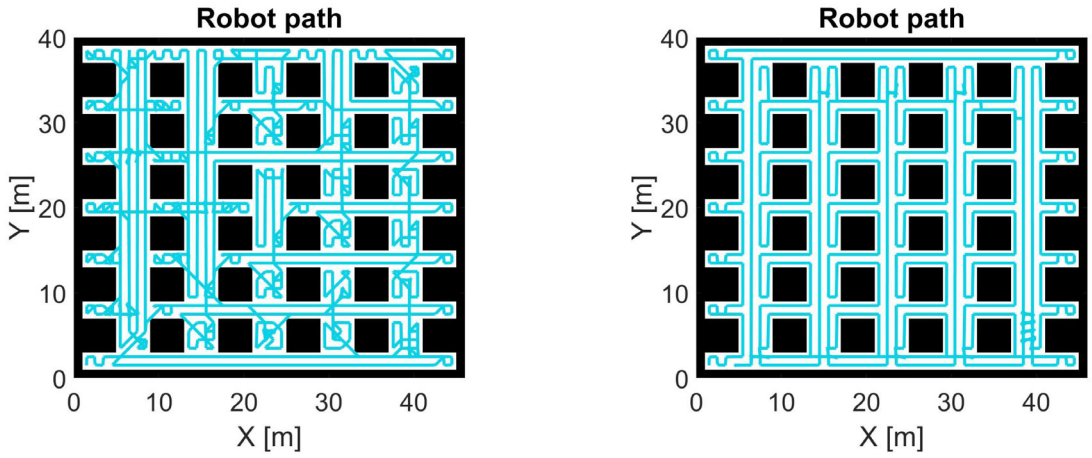
FIGURE 12 Path generated by the GBNN algorithm to guarantee the complete disinfection of the environment.

Figure 13 shows the path followed in each case to complete the disinfection process with UV-C radiation. It is possible to observe how when the enhancement is not implemented, Figure 13a, there are many more unnecessary movements. The areas where the robot finds a deadlock can be distinguished. In these conditions, the algorithm relies on the neural activity of the environment to be able to find areas to visit. These deadlock situations will produce a negative effect in terms of excess of dose received.

As said before, in Figure 14 it can be seen that there are some peaks due to the deadlocks. In these cases, the speed regulator is not able to fully reduce the radiation peaks; though, the controller maintains the received dose within the appropriate range in the rest of areas. However, it is interesting to remark that even with some remaining deadlocks, the enhanced algorithm achieves a reduction of 44% in the maximum dose. In addition, it may be noted that near the walls the received dosage is lower than elsewhere in the room.

6.2 | Evaluation with multiple radiation dosages

We have also tested the algorithm in a map with four regions with different D_{req} values each one: 400, 500, 600, and 700 J/m^2. As it is possible to observe in Figure 15, although the UV dose distribution has a similar shape to the surface of Figure 14, in this case the dosage levels are different according to the four regions in the map, each one requiring a specific dose.



(a) WMP/WMP & WVC (b) Enhanced GBNN/WVC

FIGURE 13 Path followed by the robot at the end of the disinfection process with boustrophedon motion.

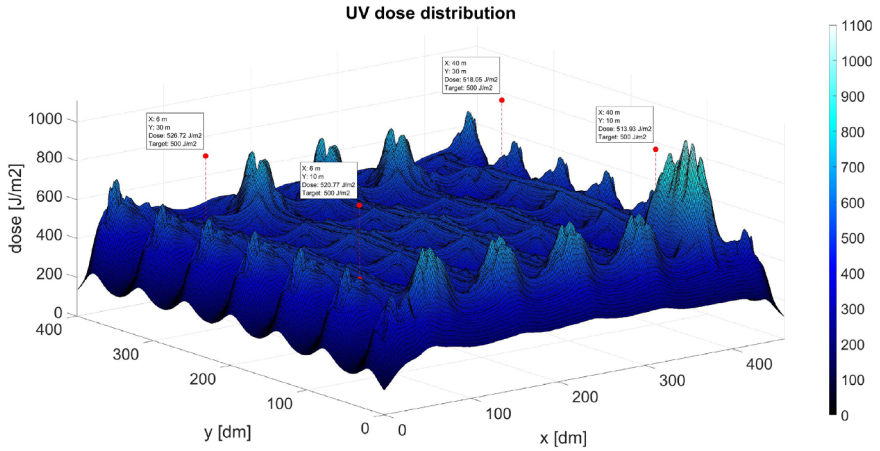


FIGURE 14 Dose received considering the UV-C distribution model described in Figure 4 and assuming a lamp power $I_{UV} = 550\,\mu W/cm^2$ using the enhanced GBNN algorithm.

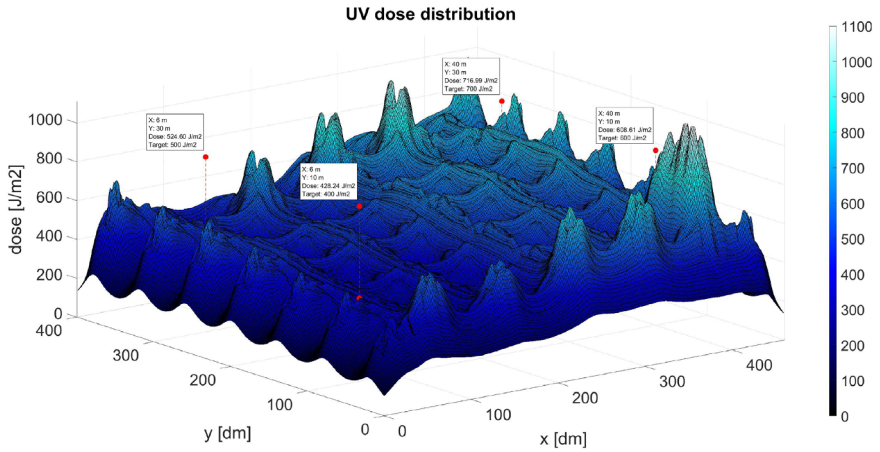


FIGURE 15 Dose received considering a map with 4 different D_{req} values: 400, 500, 600 and 700 J/m^2.

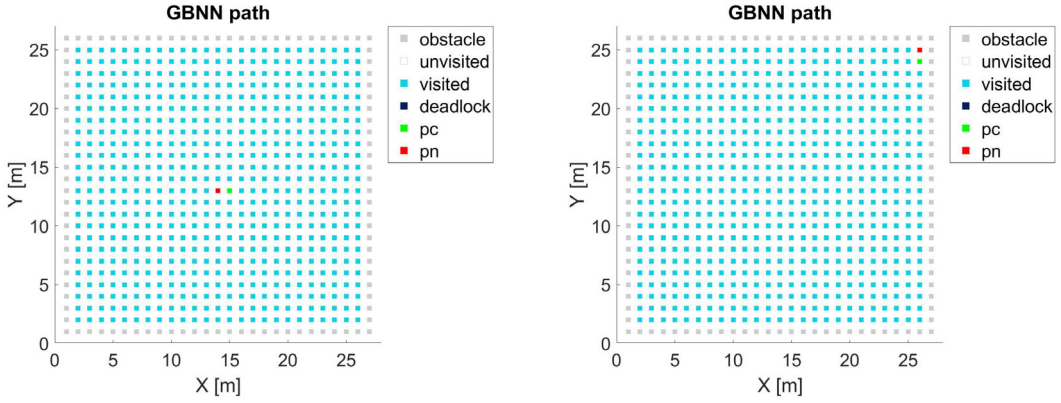
6.3 | Evaluation in different scenarios

6.3.1 | Test 1: Empty map

First, an empty map is used. This space is a working environment free of obstacles but surrounded by walls. In Figure 16, the paths involved are reasonably simple and without further complications. The robot moves from the outside to the inside using spiral motion until it reaches the centre of the map. On the other hand, with boustrophedon movements, the robot moves around the environment in a zigzag way.

