Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Optimization of Targeted Aerosol Disinfection via Integrated Physical Modeling and B-Spline Trajectory Planning

Manuel Alò

manuel.alo@supsi.ch

University of Applied Sciences and Arts of Southern Switzerland

Carlos Alberto Gagliano

99 Technologies S.A.

Mikael Andreas Bianchi

University of Applied Sciences and Arts of Southern Switzerland

Research Article

Keywords: Coverage Path Planning, B-Spline Trajectory Optimization, Constrained Traveling Salesman Problem, Autonomous Disinfection Robots

Posted Date: March 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8838805/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Optimization of Targeted Aerosol Disinfection via Integrated Physical Modeling and B-Spline Trajectory Planning

Manuel Alò[1*], Carlos Alberto Gagliano[2] and Mikael Andreas Bianchi[1]

[1*]Institute of Systems and Applied Electronics (ISEA), University of Applied Sciences and Arts of Southern Switzerland (SUPSI), Lugano, TI, Switzerland.

[2]Technical Support Specialist, 99 Technologies S.A., Lugano, TI, Switzerland.

*Corresponding author(s). E-mail(s): manuel.alo@supsi.ch;
Contributing authors: c.gagliano@99technologies.ch;
mikael.bianchi@supsi.ch;

Abstract

Effective autonomous disinfection in healthcare and office environments requires a precise balance between disinfectant coverage uniformity and operational efficiency. Unlike traditional full-coverage methods that rely on room saturation, this paper introduces a system-integrated framework that leverages a physical deposition model to drive an optimized trajectory-planning strategy. The methodology identifies critical disinfection keypoints via spatial analysis and sequences them using a constrained Traveling Salesman Problem (TSP) tailored to the robot's navigable space. The core innovation lies in a unified mathematical architecture that decouples the robot's navigation path from the disinfection target trajectory through B-Spline representations. By integrating an anisotropic Gaussian model to account for the disinfectant's spatiotemporal behavior, the framework enables independent optimization of robot mobility and spray orientation. This architecture provides real-time adaptability to environmental constraints by allowing independent control of navigation and nebulization without the need for computationally expensive replanning. The framework was validated through experimental trials in real-world indoor scenarios using a custom-built robotic platform. Results demonstrate a markedly more uniform disinfectant distribution on target surfaces, with concentration variability reduced from approximately

25% to 6%, underscoring that room saturation does not inherently ensure concentration homogeneity across specific points of interest. Furthermore, the methodology reduces execution time by nearly 50% and decreases disinfectant consumption by more than 40% compared to a brute-force strategy.

Keywords: Coverage Path Planning, B-Spline Trajectory Optimization, Constrained Traveling Salesman Problem, Autonomous Disinfection Robots

1 Introduction

In healthcare environments, effective disinfection is crucial for preventing infections and maintaining a safe environment for patients and staff. Different disinfection technologies are used in autonomous systems, including UV-C radiation and hydrogen peroxide aerosol. Compared to UV-C, hydrogen peroxide nebulization offers the advantage of reaching exposed and hidden surfaces by dispersing through the air. While this method is effective, achieving optimal dispersion and maintaining the correct H_2O_2 concentration can be challenging in complex environments. Traditional peroxide-based strategies often rely on a fixed-position fogging device that saturates the entire room volume to achieve microbial neutralization. However, this brute-force room-saturation approach is inefficient, leading to prolonged operational times, excessive use of hazardous disinfectants, and extended periods when treated areas must remain unoccupied. To mitigate these limitations, recent efforts have explored the use of mobile robots capable of navigating the environment to deliver disinfectant to specific target regions. This setting introduces the challenge of trajectory design, where the robot must visit and orient towards localized targets while ensuring adequate exposure, process homogeneity, and operational efficiency. Initial improvements focused on modeling the disinfection process to refine coverage strategies [1]. This trend is mirrored in related fields, where continuous trajectories modulate robot motion (e.g., speed profiles) to guarantee the required dose delivery [2], and advanced methods use GPU acceleration or linear programming to optimize dosage with precision and occlusion awareness [3, 4]. More recently, neural-based strategies have shown promise for adaptive planning in cluttered environments [5]. Despite their sophistication, these dosage-centric methods do not explicitly model the complex spatiotemporal distribution of an aerosol. Other general Coverage Path Planning (CPP) strategies, while highly developed, are also ill-suited. Techniques like dividing environments into planar patches [6], designing energy-efficient paths [7], or using frequency-domain analysis for precision [8] focus on geometric or kinematic optimization, not the physical behavior of the disinfectant. Similarly, algorithms developed for multi-robot systems, such as those for distributed line coverage [9], hierarchical exploration [10], or decentralized task allocation [11, 12], are designed for different scales and objectives, introducing unnecessary computational overhead for a single robot operating in a constrained indoor space. Rather than relying on generic coverage, the challenge of autonomous disinfection is more accurately framed as a specialized case of persistent monitoring

and area coverage. As discussed by [13], minimizing uncertainty in environmental monitoring requires an optimized balance between trajectory speed and sensor dwell time. In the context of aerosol-based disinfection, this translates to maintaining a consistent concentration gradient, similar to the precision requirements found in autonomous agricultural spraying systems [14], where machine vision and nozzle control are integrated to minimize chemical waste while ensuring total coverage. To address these limitations, this paper introduces an adaptive trajectory planning methodology tailored for autonomous aerosol-based disinfection. Our core innovation is a framework that defines the robot's navigation path and disinfection trajectory (i.e., the control of the aerosol release parameters) within a unified mathematical structure. This allows them to be modeled and optimized independently, yet executed as a single, seamless motion. By modeling aerosol distribution across target surfaces, our approach eliminates the need for room saturation and instead focuses on efficiently achieving a desired concentration profile. This integration provides the real-time flexibility needed to adapt to dynamic environments and operational constraints without requiring computationally expensive replanning, thereby minimizing disinfectant usage, reducing operational time, and accelerating the safe re-entry of personnel into the treated space.

2 Setup description

The disinfection robot is designed with a mobile base, as shown in Fig. 1, that incorporates two traction wheels and a pivot wheel, allowing it to maneuver along straight lines and curved trajectories. The robot operates within the ROS (Robot Operating System) framework, specifically using the NAV2 stack, which integrates mapping, localization, and obstacle avoidance, allowing the robot to adapt dynamically to its surroundings. For these operations, the robot employs a rotating LiDAR sensor positioned at the base level. This sensor continuously scans the environment to construct a 2D occupancy grid map, which is stored as an image representation (an example of which is shown in Fig. 2a) and used by the robot for navigation and planning. Serving as the basis for trajectory planning, the environment map is preprocessed to extract essential spatial features, such as the environment's geometry and obstacle locations, to define the robot's navigable space (Fig. 2b). Disinfection targets are identified using data from a stereo camera with an integrated LiDAR module, which detects surfaces within a specified height range. This spatial information is then merged with the preprocessed environment map to generate the disinfection target map (Fig. 2c). The combination of these datasets enables the robot to compute trajectories that account for both the environment's physical layout and the specific zones requiring disinfection. Mounted atop the mobile base, a rotating platform supports the nebulization unit, allowing the nebulizer to orient independently of the robot's movement. The unit includes ten ultrasonic nebulizers that atomize hydrogen peroxide into micrometer-scale droplets and a fan system that adjusts aerosol dispersal distance based on its speed.