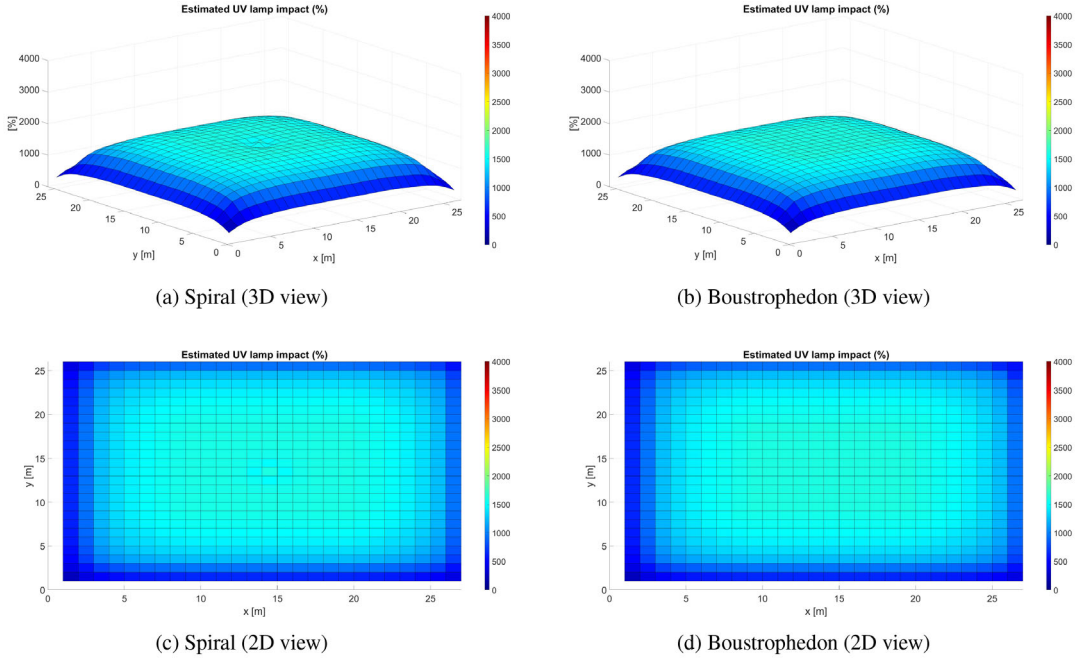
In Figure 17, it can be noted how there is a higher dose coverage overlap in the mid zone, using boustrophedon motion. However, in the case of using spiral motion, the dose received in the map centre decreases in the same way as in the edges. This phenomenon is explained because this point is the last one to be visited, and therefore the robot will have to go slower in this area. Something similar happens at the last point visited with boustrophedon motion, but it is less visually evident.

The paths followed by the robot in each situation are shown in Figure 18. They are two routes, which do not present any problems during the entire journey. It is important to highlight the different speed settings used in each case. The robot that uses spiral motion has a gradual increase as it moves closer to the central point. For the other robot, the speed presents a more uniform profile during the path.



(a) Spiral (b) Boustrophedon

FIGURE 16 GBNN path for Test 1, generated by complete coverage path planning.



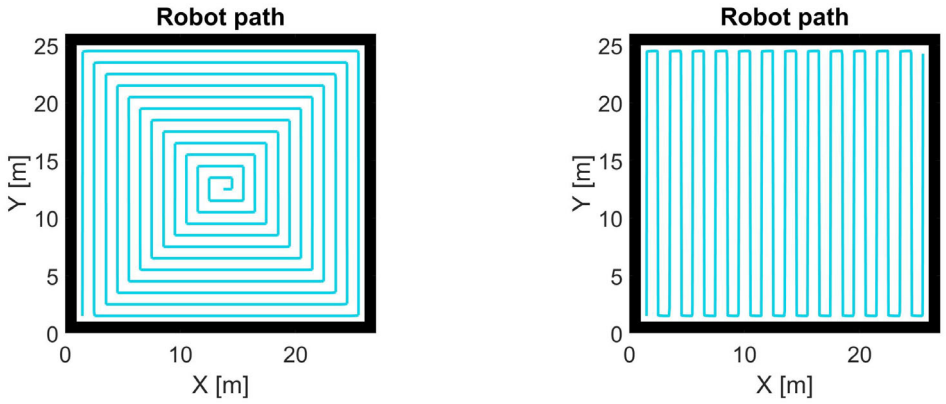
(a) Spiral (3D view) (b) Boustrophedon (3D view)

(c) Spiral (2D view) (d) Boustrophedon (2D view)

FIGURE 17 UV % estimated during the complete coverage path planning of Test 1.

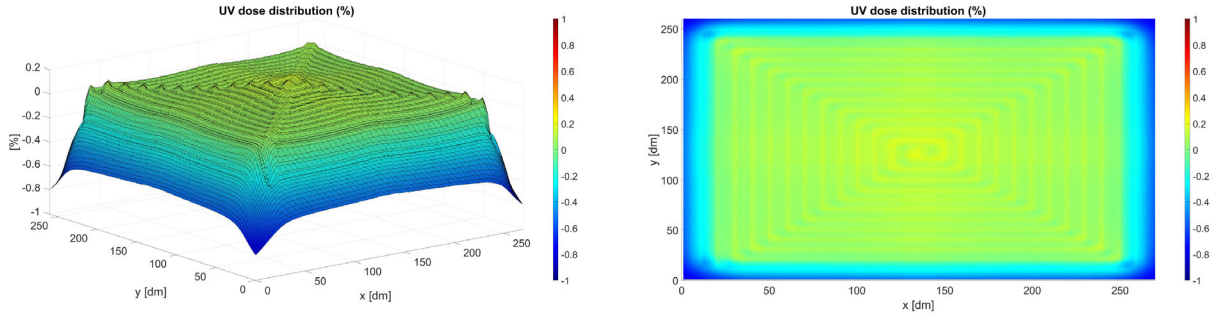
Figure 19 shows a very uniform dose distribution. It is possible to see where the robot passed since the disinfection is lower, forming a kind of lane. The disinfection is lower closer to the walls, and the peaks that appear are caused when the robot turns. Figure 20 shows a very uniform dose distribution. It is possible to see where the robot passed since the disinfection is lower, like in the previous case. The disinfection is lower, closer to the walls too.

Lastly, in Figure 21, it is confirmed that the radiation received is mostly inside the expected range. It is evident where the robot has passed because the dose received is lower. Similarly, near the walls, the desired dose is not reached.



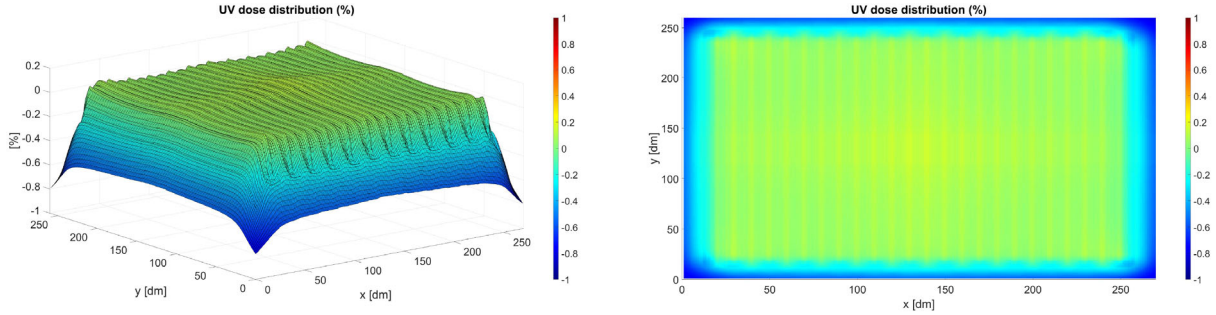
(a) Spiral (b) Boustrophedon

FIGURE 18 Robot path for Test 1.