Fig. 1 Robot prototype

3 Proposed Methodology

The architecture of the proposed trajectory generation framework follows a decoupled approach, as illustrated in Figure 3, which independently processes the navigation and disinfection domains before synthesizing them into a unified multi-dimensional trajectory. The process begins with integrating environmental occupancy grids and disinfection target data through a preliminary spatial analysis to identify navigable areas and establish safe operational boundaries. Following this, the framework branches into two parallel geometric pre-processing pipelines: the navigation domain focuses on computing the Robot Maximal Enclosing Path (RMEP) to ensure collision-free movement, while the disinfection domain extracts localized target keypoints and groups them using the DBSCAN clustering algorithm to define distinct treatment zones. Once these geometric baselines are established, both domains undergo a task-specific optimization phase. The RMEP is refined through boundary and waypoint adjustment to optimize approach maneuvers, while the keypoint sequence is ordered

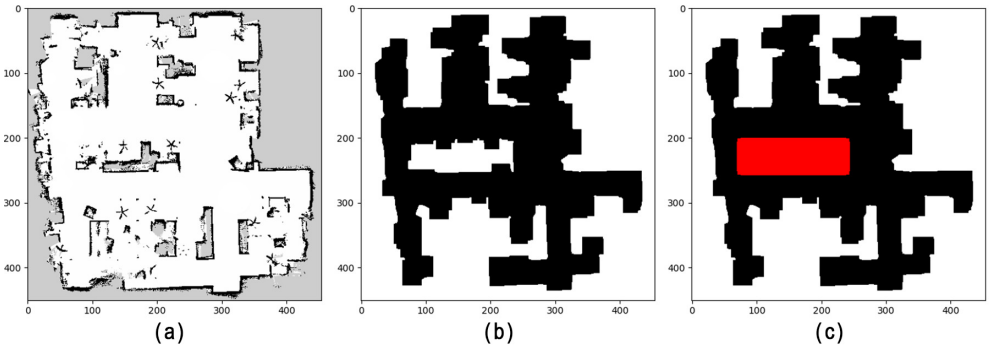


Fig. 2 Map definitions needed for the robot trajectory computation. (a) Environment map, (b) Preprocessed map, (c) Target map

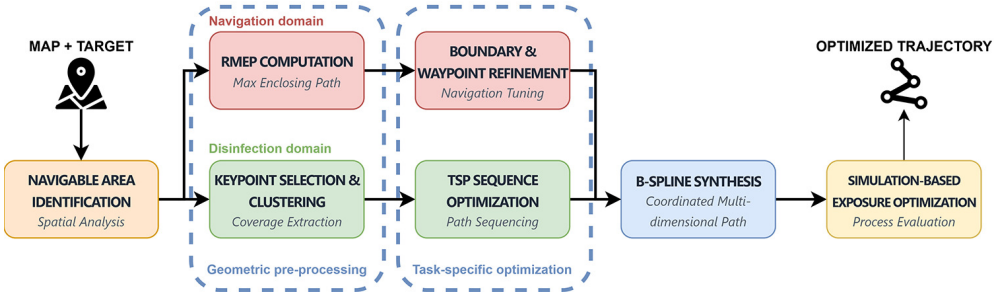


Fig. 3 Flowchart of the algorithm. Icons made by Freepik from www.flaticon.com

by solving a constrained Traveling Salesman Problem (TSP) to identify the most efficient traversal path. In the synthesis phase, these optimized components are merged into a unified 4D B-Spline representation. This multi-dimensional architecture enables the simultaneous control of the robot's pose and the nebulizer's orientation within a single, continuous mathematical structure. Finally, a simulation-based exposure evaluation is conducted to verify disinfectant coverage and spatial homogeneity. In this phase, maintaining a constant robot velocity is enforced as a critical kinematic constraint to preserve the ratio between the platform's speed and the aerosol flow rate, thereby ensuring an even disinfectant layer [8].

4 B-Spline Representation for Navigation and Disinfection

To ensure the efficient execution of both navigation and disinfection processes, we approximate their trajectories using B-Spline curves. B-Splines provide a computationally efficient and flexible representation that ensures smooth, continuous motion while minimizing curvature variations, making them particularly suitable for real-time

robotic applications [15]. They can also be optimized to improve trajectory smoothness and reduce execution time [16]. This approach has been widely adopted in trajectory planning for autonomous systems, including obstacle avoidance and high-speed path traversal [17]. By modeling both the robot's motion and the disinfection path using B-Splines, a robust and adaptable trajectory formulation can be achieved, regardless of navigation deviations. Mathematically, the two trajectories are defined as:

$$\mathbf{r}_n(t) = \sum_{i=0}^{N} \mathbf{P}_i B_{i,K}(t, U_n), \quad t \in [t_0, t_m] \tag{1}$$

$$\mathbf{r}_d(s) = \sum_{j=0}^{M} \mathbf{Q}_j B_{j,K}(s, U_d), \quad s \in [s_0, s_m] \tag{2}$$

where:

- $\mathbf{r}_n(t)$ represents the navigation trajectory with control points $\mathbf{P}_i$ and knot vector $U_n = \{u_{n,0}, \ldots, u_{n,N+K+1}\}$.
- $\mathbf{r}_d(s)$ represents the disinfection trajectory with control points $\mathbf{Q}_j$ and knot vector $U_d = \{u_{d,0}, \ldots, u_{d,M+K+1}\}$.
- $B_{i,K}(t, U_n)$ and $B_{j,K}(s, U_d)$ are the B-Spline basis functions of degree K.
- t and s are independent parameters for navigation and disinfection, respectively.

The trajectory definition using a B-Spline representation starts by selecting significant vertices to construct a sequence of Bézier curves following the methodology described in [18], then concatenating them to generate a unique B-Spline. The single segments are classified as linear or curved based on geometric continuity constraints. As demonstrated in [18], the G^2 continuity of the individual Bézier curves can be preserved after the concatenation by imposing at the junctions of successive spline segments the following constraints:

$$D\mathbf{r}_d(s_e) = D\mathbf{r}_d(s_s), \quad D^2\mathbf{r}_d(s_e) = D^2\mathbf{r}_d(s_s) \tag{3}$$

where s_s and s_e represent a segment's starting and ending parameters. This guarantees that both velocity and acceleration transitions are smooth, preventing abrupt changes. Finally, to achieve a constant-speed execution of the B-Spline trajectories, the knot vectors are redefined through arc-length-based sampling to ensure constant speed when the spline is traversed at uniform temporal intervals. Instead of using explicit computation or inversion of the arc-length function, a sequence of points along the curve spaced at equal distances has been extracted, and normalized parameter values have been assigned accordingly [19, 20].

4.1 Unified B-Spline parametrization of Navigation and Disinfection Trajectories

To ensure the seamless and synchronized execution of both navigation and disinfection processes, the two B-Spline trajectories are reparametrized into a single unified B-Spline. This approach ensures a precise alignment between the two processes while

simplifying the trajectory representation. The reparametrization process begins by adapting the length of the knot vector of the disinfection trajectory to match that of the navigation trajectory:

$$U_d^* = \frac{U_d}{u_{d,M+K+1}} \cdot u_{n,N+K+1} \tag{4}$$

Subsequently, a knot insertion is performed on both B-Splines to unify their knot sequences, ensuring compatibility for merging (same degree and common knot vector U). The final 4D trajectory is then constructed by pairing the control points index-wise and stacking them into a single control-point sequence $\mathbf{C}_i = [P_{x,i}, P_{y,i}, Q_{x,i}, Q_{y,i}]^T$, while using the unified knot vector U, obtaining:

$$\mathbf{r}(u) = \sum_{k=0}^{L} \mathbf{C}_k B_{u,K}(u, U), \quad u \in [u_0, u_m] \tag{5}$$

where L denotes the number of control points after knot unification. By applying a knot reduction after merging, the resulting B-Spline becomes more compact, computationally efficient, and easier to execute. The B-Spline trajectory is now parameterized to yield four components:

$$\mathbf{r}(u) = [x_n(u), y_n(u), x_d(u), y_d(u)] \tag{6}$$

where $X_n = [x_n(u), y_n(u)]$ are the coordinates of the navigation trajectory and $X_d = [x_d(u), y_d(u)]$ are those of the disinfection target trajectory.

5 Robot trajectory evaluation

To plan the robot's trajectory, the algorithm starts by extrapolating from the map the isolated areas representing a specific disinfection target. Each area is approximated using a convex polygon, which simplifies the generation of smooth, collision-free trajectories and reduces the chances of the algorithm getting stuck in local minima [21]. Next, the robot's physical dimensions and the nebulizer jet's reach are used to define a navigable safety area around each polygon. In practice, the surroundings of the polygon are evaluated in three safety boundaries, as shown in Fig. 4a: the Inner Safety Boundary (ISB, in red), which ensures collision avoidance; the Optimal Spraying Boundary (OSB, in yellow), representing the ideal nebulization distance; and the Maximum Spraying Boundary (MSB, in green), indicating the furthest range the spray can effectively reach. The combination of these three boundaries defines the safety zone (blue area), within which the robot can navigate around the target. From the safety area, an open or closed polyline is extracted by selecting the portion closest to the disinfection target, representing the initial trajectory candidate for the robot. An offset operation is then applied to this path, ensuring the robot maintains an optimal distance from the disinfection target and, at the same time, helping smooth out sharp corners, as illustrated in Fig. 4b and Fig. 4c. Finally, the vertices of this processed

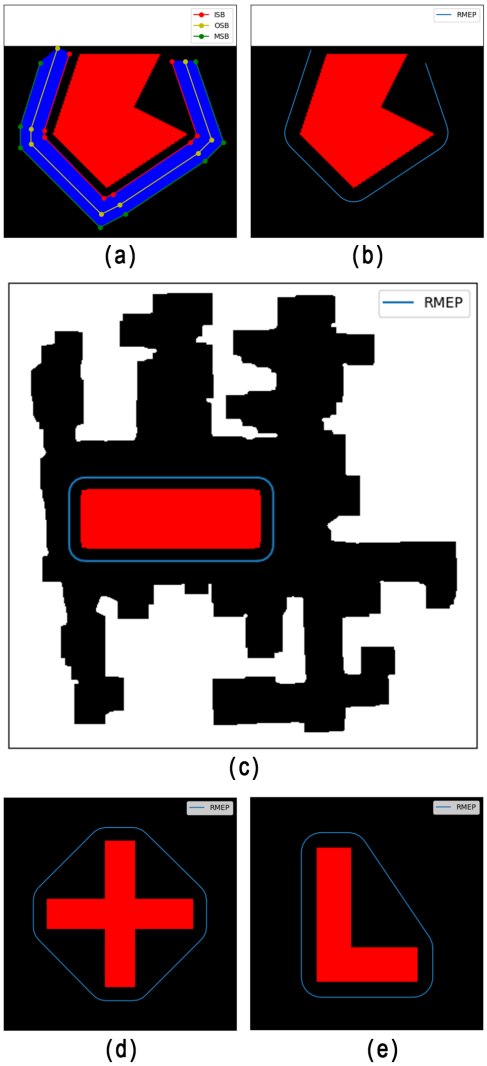


Fig. 4 Robot maximal enclosing path (RMEP) representation in different scenarios.

path are used to generate a B-Spline curve (as described in section 4), which defines the robot's maximal enclosing path (RMEP) around each different disinfection area. This initial path offers a flexible basis that can be adapted to specific disinfection process constraints, such as minimizing the overall path length or avoiding restricted zones within the environment. Such adaptations can be implemented by adjusting the B-Spline trajectory by redefining its start and end points, while ensuring the robot maintains a constant speed during operation. The use of convex approximation in polygon generation directly affects the final robot trajectory, leading to situations where the robot is far from the disinfection area, as shown in Fig. 4d and Fig. 4e. This must be taken into account in the mechanical sizing of the nebulizer, ensuring that the jet

can reach even the greatest distances, or it must be introduced as an optimization parameter for the robot's scaled trajectory.

6 Nebulization trajectory evaluation

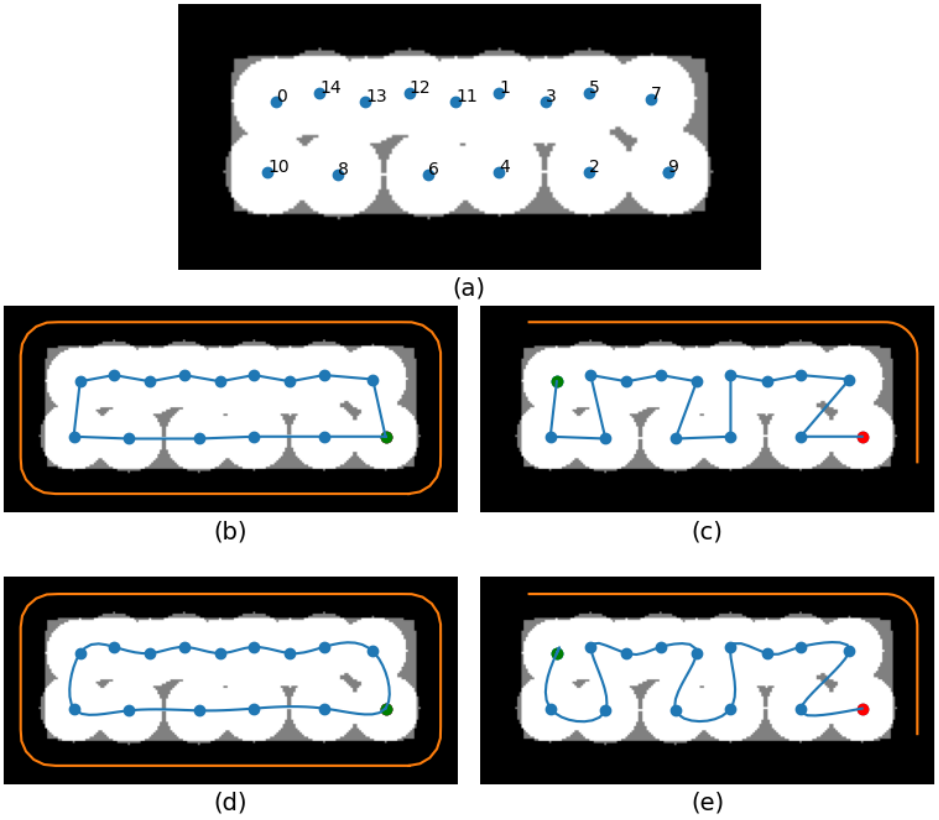


Fig. 5 Example of nebulization trajectory generation. (a) Placement of key points for coverage path planning; (b) TSP-based trajectory covering the RMEP; (c) TSP-based trajectory limited to a subset of the RMEP; (d) B-Spline representation of the disinfection trajectory based on the RMEP; (e) B-Spline representation of the disinfection trajectory based on a partial RMEP.

This section outlines the methodology used to generate the nebulization trajectory to achieve uniform and complete disinfection of designated target areas. The process consists of structured steps that ensure both geometric coverage and feasibility for robotic execution. Initially, keypoints are identified along the perimeters of the target regions. These points are selected based on the spatial distribution of the surfaces to be disinfected and the nebulizer's effective range. Their purpose is to guide the

robot along a path that maximizes aerosol exposure over critical areas. Subsequently, the keypoints are connected through a graph-based representation, and the optimal visiting sequence is computed by solving a Traveling Salesman Problem (TSP). This optimization step minimizes the overall trajectory length while ensuring full keypoint coverage.

6.1 Keypoint selection

The objective is to determine a set of keypoints (x_i, y_i) such that a set of circles with radius r maximizes the coverage of the accessible target area while minimizing the number of placements. Let $A_{\text{initial}}(i,j) \in \{0,1\}$ denote a binary occupancy map defined on a discrete spatial grid, where:

$$A_{\text{initial}}(i,j) = \begin{cases} 1, & \text{target area to be disinfected} \\ 0, & \text{external area.} \end{cases} \tag{7}$$

For each candidate grid cell (i,j), a coverage score is computed as:

$$C(i,j) = \sum_{(p,q)\in B_r(i,j)} A(p,q), \tag{8}$$

where $B_r(i,j)$ denotes the discrete circular neighborhood of radius r centered at (i,j):

$$B_r(i,j) = \{(p,q) \mid (p-i)^2 + (q-j)^2 \le r^2\}. \tag{9}$$

At each iteration, the keypoint maximizing the coverage score is selected:

$$(i^*, j^*) = \arg\max_{(i,j)} C(i,j). \tag{10}$$

After selecting (i^*, j^*), the covered region is removed from the remaining target area by updating the binary map as:

$$A_{\text{new}}(i,j) = A_{\text{old}}(i,j) \cdot \left(1 - \mathbb{I}_{B_r(i^*,j^*)}(i,j)\right), \tag{11}$$

where $\mathbb{I}_{B_r}$ denotes the indicator function of the circular region. The iterative process continues until the fraction of uncovered area satisfies:

$$\frac{\sum_{i,j} A(i,j)}{\sum_{i,j} A_{\text{initial}}(i,j)} < \epsilon, \tag{12}$$

where ϵ defines the stopping threshold. In this work, ϵ is typically set to 5%. To further reduce redundancy and improve spatial homogeneity, the selected keypoints are clustered using a DBSCAN algorithm with a distance threshold equal to the radius r. This step merges nearby keypoints into representative clusters, optimizing their distribution without affecting coverage quality. Fig. 5a shows an example of the keypoints placement based on a specific target area.