(a) D_{UV} (3D view) (b) D_{UV} (2D view)

FIGURE 19 D_{UV} received with spiral motion for Test 1.



(a) D_{UV} (3D view) (b) D_{UV} (2D view)

FIGURE 20 D_{UV} received with boustrophedon motion for Test 1.

6.3.2 | Test 2: Low complexity map

In this workspace, the robot must access different corridors. This map makes the disinfection work harder if it is compared with the previous scenario.

In Figure 22, it can be seen that in both cases the behaviour is similar when returning to the initial corridor. However, with spiral motion it seems to be a bit more chaotic (although it only takes 4 more iterations to complete the map).

In Figure 23, it can be perceived that the path generated by the algorithm using boustrophedon motion obtains a more balanced dose distribution. The peak that appears using spiral motion should be corrected by the speed control. This overshoot of radiation is caused because the robot needs to pass again through a previously visited area. This type of problem increases as the complexity of the maps grows.

The real routes pursued by the robots are shown in Figure 24.

Figure 25 details the final dose received in the workspace once the complete coverage path planning is completed. A higher amount of the dose received appears in the area with deadlock situations. Even though the speed controller manages to reduce the final dose, there are still peaks in that area as expected in Figure 23. In the case considering boustrophedon motion (Figure 26), the dose received is more uniform, showing the typical peaks of each turn. As in the previous case, it is also worth mentioning the section where a deadlock appears whose received dose is higher than in the rest of the regions.

Lastly, in Figure 27, it can be seen that more areas exceed the threshold for the path generated by spiral motion. In both cases, there are areas below the expected threshold due to the influence of contours on disinfection. A dose drop will always be observed near obstacles.

6.3.3 | Test 3: Medium complexity map

On this occasion, considering the distribution of obstacles on the map, the need to escape deadlock situations has been more frequent, independently of the motion used. In Figure 28, it can be seen that both paths are very similar.

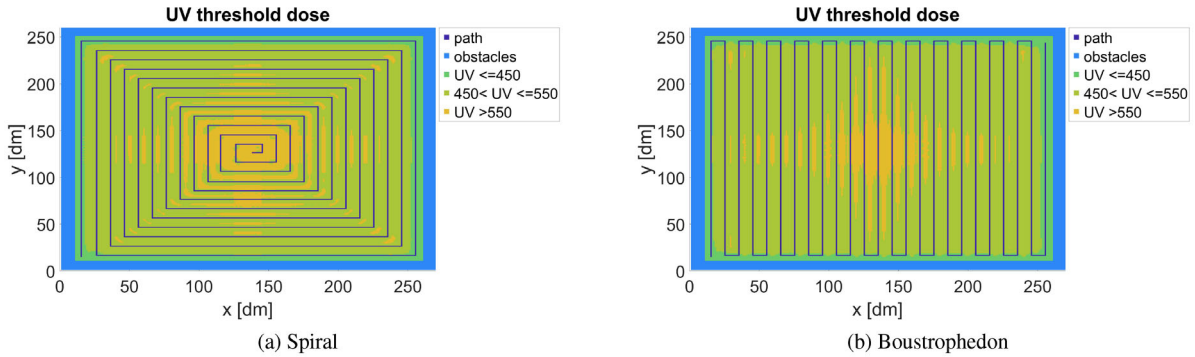


FIGURE 21 D_{UV} received categorized into thresholds for Test 1.

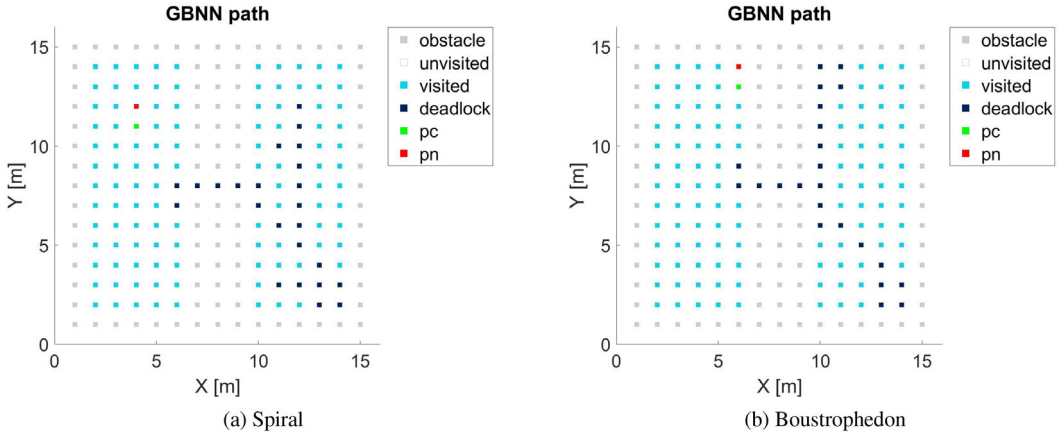
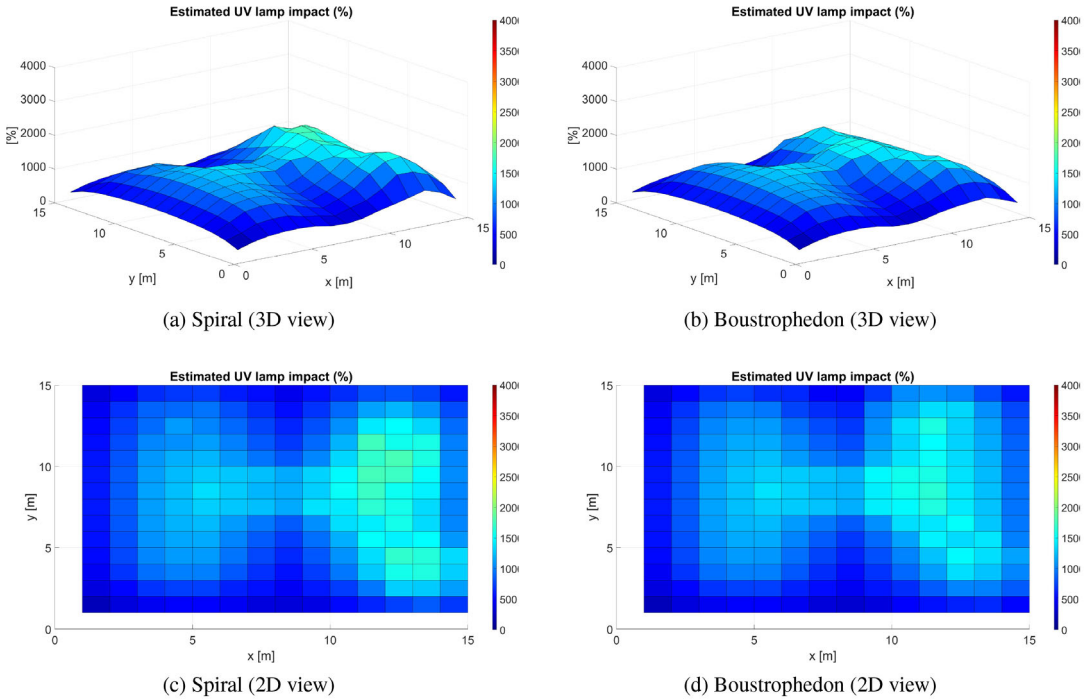


FIGURE 22 GBNN path for Test 2, generated by complete coverage path planning.