6.2 Coverage Guarantee with Greedy Approximation

The keypoint placement strategy employs a greedy approach to maximize the coverage of the target area. While the general coverage problem is NP-hard, the specific coverage function C is non-negative, monotone, and submodular. Consequently, the greedy algorithm provides a provable approximation bound of $(1 - \frac{1}{e}) \approx 63\%$ of the optimal solution, as established by Nemhauser [22]:

$$C(S^*) \geq \left(1 - \frac{1}{e}\right) C(S_{\text{opt}}) \tag{13}$$

where S^* is the set of keypoints selected by the greedy algorithm, and S_{opt} is the optimal set. This theoretical bound is particularly relevant for the proposed framework for two main reasons. First, it ensures that the selected keypoints achieve a constant-factor approximation of the optimal coverage without the computational overhead of global optimization. Second, since the coverage integrity is established at the keypoint level, the validity of the disinfection sites remains invariant to subsequent path modifications. Whether the continuous trajectory is altered by energy constraints or by local obstacle avoidance, the fundamental coverage guarantee is preserved. This mathematical property justifies the proposed decoupling strategy, enabling real-time navigation adaptations without recomputing the global coverage plan.

6.3 Optimization of Keypoint Traversal Sequence

The sequencing of disinfection keypoints is optimized using a variant of the Traveling Salesman Problem (TSP) formulation, where the first and last nodes are fixed. To align the keypoints visitation sequence with the robot B-Spline motion path, the keypoints closest to the initial and final positions of the trajectory are identified by minimizing the Euclidean distance $d((x, y), r_n(t))$, between the keypoints and the start and end of the B-Spline curve (Eq. 1):

$$(x_{\text{start}}, y_{\text{start}}) = \arg \min_{(x,y) \in C} d((x, y), r_n(t_0)) \tag{14}$$

$$(x_{\text{end}}, y_{\text{end}}) = \arg \min_{(x,y) \in C} d((x, y), r_n(t_m)) \tag{15}$$

Once the start and end points are fixed, the remaining keypoints are ordered by solving a TSP with constrained endpoints. We adopt the Held-Karp dynamic programming algorithm to minimize the total traversal cost:

$$\min_{\pi} \sum_{i=1}^{N-1} d(C_{\pi(i)}, C_{\pi(i+1)}) \tag{16}$$

subject to:

$$\pi(1) = (x_{\text{start}}, y_{\text{start}}), \quad \pi(N) = (x_{\text{end}}, y_{\text{end}}) \tag{17}$$

where π is a permutation representing the keypoint visitation order, and $C_{\pi(i)}$ indicates the i-th keypoint in the optimized sequence. The resulting ordered set of points defines the sequence:

$$P = C_{\pi(1)}, C_{\pi(2)}, \ldots, C_{\pi(N)} \tag{18}$$

which serves as input to the final B-Spline interpolation, resulting in a smooth, executable trajectory for the disinfection process, as described in Eq. 2. A key advantage of this method is its modularity; once the disinfection keypoints have been computed and ordered, e.g., using the maximal enclosing path to define the start and end positions (Fig. 5.b), the robot trajectory can be reshaped (e.g., shortened, smoothed, or adapted to dynamic constraints) without modifying the keypoints themselves. This decoupling between keypoint placement and disinfection path generation enables iterative refinement of the robot's motion plan, even using a subset of the original trajectory. For instance, Fig. 5.c shows a case where only half of the planned path is used. Such flexibility allows the system to adapt to environmental changes in scenarios involving real-time constraints or energy limitations. Moreover, this approach enables better connections between multiple trajectories from different target areas by optimizing the start and end points of the robot's motion. Since disinfection coverage is intrinsically encoded in the keypoint placement, process reliability is preserved regardless of the specific path optimization strategy. The only trade-off is the need to adjust the robot's speed based on the total path length.

7 Simulation-Based Evaluation of Nebulization Trajectories

After computing the final B-Spline trajectory that connects the ordered disinfection keypoints (see examples in Fig. 5.d and Fig. 5.e), the next step is to simulate aerosol dispersion along the trajectory to evaluate the effectiveness of the disinfection process. To do that a model of the peroxide aerosol distribution is required first.

7.1 Emulation of the Nebulization Process

Hydrogen peroxide aerosol distribution and deposition are simulated to model the nebulization process using a Computational Fluid Dynamics (CFD) pre-analysis (Fig. 6a). This simulation helps us understand the behavior of the aerosol jet upon injection and its resulting deposition on the target surface (Fig. 6b). Analysis of the spatial distribution from these simulations confirmed that the complex deposition pattern can be approximated by an unnormalized anisotropic Gaussian distribution of flow density:

$$G(x, y) = \exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}\right) \tag{19}$$

where:

- μ_x, μ_y are the means along the x and y axes, assumed to be zero in our case;
- σ_x, σ_y are the standard deviations along the x and y axes;
- (x, y) are the coordinates in the two-dimensional space.

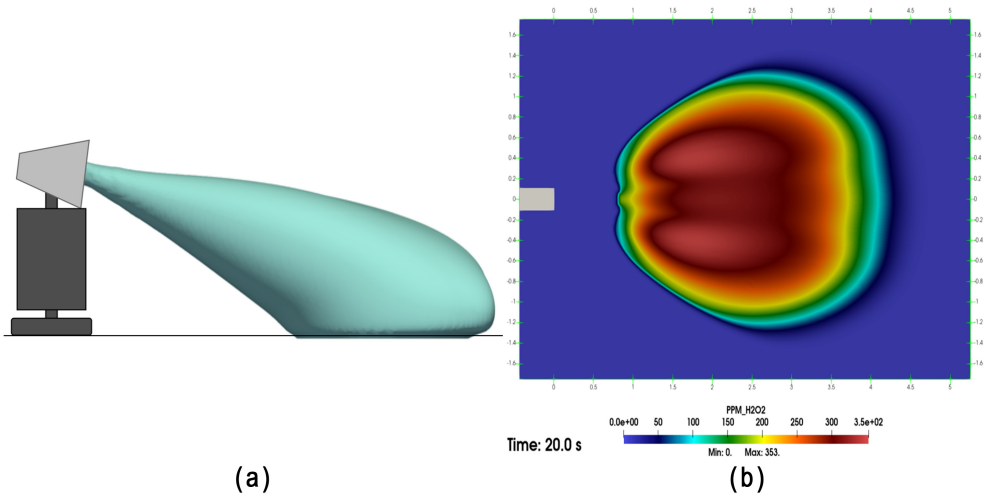


Fig. 6 Model of aerosol spatial distribution

This approximation is supported by its proven effectiveness in modeling aerosol dispersion in industrial contexts, with strong agreement to experimental data [23, 24], aligning with recent probabilistic coverage and search approaches that employ Gaussian and kernel-based modeling to improve efficiency and precision. It is further validated in biomedical and indoor aerosol studies, where Gaussian models consistently capture symmetric deposition patterns while accounting for turbulence and inertial dynamics [25–27]. To integrate the deposition model within the robot navigation and mapping framework, the continuous kernel is discretized using the same spatial scaling factor s (m/pixel) adopted for the LiDAR-based occupancy map. A finite computational window is defined by truncating the kernel at $\pm k\sigma$, resulting in a discrete support of $N_x = 2\left\lceil \frac{k\sigma_x}{s} \right\rceil + 1$ and $N_y = 2\left\lceil \frac{k\sigma_y}{s} \right\rceil + 1$ (in pixels), where k is a numerical truncation factor used to limit the Gaussian distribution. In this work, $k = 2$ is adopted, corresponding to a truncation that retains the dominant portion of the spatial distribution. The physical coordinates associated with each discrete cell (i, j) are defined as $x_i = \left(i - \frac{N_x - 1}{2}\right) s$ and $y_j = \left(j - \frac{N_y - 1}{2}\right) s$ and the discrete deposition kernel is obtained by sampling the continuous model:

$$K_{i,j} = G(x_i, y_j) \tag{20}$$

The discrete deposition kernel is scaled by the peak nebulization flow rate $\dot{C}_{\text{peak}}$ (ppm/s), defining the local concentration deposition rate:

$$\dot{C}(i, j, t) = \dot{C}_{\text{peak}}\, K_{i,j}\, e^{-t/\tau} \tag{21}$$

where τ denotes the effective decay time constant accounting for chemical decomposition and evaporation phenomena and t denotes physical time expressed in seconds. The

cumulative deposited concentration at each spatial cell is obtained by time integration of the deposition rate:

$$C(i,j,t) = \int_0^t \dot{C}(i,j,\xi)\, d\xi. \tag{22}$$

In the numerical implementation, the integration is performed in discrete time using the same sampling interval adopted for the robot motion planner. Although $C(i,j,t)$ does not describe the instantaneous airborne concentration, it represents the cumulative deposited dose resulting from the nebulization process. This formulation captures the dominant spatial–temporal effects of aerosol propagation while enabling the explicit integration of exposure constraints within the motion planning framework.