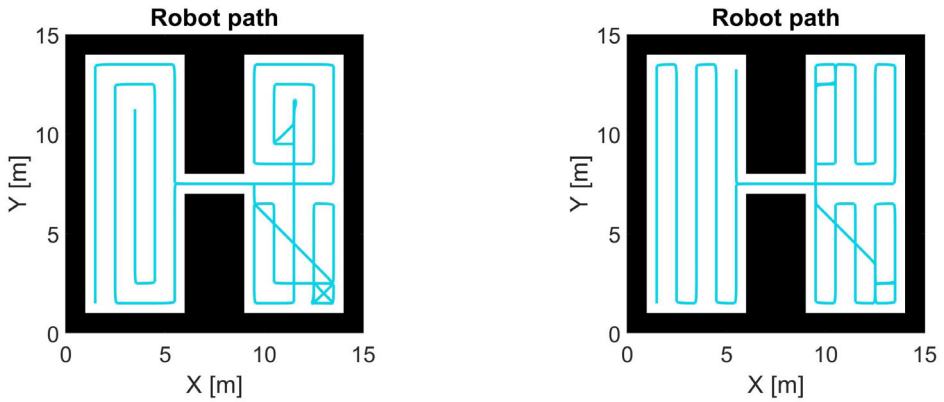
(a) Spiral (3D view)

(b) Boustrophedon (3D view)

(c) Spiral (2D view)

(d) Boustrophedon (2D view)

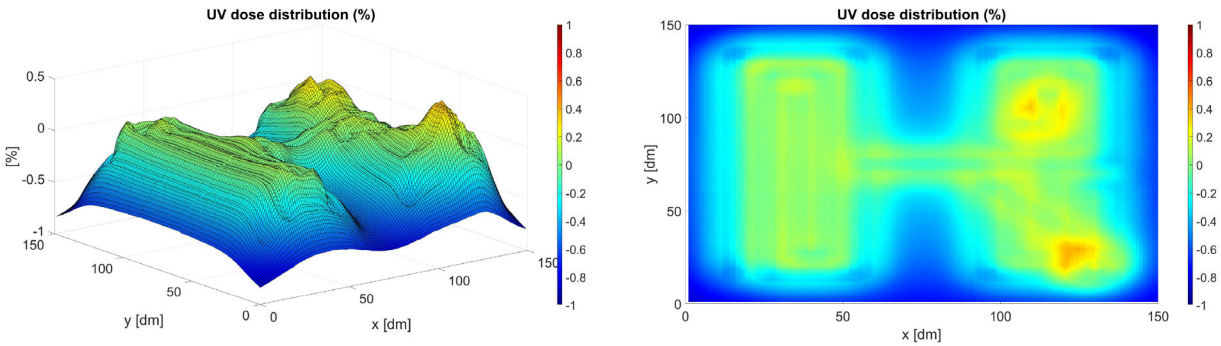
FIGURE 23 UV % estimated during the complete coverage path planning of Test 2.



(a) Spiral

(b) Boustrophedon

FIGURE 24 Robot path for Test 2.



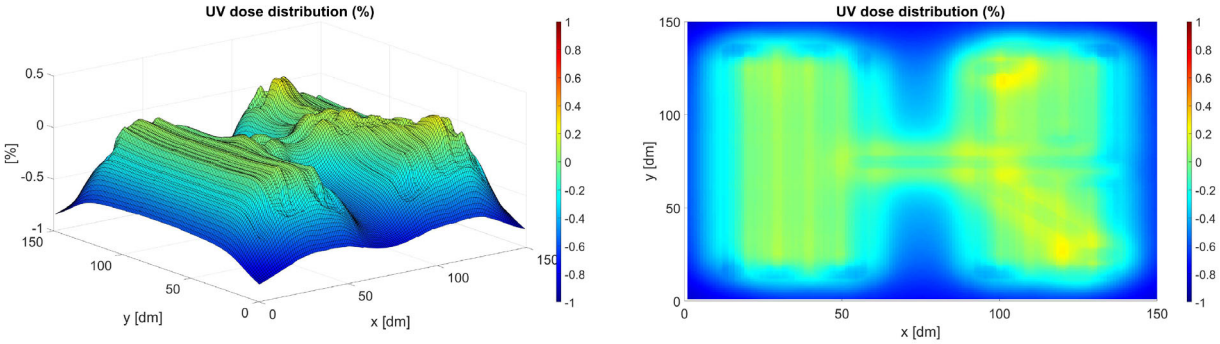
(a) D_{UV} (3D view)

(b) D_{UV} (2D view)

FIGURE 25 D_{UV} received with spiral motion for Test 2.

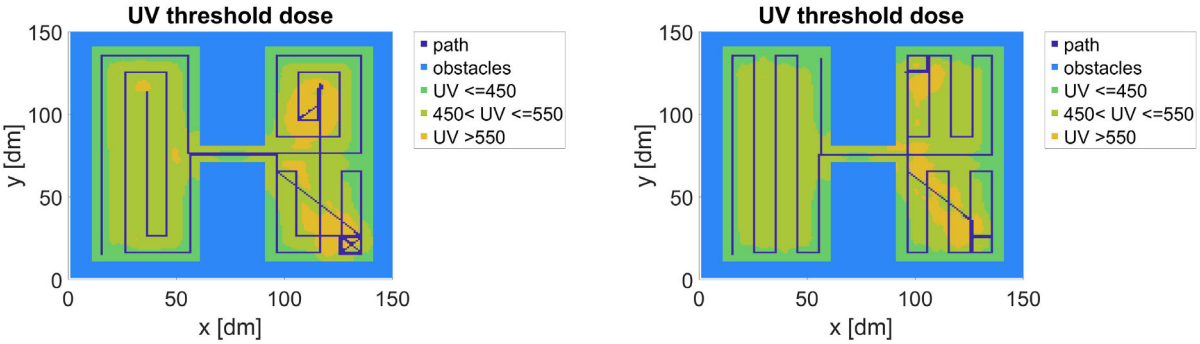
In Figure 29, when using spiral motion, minor excesses of radiation can be detected spread over the map. On the other hand, the robot that uses boustrophedon motion has a very specific area where all the excess dose is located. Although the speed controller will attempt to reduce it, in Figure 32, it can be shown that this region has the highest concentration considering the rest of the map.

The paths followed by each robot are shown in Figure 30. Both routes, each with a different motion technique, achieve to cover the whole space. However, neither succeeds in covering the entire workspace without having to pass through previously visited areas. It is remarkable the knot that originates in the upper left of the map when spiral motion is used.



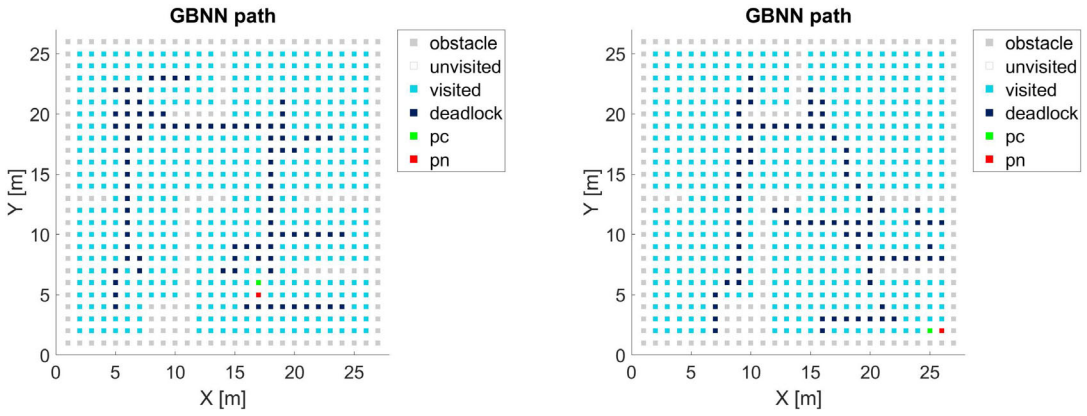
(a) D_{UV} (3D view) (b) D_{UV} (2D view)

FIGURE 26 D_{UV} received with boustrophedon motion for Test 2.



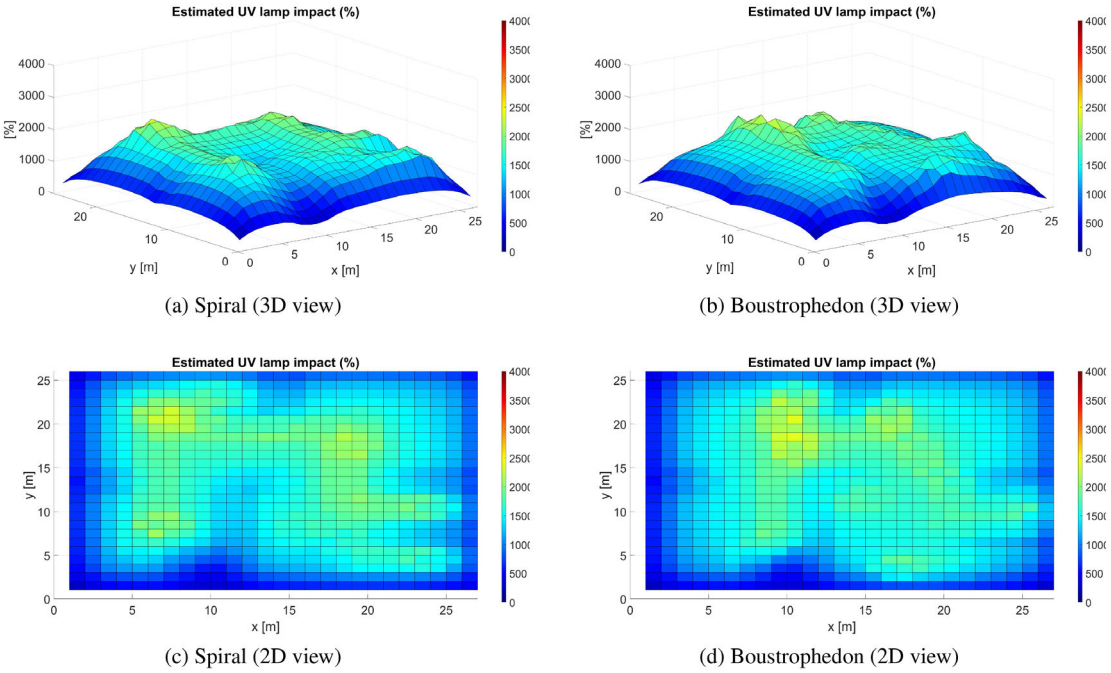
(a) Spiral (b) Boustrophedon

FIGURE 27 D_{UV} received categorized into thresholds for Test 2.



(a) Spiral (b) Boustrophedon

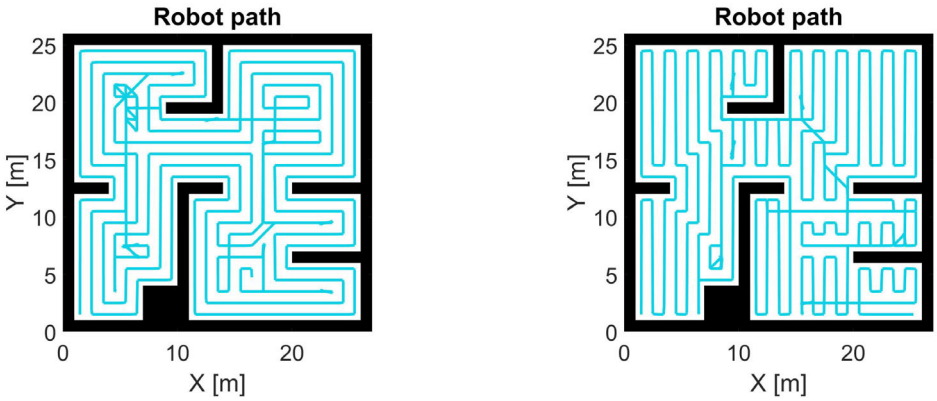
FIGURE 28 GBNN path for Test 3, generated by complete coverage path planning.



(a) Spiral (3D view) (b) Boustrophedon (3D view)

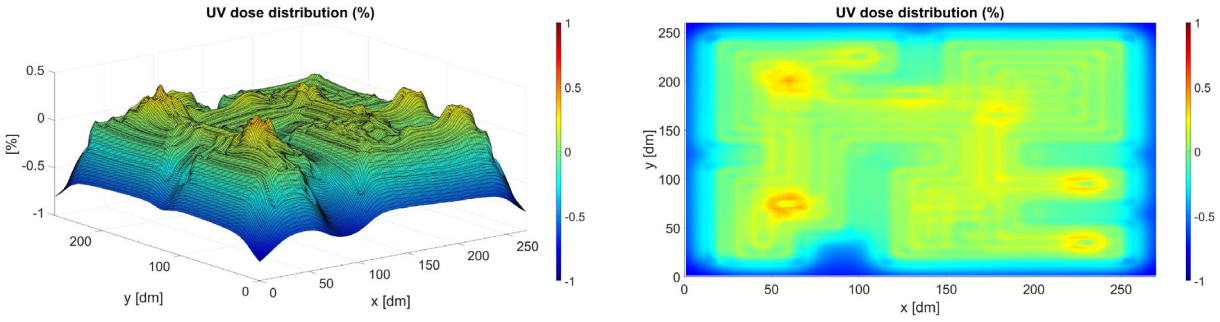
(c) Spiral (2D view) (d) Boustrophedon (2D view)

FIGURE 29 UV % estimated during the complete coverage path planning of Test 3.



(a) Spiral (b) Boustrophedon

FIGURE 30 Robot path for Test 3.



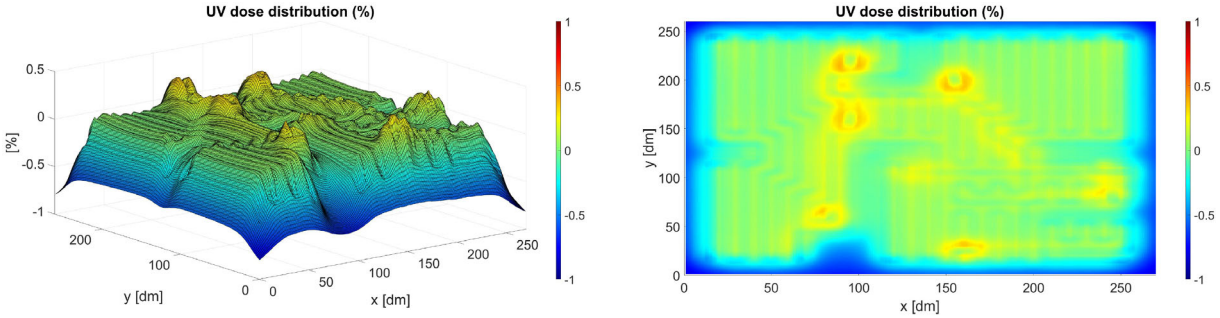
(a) D_{UV} (3D view) (b) D_{UV} (2D view)

FIGURE 31 D_{UV} received with spiral motion for Test 3.

In Figure 31, it can be appreciated that the areas which receive more radiation are those registered in Figure 29. The speed controller can smooth these regions but does not achieve a perfect accuracy.

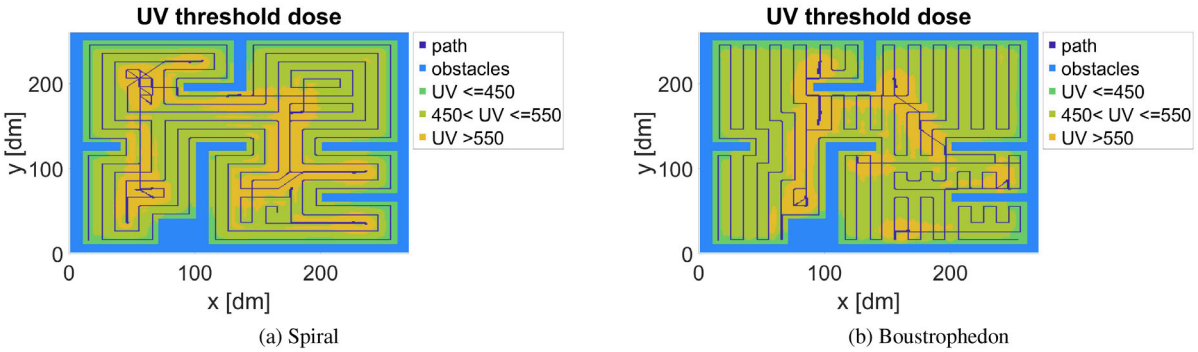
Similarly, Figure 32 shows an equivalent result. Even though the excessive amount of radiation seen in Figure 29 was reduced, the problem is the same as with spiral motion. The speed control reduces the dose received in that area, but it exceeds the desired quantity. This fact is enhanced as the map complexity grows.