7.2 Exposure simulation and velocity tuning

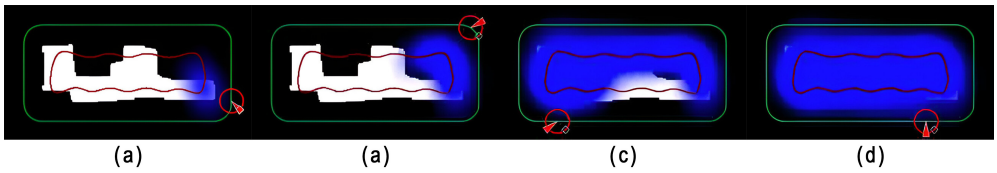


Fig. 7 Disinfection simulation in different step moments based on RMEP: (a) 3.8%, (b) 15.4%, (c) 75%, and (d) 99.9%

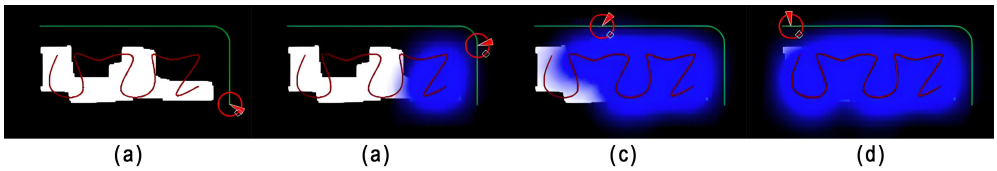


Fig. 8 Disinfection simulation in different step moments based on partial RMEP: (a) 7.5%, (b) 23%, (c) 74.5%, and (d) 98.2%

Given the peroxide aerosol distribution model introduced above, we simulate the nebulization process along the planned trajectory to estimate the spatial distribution of delivered dose on the target and tune the robot velocity to satisfy a minimum exposure requirement. The same target example is shown in Figs. 7 and 8, illustrating intermediate simulation steps for two different trajectory optimizations; different shades of blue represent the progressive accumulation of dose over time. Given the parametrized disinfection trajectory $\mathbf{r}_d(s)$, with $s \in [s_0, s_m]$, the emission position at each time step is $\mathbf{q}(s) = [x_d(s), y_d(s)]$. At each position, the discretized kernel $K_{i,j}$ is placed on the environment grid and scaled according to the deposition rate model $\dot{C}(i,j,t)$, which is integrated in time (Eq. 22). This yields a cumulative exposure (dose

proxy) map $E(\mathbf{q})$ defined on Ω_{target}. To ensure effective disinfection, the exposure must reach at least the minimum required threshold $E_{\min}$:

$$E(\mathbf{q}) \geq E_{\min}, \quad \forall \mathbf{q} \in \Omega_{\text{target}} \tag{23}$$

where $\Omega_{\text{target}} \subset \mathbb{R}^2$ denotes the set of spatial positions corresponding to surfaces that require treatment, discretized for simulation purposes. Given this requirement, the simulation evaluates whether the selected velocity v guarantees adequate exposure. If $E(\mathbf{q}) < E_{\min}$ for any $\mathbf{q} \in \Omega_{\text{target}}$, the speed is changed iteratively using a binary search strategy to identify the maximum admissible velocity that satisfies the exposure criterion and respects speed maximum and minimum constraints.

8 Trajectory execution method

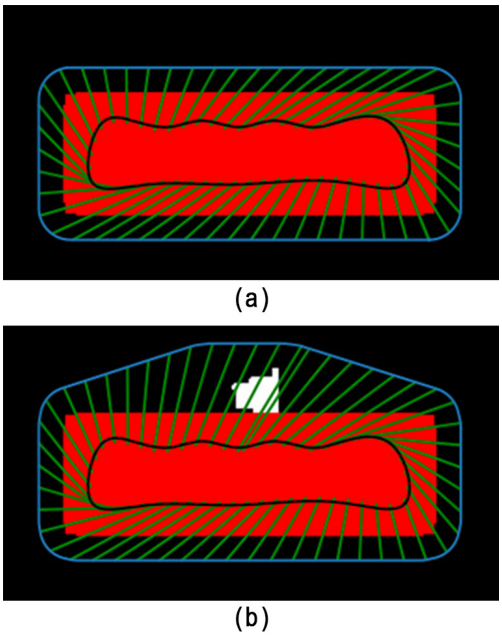


Fig. 9 Example of trajectory adaptation for obstacle avoidance. (a) Planned trajectory based on initial computation. (b) Corrected trajectory after deviation due to an unforeseen obstacle.

In complex environments with many systems to manage, discrepancies in update frequency can arise between the trajectory target and the nebulizer control. While navigation remains largely unaffected, lower update rates for the disinfection trajectory may result in inconsistencies in area coverage or peroxide deposition. To mitigate this challenge, the execution framework employs a decoupled approach, continuously estimating the robot's position $X_r = [x_r, y_r]$ along the navigation trajectory by solving

an optimization problem that minimizes the Euclidean distance:

$$u^* = \arg\min_u |X_n - X_r| \tag{24}$$

Once the corresponding parameter is identified, the disinfection target position is extracted from the same unified B-Spline representation as $X_d(u^*)$. This method guarantees robustness against deviations from the initially planned trajectory, as disinfection targets are dynamically recalculated based on the robot's current position. This mechanism is illustrated in Fig. 9(a), where the planned trajectory is depicted. When deviations occur, as shown in Fig. 9(b), the system compensates by adjusting the nebulization parameters, specifically the direction θ and the power level p, as a function of the robot's current position, its velocity, and the processed target trajectory:

$$P = [\theta, p] = f\left(X_r, \dot{X}_r, X_d(u^*)\right) \tag{25}$$

From a computational perspective, the proposed execution strategy differs from optimization-based replanning approaches commonly adopted in coverage and cleaning robotics. For instance, [28] relies on a receding-horizon optimization framework, in which trajectory updates require repeated solutions of constrained optimization problems, leading to replanning cycles typically on the order of seconds. In contrast, the proposed method does not perform online trajectory optimization. Execution-time adaptation is obtained through geometric spline projection and direct evaluation of the associated disinfection trajectory. Since this update reduces to a nearest-point-on-curve computation based solely on the current robot pose and precomputed spline coefficients, the computational complexity of the online module remains constant and independent of the environment size or coverage resolution. This allows the system to compensate for execution deviations while preserving the structure of the offline-optimized plan and maintaining the required exposure conditions across the target area. Compared to approaches based on learned motion policies for uncertain environments [29] or tree-based next-best-trajectory generation [30], the proposed framework achieves adaptive behavior through an explicit geometric and physical formulation. This results in reduced algorithmic complexity and improved execution predictability, while avoiding the need for stochastic policies or online search-based planning.

9 Experimental tests

This section describes the experimental campaign to identify the parameters of the proposed deposition model and to validate the disinfection framework in representative indoor environments. Hydrogen peroxide concentrations were monitored using proprietary, custom-developed sensors. These devices were cross-validated against a certified reference instrument (PortaSens), ensuring high fidelity in capturing both temporal evolution and peak concentration values.