Analysing the Figure 33, in both cases the expected threshold is exceeded by a similar percentage. However, with boustrophedon motion case is possible to be closer to the expected dose.



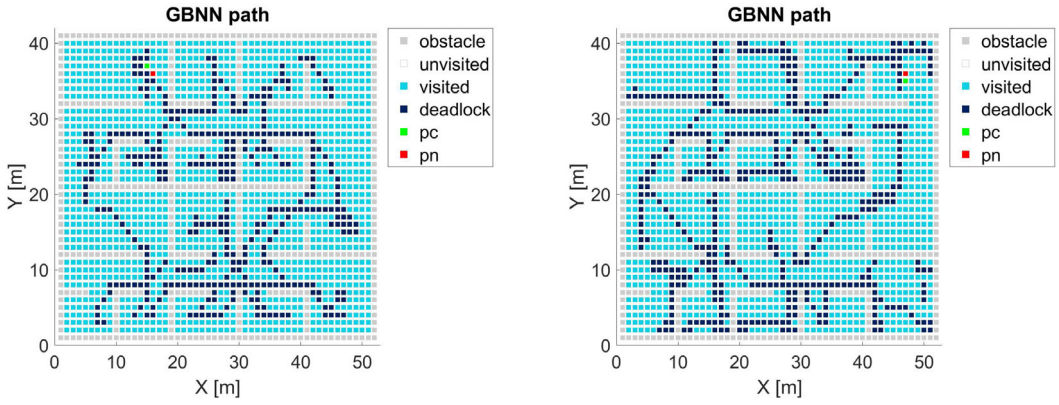
(a) D_{UV} (3D view) (b) D_{UV} (2D view)

FIGURE 32 D_{UV} received with boustrophedon motion for Test 3.



(a) Spiral (b) Boustrophedon

FIGURE 33 D_{UV} received categorized into thresholds for Test 3.



(a) Spiral (b) Boustrophedon

FIGURE 34 GBNN path for Test 4, generated by complete coverage path planning.

6.3.4 | Test 4: High complexity map

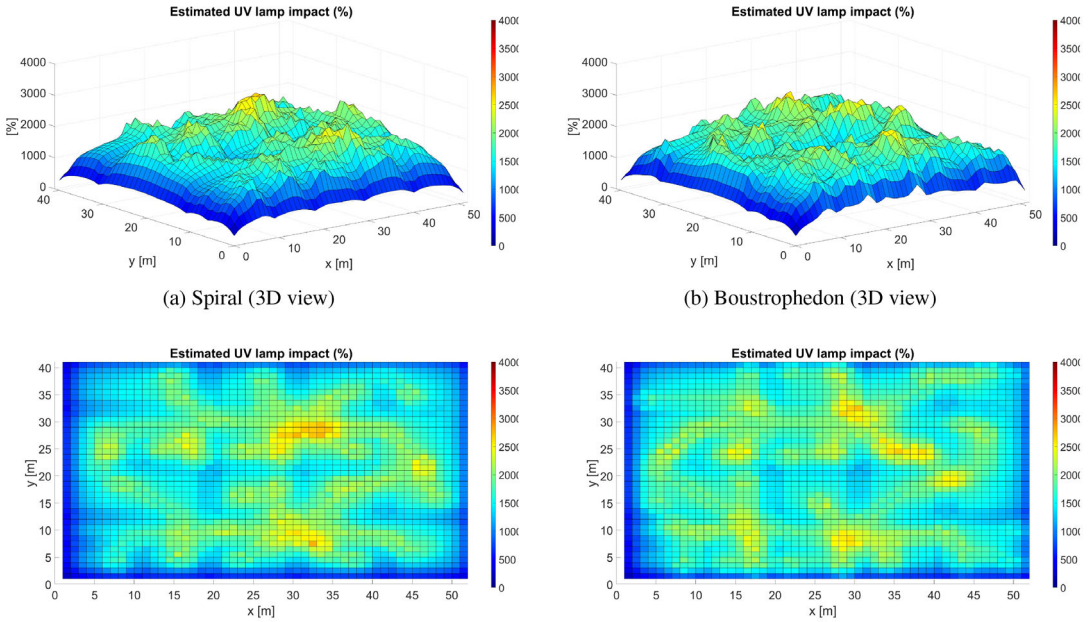
This map has a significant number of rooms. In Figure 34, the different paths followed by each robot appear. In both cases, it can be seen that the robot passes into a particular room and disinfects it. In this environment, the true potential of the proposed algorithm is tested.

In Figure 35, it can be seen that in both situations, there are some unwanted peaks. This effect is due to the increase in deadlock situations, besides the number of turns needed to move around the proposed environment. The peaks are higher in the robot that uses spiral motion.

In Figure 36, the path covered by both robots can be seen on the map. Although both routes are similarly complex, it is a fact that the robot with spiral motion uses fewer turns to perform the disinfection. The speed in both cases is also similar. Although when using boustrophedon, the speed is higher, as is usual in all the tests done.

Due to the complexity of the map, in Figures 37 and 38, it is evident that the speed controller is not able to smooth the received dose in an expected way. This result is manifested in a large number of peaks, which appear mainly due to the need to make a large number of turns to disinfect the environment and deadlocks situations.

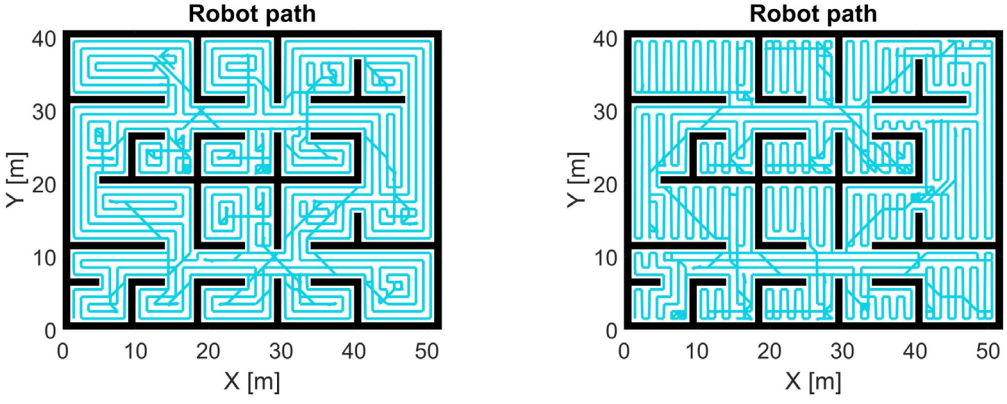
In contrast to Figure 35, after simulating the system, the speed controller has performed better than the spiral case, and we see higher peaks with boustrophedon motion.