9.1 Estimation of the Nebulization Pattern Parameters

The parameters characterizing the proposed nebulization model were experimentally identified using dedicated hydrogen peroxide concentration sensors. A sensor grid was

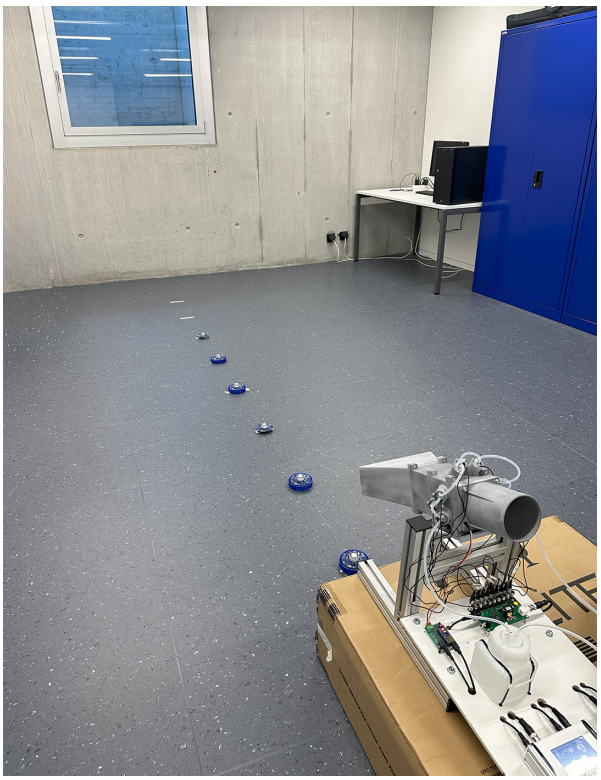


Fig. 10 Experimental setup for the spatial distribution of hydrogen peroxide concentration sensors along the y-axis.

deployed to measure hydrogen peroxide concentrations at various spatial locations, either arranged along the x and y axes, as illustrated in Fig. 10, or distributed in a matrix configuration. By analyzing the temporal evolution of the measured concentrations (Fig. 11) and associating them with the corresponding sensor positions, it is possible to validate the Gaussian deposition model and identify the parameters that best approximate the real deposition pattern. In this specific configuration, the spatial dispersion along the y axis was estimated by determining, for each sensor, the time at which the concentration reached its maximum. The average of these peak times, t_{peak} defines a common reference instant, at which the measured peak concentration values A were fitted with a one-dimensional Gaussian function:

$$G(y) = A \exp\left(-\frac{(y-\mu_y)^2}{2\sigma_y^2}\right) \tag{26}$$

The fitting is performed using a non-linear least-squares method (e.g., *curve_fit* from the SciPy library), allowing the estimation of both the spatial standard deviation σ_y and the mean position μ_y of the distribution. The estimated mean position is not used in the deposition calculations but is instead employed to relate the fan power to the

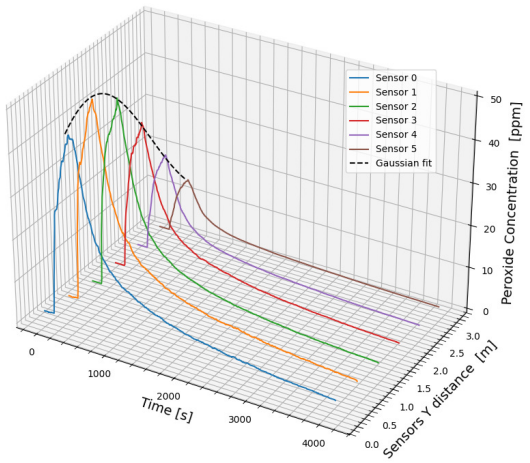


Fig. 11 Temporal evolution of the measured concentration profiles for each sensor.

lateral displacement of the aerosol pattern. Experimental observations confirmed that modulating fan speed induces a controlled shift of the distribution, approximately proportional to the airflow intensity:

$$\mu_y = k_f \cdot P_{\text{fan}} \tag{27}$$

where P_{fan} is the fan power and k_f is an empirical constant determined experimentally. As for the temporal decay constant τ, to estimate it, each concentration curve's post-peak phase was analyzed and fitted with the equation:

$$C(t) = C_{\max} \cdot \exp\left(-\frac{t - t_{\text{peak}}}{\tau}\right), \quad t \geq t_{\text{peak}} \tag{28}$$

Finally, the peak nebulization flow rate $\dot{C}_{\text{peak}}$ can be estimated by integrating the deposition rate over the nebulization interval T_n:

$$C_{\max}(i,j) = \int_0^{T_n} \dot{C}(i,j,t)\, dt = \dot{C}_{\text{peak}}\, K_{i,j}\, \tau \left(1 - e^{-T_n/\tau}\right). \tag{29}$$

It is worth noting that certain parameters, particularly τ, are strongly influenced by environmental factors such as ambient humidity, temperature, and the initial contamination level of the surfaces. As these conditions vary across deployments, the decay behavior may change accordingly.

9.2 Experimental campaign and evaluation protocol

The validation campaign was conducted in a controlled indoor environment (Fig. 12). All experiments were performed in the same room to maintain consistent environmental conditions while systematically varying the disinfection targets, required coverage



Fig. 12 Experimental setup showing the test environment and the nebulization process.

percentages, and obstacle configuration. In total, several tens of experimental runs were executed. Between trials, chairs and small pieces of furniture were repositioned with respect to the original environment map, intentionally introducing deviations between the planned and executed robot trajectories. This procedure allowed the evaluation of the execution-level adaptation mechanism under realistic disturbances, including trajectory deviations and localized airflow perturbations, without modifying the offline planning stage. Four hydrogen peroxide concentration sensors (S1-S4) were placed at representative locations on the target desk to evaluate both target reach and

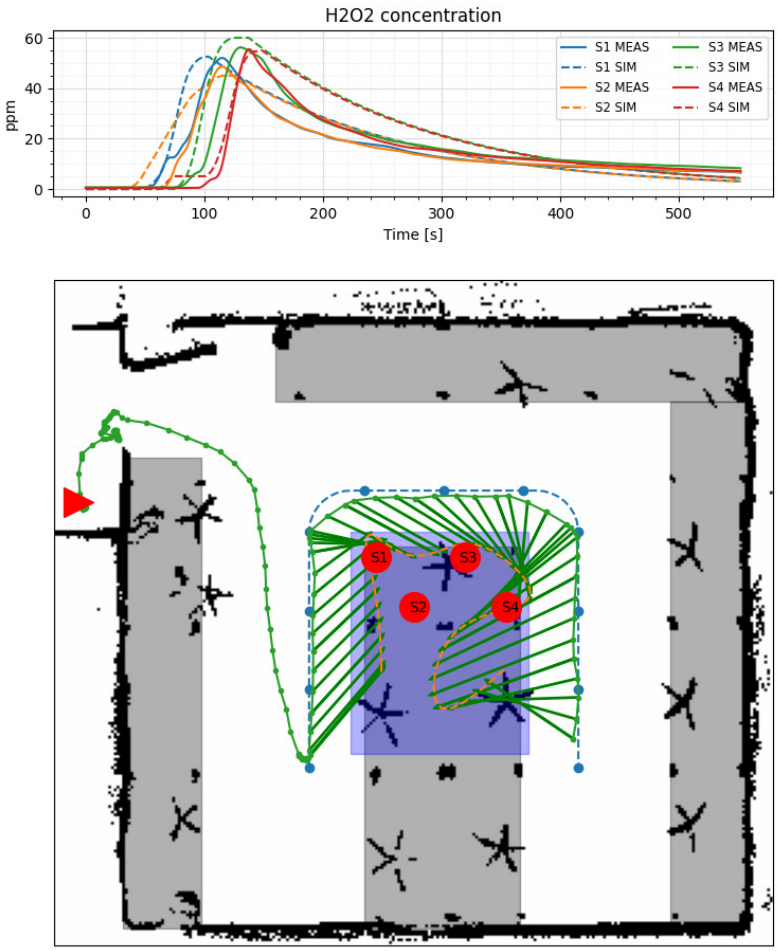


Fig. 13 The top part shows the temporal evolution of both simulated and measured peroxide concentrations at all sensor locations. The bottom part illustrates the planned disinfection trajectory (hatched blue), its discretized navigation points (blue dots), and the robot's path (green), overlaid on the environment map. The expected disinfection coverage is highlighted in solid blue.

spatial homogeneity of the aerosol distribution. The disinfection trajectories were generated offline to achieve predefined exposure thresholds, typically targeting 50 ppm. Two execution modes are compared:

- **System A (brute-force)**: conventional room-saturation strategy implemented with a fixed-position fogging device that releases a prescribed amount of disinfectant to saturate the room air volume. The total delivered liquid is set a priori as a function of the room volume (m^3), following standard dosing practice.