(a) Spiral (3D view) (b) Boustrophedon (3D view)

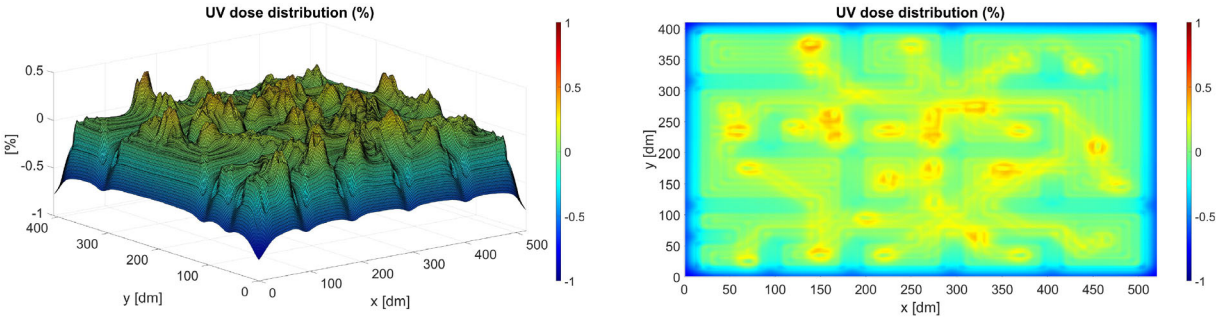
(c) Spiral (2D view) (d) Boustrophedon (2D view)

FIGURE 35 UV % estimated during the complete coverage path planning of Test 4.



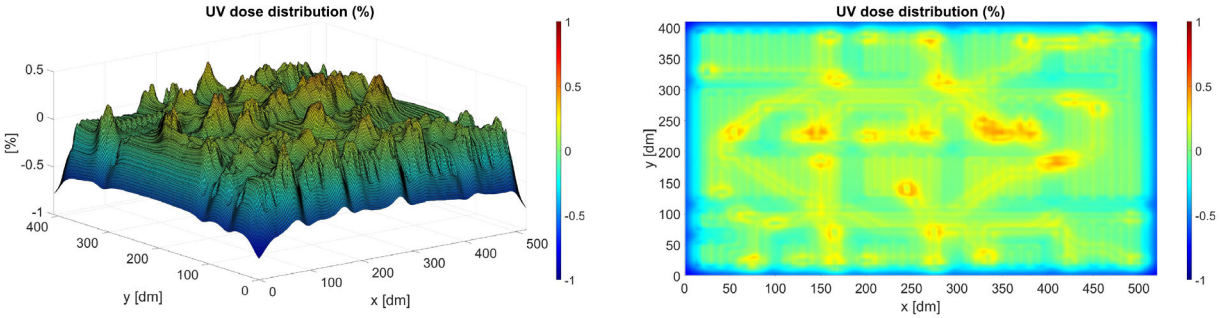
(a) Spiral (b) Boustrophedon

FIGURE 36 Robot path for Test 4.



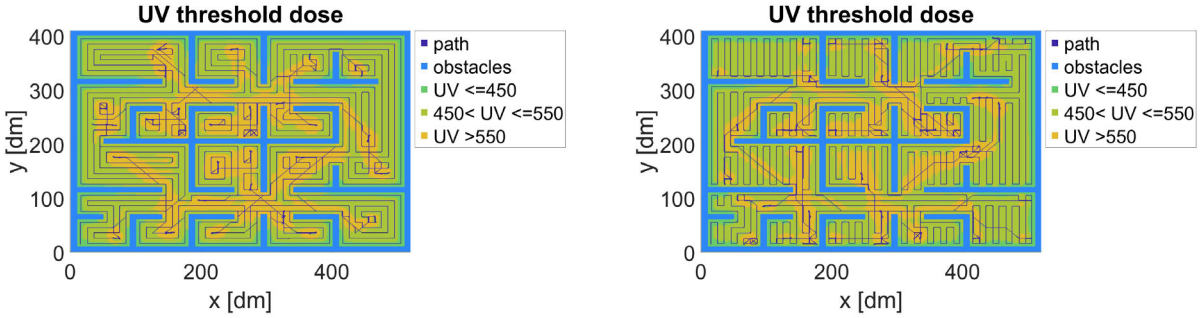
(a) D_{UV} (3D view) (b) D_{UV} (2D view)

FIGURE 37 D_{UV} received with spiral motion for Test 4.



(a) D_{UV} (3D view) (b) D_{UV} (2D view)

FIGURE 38 D_{UV} received with boustrophedon motion for Test 4.



(a) Spiral (b) Boustrophedon

FIGURE 39 D_{UV} received categorized into thresholds for Test 4.

In the Figure 37 the same occurs. Similarly, the undesired effect of the turns is visible. On this route, the number of turns is greater, and they are also closer turns, which causes higher peaks. Also, the unwanted effect of deadlock situations can be seen quite well in the 2D graph.

The radiation excess is higher than in previous cases, as shown in Figure 39. Once the complexity of the map is increased, reaching the desired radiation is not a problem. The trouble comes from the regions that exceed this threshold. This effect is quite noticeable in deadlock escape routes.

7 | CONCLUSIONS AND FUTURE WORKS

The use of UV-C radiation in combination with mobile robotics expands the capabilities of disinfection of this technique. For this purpose, in this work a complete coverage path planning algorithm for UV-C disinfection based on boustrophedon and spiral movements is proposed.

A path is obtained using neuronal activity, starting from a known environment. To do so, the GBNN algorithm, adapted to the requirements of disinfection by UV radiation, has been developed. An approximate model has been used to infer the dose received.

This approach gives a representative and useful demonstration of the results obtained with the boustrophedon and spiral motion techniques. Remarkably, the proposed speed controller can adjust the speed of the robot so that the dosage is within the desired range under normal conditions.

The algorithm has been testing in different scenarios. It is applicable to all disinfection robots and allows maximizing their utilization. Furthermore, the use of this algorithm makes a larger number of cells the desired radiation dose. Specifically, it has been observed that 229% more cells receive the required dose; the maximum dose of radiation is 44% lower, which ensures less damage to surrounding materials; the distance travelled is 17% less, which saves energy and, finally, the number of iterations is 15% less.

As future lines of work we can highlight the simulation of 3D environments instead of 2D; the simulation of other radiation patterns; consideration of materials with different absorption coefficients; and the extension of the algorithm to manage a fleet of mobile robots.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

J. E. Sierra-García https://orcid.org/0000-0001-6088-9954
M. Santos https://orcid.org/0000-0003-1993-8368

REFERENCES

Arguelles, P. Estimating uv-c sterilization dosage for covid-19 pandemic mitigation efforts, Preprint April, 2020.

Biasin, M., Bianco, A., Pareschi, G., Cavalleri, A., Cavatorta, C., Fenizia, C., Galli, P., Lessio, L., Lualdi, M., Tombetti, E., Ambrosi, A., Redaelli, E. M. A., Saulle, I., Trabattoni, D., Zanutta, A., & Clerici, M. (2021). Uv-c irradiation is highly effective in inactivating sars-cov-2 replication. *Scientific Reports*, *11*(1), 1–7.

Burgos-Artizzu, X. P., Ribeiro, A., & de Santos, M. (2007). Controlador borroso multivariable para el ajuste de tratamientos en agricultura de precisión. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, *4*(2), 64–71.

Chen, B., Zhang, H., Zhang, F., Liu, Y., Tan, C., Yu, H., & Wang, Y. (2022). A multi-robot distributed collaborative region coverage search algorithm based on Glasius bio-inspired neural network. *IEEE Transactions on Cognitive and Developmental Systems*, *15*, 1449–1462.

Chen, M., & Zhu, D. (2019). Real-time path planning for a robot to track a fast moving target based on improved glasius bio-inspired neural networks. *International Journal of Intelligent Robotics and Applications*, *3*(2), 186–195.

Coulter, R. C. (1992). Implementation of the pure pursuit path tracking algorithm, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST. Tech. Rep.

Derraik, J. G., Anderson, W. A., Connelly, E. A., & Anderson, Y. C. (2020). Rapid review of sars-cov-1 and sars-cov-2 viability, susceptibility to treatment, and the disinfection and reuse of ppe, particularly filtering facepiece respirators. *International Journal of Environmental Research and Public Health*, *17*(17), 6117.