- **System B (proposed)**: targeted strategy, where the robot executes the planned disinfection trajectory and the exposure on the target region is evaluated through the proposed physical model.

The bottom part of Fig. 13 illustrates a representative experimental run for System B, where the discretized navigation waypoints (blue dots) guarantee satisfactory global tracking accuracy. The executed robot trajectory (green line) deviates from the nominal path by up to approximately 30 cm in the presence of obstacles. While such deviations are acceptable for navigation purposes, they are insufficient to guarantee correct nozzle orientation and exposure delivery without continuous correction. This observation motivates the execution-level adaptation strategy introduced in Section 8. The top part of Fig. 13 reports the temporal evolution of both simulated and measured hydrogen peroxide concentrations at all sensor locations. Across repeated trials, the simulated accumulation trends closely match the measured profiles, confirming the capability of the proposed physical model to approximate the dynamic exposure process. Minor discrepancies are mainly attributable to unmodeled local airflow effects and to environmental factors such as humidity, temperature, and reactive consumption of the disinfectant, which influence the decay phase. To quantify coverage effectiveness and spatial uniformity, we report both map-based and sensor-based homogeneity metrics. Let Ω_{target} denote the discretized target surface (e.g., pixels/cells in the exposure map). For each cell $k \in \Omega_{\text{target}}$, let E_k be the simulated cumulative exposure (ppm·s) and let $E_{\min}$ be the required exposure threshold. Let $\mathcal{S}$ denote the sensor set and let c_s be the measured concentration value (ppm) associated with sensor $s \in \mathcal{S}$ (here we use the peak value during the run). We define:

$$\text{PAC} = \frac{|\{k \in \Omega_{\text{target}} : E_k \geq E_{\min}\}|}{|\Omega_{\text{target}}|} \cdot 100 \tag{30}$$

$$\text{PCV} = \frac{\sigma_c}{\mu_c} \cdot 100, \quad \mu_c = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} c_s \tag{31}$$

$$\text{UIDX} = \frac{\min\limits_{s \in \mathcal{S}} c_s}{\max\limits_{s \in \mathcal{S}} c_s} \tag{32}$$

where σ_c is the standard deviation of $\{c_s\}_{s \in \mathcal{S}}$. Here, PCV (percentage coefficient of variation) captures the relative dispersion of the measured concentrations, while UIDX is a unitless uniformity index ranging in $(0, 1]$. For convenience, we also report a combined homogeneity score $\text{HIDX} = \text{PAC} \cdot (100 - \text{PCV})$. Table 1 summarizes the results for the brute-force baseline (System A) and the proposed targeted method (System B). Despite comparable mean concentration levels measured on the target, the two approaches differ substantially in terms of spatial homogeneity and delivery efficiency. System B achieves a markedly lower relative variability (PCV = 5.70%) compared to System A (PCV = 24.84%), resulting in significantly higher uniformity (UIDX = 0.86 vs. 0.44) and overall homogeneity (HIDX = 91.09% vs. 75.01%). While both approaches yield high PAC values (A: expected 99.8%, B: simulated

96.6%), these results highlight that threshold exceedance alone is insufficient to characterize disinfection quality. From an efficiency perspective, the brute-force strategy

Table 1 Comparison between brute-force room saturation (A) and proposed targeted system (B).

Metric	A (brute-force)	B (proposed)
Flow rate [ml/min]	22.2	25.0
Time [s]	281	146
Area [m^2]	40	4
Mean concentration [ppm]	56.14	53.05
Std concentration [ppm]	13.74	3.02
PAC [%]	99.8†	96.6‡
PCV [%]	24.84	5.70
UIDX []	0.44	0.86
HIDX [%]	75.01	91.09
Total liquid [ml]	104.0	60.8
Areal dose [ml/m^2]	2.60	15.2
Coverage rate [m^2/min]	8.54	1.64
PAC/PCV score []	4.3	13.92

†Expected PAC under room-saturation assumption. ‡Simulated PAC from exposure map.

distributes the disinfectant over a large volume, yielding a higher apparent areal throughput (8.54 m^2/min vs. 1.64 m^2/min), but a much lower areal dose (2.60 ml/m^2 vs. 15.2 ml/m^2). The proposed method intentionally concentrates disinfectant delivery within the target region, improving spatial homogeneity and repeatability under execution disturbances at the cost of reduced coverage. From a computational perspective, the proposed execution strategy differs from optimization-based receding-horizon approaches commonly adopted in coverage and cleaning robotics. For instance, the framework presented in [28] formulates trajectory adaptation as a nonlinear optimization problem that must be solved repeatedly online, resulting in a computational load that depends on the adopted solver, the prediction horizon, and the platform's capabilities. In contrast, the proposed method avoids online trajectory re-optimization entirely. Execution-time adaptation is achieved through geometric spline projection and direct evaluation of the associated disinfection trajectory, yielding an online computational cost primarily dependent on spline order and evaluation frequency, and independent of environment size or coverage resolution. While a direct experimental comparison with existing approaches is not feasible due to differences in robotic platforms and disinfection technologies, the obtained results can be contextualized within the broader coverage-planning literature. Optimization-based cleaning strategies often focus on traversal efficiency or energy minimization, for example, through genetic-algorithm formulations operating on planar representations of the environment [31]. Similarly, spline-based trajectory generation methods have been extensively studied to ensure kinematic feasibility, curvature continuity, and speed adaptation within safe motion corridors [32, 33]. In contrast, the present work explicitly integrates chemical–physical exposure constraints into the planning process. Rather than treating disinfection as a purely geometric or kinematic problem, a minimum exposure threshold derived from

a physical deposition model is enforced as a first-class planning constraint, demonstrating that physically grounded disinfection objectives can be achieved within an execution-adaptive trajectory planning architecture.

10 Conclusion

This work presented a framework for autonomous indoor disinfection that integrates physical aerosol modeling with adaptive trajectory planning. The methodology is based on a unified mathematical architecture that decouples the robot's navigation path from the disinfection target trajectory using B-Spline representations. This decoupling, combined with a constrained Traveling Salesman Problem (TSP) formulation, enables independent optimization of robot mobility and spray parameters to maintain a consistent nebulization profile along the path. A key characteristic of this architecture is the reduced dependency of the nebulization consistency on the specific robot trajectory. Since the disinfection requirements are encoded as an independent trajectory component, the system can execute partial navigation paths relative to the maximal enclosing path without compromising the treatment efficacy. This property potentially makes the framework a robust foundation for global multi-target optimization algorithms. In such scenarios, inter-target navigation can be independently optimized for time, distance, or energy efficiency, while the framework ensures that the aerosol deposition profile on each localized target remains consistent and predictable. Experimental results indicate that while volumetric saturation is an effective and common practice for treating the overall environment, the proposed targeted approach can further enhance concentration uniformity on specific surfaces of interest. In our trials, concentration variability was reduced from approximately 25% to 6%, while simultaneously decreasing execution time and disinfectant consumption by nearly 50% and 40%, respectively. Validated on a custom-built robotic platform, the proposed method demonstrates how integrating physical constraints into the planning architecture can contribute to more precise and efficient disinfection processes. Future developments will focus on integrating hydrogen peroxide concentration sensors for real-time feedback and extending the framework to multi-target coordination in complex indoor environments.

Data Availability

The datasets supporting the conclusions of this article have been deposited in a public repository and are available under DOI: 10.5281/zenodo.18672962. The repository includes all experimental time-series, sensor positions, aggregated metrics, environment maps, and target masks used in the analyses reported in the manuscript.