Fischer, R. J., Morris, D. H., van Doremalen, N., Sarchette, S., Matson, M. J., Bushmaker, T., Yinda, C. K., Seifert, S. N., Gamble, A., Williamson, B. N., Judson, S. D., de Wit, E., Lloyd-Smith, J. O., & Munster, V. J. (2020). Effectiveness of n95 respirator decontamination and reuse against sars-cov-2 virus. *Emerging Infectious Diseases*, *26*(9), 2253–2255.

Han, L., Tan, X., Wu, Q., & Deng, X. (2023). An improved algorithm for complete coverage path planning based on biologically inspired neural network. *IEEE Transactions on Cognitive and Developmental Systems*, *15*, 1605–1617.

Kowalski, W., Walsh, T., & Petraitis, V. (2020). *Covid-19 coronavirus ultraviolet susceptibility*. Purplesun Inc.

Li, J., Xu, Z., Zhu, D., Dong, K., Yan, T., Zeng, Z., & Yang, S. X. Bio-inspired intelligence with applications to robotics: a survey, *arXiv preprint arXiv*: *2206.08544*, 2022.

Li, J., Yang, S. X., & Xu, Z. (2019). A survey on robot path planning using bio-inspired algorithms. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, *2019*, 2111–2116.

Luo, C., & Yang, S. X. (2008). A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Transactions on Neural Networks*, *19*(7), 1279–1298.

Munoz-Ceballos, N. D., & Suarez-Rivera, G. (2022). Criterios de desempeño para evaluar algoritmos de navegación de robots móviles: una revisión. *RIAI Revista Iberoamericana de Automatica e Informatica Industrial*, *19*(2), 132–143.

Muthugala, M. V. J., Samarakoon, S. B. P., & Elara, M. R. (2022). Toward energy-efficient online complete coverage path planning of a ship hull maintenance robot based on Glasius bio-inspired neural network. *Expert Systems with Applications*, *187*, 115940.

Rodrigo, D. V., Sierra-Garca, J. E., & Santos, M. (2023). Glasius bio-inspired neural networks based uv-c disinfection path planning improved by preventive deadlock processing algorithm. *Advances in Engineering Software*, *175*, 103330.

Sabino, C. P., Sellera, F. P., Sales-Medina, D. F., Machado, R. R. G., Durigon, E. L., Freitas-Junior, L. H., & Ribeiro, M. S. (2020). Uv-c (254 nm) lethal doses for sars-cov-2. *Photodiagnosis and Photodynamic Therapy*, *32*, 101995.

Samarakoon, S. B. P., Muthugala, M. V. J., & Elara, M. R. (2022). Online Complete Coverage Path Planning of a Reconfigurable Robot using Glasius Bio-inspired Neural Network and Genetic Algorithm, *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1em plus 0.5em minus 0.4em IEEE. 5744–5751.

San Juan, V., Santos, M., & Andújar, J. M. (2018). Intelligent uav map generation and discrete path planning for search and rescue operations. *Complexity*, *2018*, 1–17.

Sánchez, R., Sierra-Garca, J. E., & Santos, M. (2022). Modelado de un agv hbrido triciclo-diferencial. *RIAI Revista Iberoamericana de Automatica e Informatica Industrial*, *19*(1), 84–95.

Sellera, F. P., Sabino, C. P., Cabral, F. V., & Ribeiro, M. S. (2021). A systematic scoping review of ultraviolet c (uvc) light systems for sars-cov-2 inactivation. *Journal of Photochemistry and Photobiology*, *8*, 100068.

Sierra Garcia, J., Guillen-Grima, F., Garcia-Garcia, M., Cascajar, L., & Rodriguez-Merino, F. (2021). Evaluation of an uvc robot for terminal disinfection of hospital rooms. *Antimicrobial Resistance and Infection Control*, *10*(S1).

Sierra-Garcia, J. E., & Santos, M. (2022). Combining reinforcement learning and conventional control to improve automatic guided vehicles tracking of complex trajectories. *Expert Systems*, e13076. https://doi.org/10.1111/exsy.13076

Spanish Standardization Association. (2020). Une 0068:2020 safety requirements for uv-c equipment used for room air disinfection and surfaces. https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0064094

Sun, B., Zhu, D., Tian, C., & Luo, C. (2018). Complete coverage autonomous underwater vehicles path planning based on glasius bio-inspired neural network algorithm for discrete and centralized programming. *IEEE Transactions on Cognitive and Developmental Systems*, *11*(1), 73–84.

Tan, X., Han, L., Gong, H., & Wu, Q. (2023). Biologically inspired complete coverage path planning algorithm based on Q-learning. *Sensors*, *23*(10), 4647.

Xing, B., Wang, X., Yang, L., Liu, Z., & Wu, Q. (2023). An algorithm of complete coverage path planning for unmanned surface vehicle based on reinforcement learning. *Journal of Marine Science and Engineering*, *11*(3), 645.

Yao, P., Wu, K., & Lou, Y. (2022). Path planning for multiple unmanned surface vehicles using Glasius bio-inspired neural network with Hungarian algorithm. *IEEE Systems Journal*, *17*, 3906–3917.

Yao, P., & Zhao, Z. (2021). Improved glasius bio-inspired neural network for target search by multi-agents. *Information Sciences*, *568*, 40–53.

Yi, L., Wan, A. Y. S., Le, A. V., Hayat, A. A., Tang, Q., & Mohan, R. E. (2023). Complete coverage path planning for reconfigurable omni-directional mobile robots with varying width using GBNN (n). *Expert Systems with Applications*, *228*, 120349.

AUTHOR BIOGRAPHIES

Daniel Vicente Rodrigo Muñoz was born in Ciudad Real, Spain. He received the B.Sc. degree in industrial electronic and automatic engineering from Universidad de Castilla La-Mancha, Ciudad Real, Spain, in 2016, the M.Eng. degree in industrial engineering from Universidad Politécnica de Madrid, Madrid, Spain, in 2020 and the M.Sc in system and control engineering from Universidad Complutense de Madrid, Madrid, Spain, in 2021. He is currently an R&D embedded software engineer in the field of Autonomous Driving at TTTech Auto, Madrid, Spain. His current research interests include autonomous vehicle control, neural networks, and artificial intelligence.

J. Enrique Sierra-Garcia was born in Burgos, Spain. He received his M.Sc. degrees in Electronics and Telecommunications from the University of Valladolid (UVA) in 2007 and 2015 respectively, his M.Sc degree in Control Engineering from the National University for Distance Education (UNED) in 2014, and his Ph.D in Computer Science in 2019. Since 2012 he has been with the University of Burgos, where he is currently a Lecturer in System Engineering and Automatic Control in the Department of Electromechanical Engineering. He is the founder of the Joint Research Unit ASTI-UBU on Autonomous Vehicles, Mobile Robotics and AGVs. He has been principal researcher in more than 10 research projects related with mobile robotics. His major research interests are: Intelligent Control, Robotics, Autonomous Guided Vehicles, Modeling, Simulation, Wind energy.

Matilde Santos was born in Madrid, Spain. She received her B.Sc. and M.Sc. degrees in Physics (Computer Engineering) and her Ph.D in Physics from the University Complutense of Madrid (UCM). She is currently Full Professor in System Engineering and Automatic Control. She is member of the European Academy of Sciences and Arts. She has published many papers in international scientific journals and several books and book chapters. She has supervised more than 12 Ph.Ds. She has worked on several national, European and international research projects, leading some of them. She currently serves as member of the editorial board of prestigious journals and she is editor-in-chief assistant of one of them. Her major research interests are: Intelligent Control (fuzzy and neuro-fuzzy), Pattern Recognition, Modelling and Simulation, Wind energy.

How to cite this article: Rodrigo, D. V., Sierra-García, J. E., & Santos, M. (2023). GBNN algorithm enhanced by movement planner for UV-C disinfection. *Expert Systems*, *40*(10), e13455. https://doi.org/10.1111/exsy.13455