Acknowledgements

This work was co-funded by Innosuisse - Swiss Innovation Agency

References

[1] Chio, I., Ruan, K., Wu, Z., Wong, K.I., Tam, L.M., Xu, Q.: Design and autonomous navigation of a new indoor disinfection robot based on disinfection modeling. IEEE Transactions on Automation Science and Engineering **20**(1), 649–661 (2023) https://doi.org/10.1109/TASE.2022.3165084

[2] Dogru, S., Marques, L.: Path and trajectory planning for uv-c disinfection robots. IEEE Robotics and Automation Letters **8**(7), 4099–4106 (2023) https://doi.org/10.1109/LRA.2023.3280800

[3] Conroy, J., *et al.*: Robot development and path planning for indoor ultraviolet light disinfection. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, pp. 7795–7801 (2021). https://doi.org/10.1109/ICRA48506.2021.9561405

[4] Correia Marques, J.M., Ramalingam, R., Pan, Z., Hauser, K.: Optimized coverage planning for uv surface disinfection. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 9731–9737 (2021)

[5] Ren, H., Qureshi, A.H.: Robot active neural sensing and planning in unknown cluttered environments. IEEE Transactions on Robotics **39**(4), 2738–2750 (2023) https://doi.org/10.1109/TRO.2023.3262114

[6] Sheng, W., Chen, H., Xi, N., Chen, Y.: Tool path planning for compound surfaces in spray forming processes. IEEE Transactions on Automation Science and Engineering **2**(3), 240–249 (2005) https://doi.org/10.1109/TASE.2005.847739

[7] Cerven, W.T., Bullo, F.: Constructive controllability algorithms for motion planning and optimization. IEEE Transactions on Automatic Control **48**(4), 575–589 (2003) https://doi.org/10.1109/TAC.2003.809798

[8] Duncan, S., Jones, P., Wellstead, P.: A frequency-domain approach to determining the path separation for spray coating. IEEE Transactions on Automation Science and Engineering **2**(3), 233–239 (2005) https://doi.org/10.1109/TASE.2005.850393

[9] Agarwal, S., Akella, S.: Line coverage with multiple robots: Algorithms and experiments. IEEE Transactions on Robotics **40**, 1664–1683 (2024) https://doi.org/10.1109/TRO.2024.3355802

[10] Zhang, Y., Chen, X., Feng, C., Zhou, B., Shen, S.: Falcon: Fast autonomous aerial exploration using coverage path guidance. IEEE Transactions on Robotics **41**, 1365–1385 (2025) https://doi.org/10.1109/TRO.2024.3522148

[11] Zhou, B., Xu, H., Shen, S.: Racer: Rapid collaborative exploration with a decentralized multi-uav system. IEEE Transactions on Robotics **39**(3), 1816–1835

(2023) https://doi.org/10.1109/TRO.2023.3236945

[12] Tang, J., Mao, Z., Ma, H.: Large-scale multirobot coverage path planning on grids with path deconfliction. IEEE Transactions on Robotics **41**, 3348–3367 (2025) https://doi.org/10.1109/TRO.2025.3567476

[13] Xu, Y., Lai, S., Li, J., Luo, D., You, Y.: Concurrent optimal trajectory planning for indoor quadrotor formation switching. Journal of Intelligent & Robotic Systems **94**(2), 503–520 (2019) https://doi.org/10.1007/s10846-018-0813-9

[14] Terra, F.P., Nascimento, G.H., Duarte, G.A., Drews-Jr, P.L.J.: Autonomous agricultural sprayer using machine vision and nozzle control. Journal of Intelligent & Robotic Systems **102**(2), 38 (2021) https://doi.org/10.1007/s10846-021-01361-x

[15] Kano, H., Fujioka, H.: B-spline trajectory planning with curvature constraint. In: 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, pp. 1963–1968 (2018). https://doi.org/10.23919/ACC.2018.8431703

[16] Cheng, M., Xu, S.: Research on trajectory planning of six-degree-of-freedom robotic arm based on improved genetic algorithm. Journal of Intelligent & Robotic Systems **111**(3), 105 (2025) https://doi.org/10.1007/s10846-025-02306-4

[17] Berglund, T., Brodnik, A., Jonsson, H., Staffanson, M., Soderkvist, I.: Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles. IEEE Transactions on Automation Science and Engineering **7**(1), 167–172 (2010) https://doi.org/10.1109/TASE.2009.2015886

[18] Riboli, M., Jaccard, M., Silvestri, M., Aimi, A., Malara, C.: Collision-free and smooth motion planning of dual-arm cartesian robot based on b-spline representation. Robotics and Autonomous Systems **170**, 104534 (2023) https://doi.org/10.1016/j.robot.2023.104534

[19] Wang, J., Liu, M., Liao, W., Yun, K., Tan, Y., Zhang, Z.: Spline interpolation method based on arc length parameterization and its application in stress field interpolation for flexible plates. IEEE Access **9**, 35879–35887 (2021) https://doi.org/10.1109/ACCESS.2021.3060453

[20] Wang, H., Kearney, J., Atkinson, K.: Arc-length parameterized spline curves for real-time simulation. In: Curve and Surface Design: Saint-Malo 2002, pp. 387–396. Nashboro Press, Brentwood, TN, USA (2002)

[21] Debrouwere, F., *et al.*: Time-optimal path following for robots with convex–concave constraints using sequential convex programming. IEEE Transactions on Robotics **29**(6), 1485–1495 (2013) https://doi.org/10.1109/TRO.2013.2277565

[22] Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for

maximizing submodular set functions—i. Mathematical Programming **14**, 265–294 (1978) https://doi.org/10.1007/BF01588971

[23] Copelli, S., Barozzi, M., Fumagalli, A., Derudi, M.: Application of a gaussian model to simulate contaminants dispersion in industrial accidents. Chemical Engineering Transactions **77**, 799–804 (2019) https://doi.org/10.3303/CET1977134

[24] Sun, M.M., Gaggar, A., Trautman, P., Murphey, T.: Fast ergodic search with kernel functions. IEEE Transactions on Robotics **41**, 1841–1860 (2025) https://doi.org/10.1109/TRO.2025.3543298

[25] Lelong, N., Vecellio, L., Gélicourt, Y., Tanguy, C., Diot, P., *et al.*: Comparison of numerical simulations to experiments for atomization in a jet nebulizer. PLOS ONE **8**(11), 78659 (2013) https://doi.org/10.1371/journal.pone.0078659

[26] Buan, A., Amparan, J., Natividad, M., Ordes, R., Sierra, M.G., Lopez, E.C.R.: Recent advances in modeling of particle dispersion. In: Engineering Proceedings, vol. 56, p. 332 (2023). https://doi.org/10.3390/ASEC2023-16262

[27] Christakis, N., Drikakis, D.: On particle dispersion statistics using unsupervised learning and gaussian mixture models. Physics of Fluids **36**(9), 093317 (2024) https://doi.org/10.1063/5.0229111

[28] Baglioni, M., Jamshidnejad, A.: A novel mpc formulation for dynamic target tracking with increased area coverage for search-and-rescue robots. Journal of Intelligent & Robotic Systems **110**(4), 140 (2024) https://doi.org/10.1007/s10846-024-02167-3

[29] Tsang, F., Walker, T., MacDonald, R.A., Sadeghi, A., Smith, S.L.: Lamp: Learning a motion policy to repeatedly navigate in an uncertain environment. IEEE Transactions on Robotics **38**(3), 1638–1652 (2022) https://doi.org/10.1109/TRO.2021.3109414

[30] Lindqvist, B., Patel, A., Löfgren, K., Nikolakopoulos, G.: A tree-based next-best-trajectory method for 3-d uav exploration. IEEE Transactions on Robotics **40**, 3496–3513 (2024) https://doi.org/10.1109/TRO.2024.3422052

[31] Batista, V.R., Zampirolli, F.A.: Optimising robotic pool-cleaning with a genetic algorithm. Journal of Intelligent & Robotic Systems **95**(2), 443–458 (2019) https://doi.org/10.1007/s10846-018-0953-y

[32] Wang, L., Guo, Y.: Speed adaptive robot trajectory generation based on derivative property of b-spline curve. IEEE Robotics and Automation Letters **8**(4), 1905–1911 (2023) https://doi.org/10.1109/LRA.2023.3241812

[33] Zeng, Z., Sammut, K., He, F., Lammas, A.: Efficient path evaluation for auvs using adaptive b-spline approximation. In: 2012 Oceans, pp. 1–8 (2012). https:

//doi.org/10.1109/OCEANS.2012.6405066