POLITECNICO DI TORINO
Repository ISTITUZIONALE

Noise-Aware UAV Path Planning in Urban Environment with Reinforcement Learning

Original
Noise-Aware UAV Path Planning in Urban Environment with Reinforcement Learning / Sarhan, S., Rinaldi, M., Primatesta, S., Guglieri, G.. - In: ENGINEERING PROCEEDINGS. - ISSN 2673-4591. - ELETTRONICO. - 90:1(2025), pp. 1-9. [10.3390/engproc2025090003]

Availability:
This version is available at: 11583/2998145 since: 2025-03-07T11:50:46Z

Publisher:
MDPI

Published
DOI:10.3390/engproc2025090003

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

26 November 2026

engineering proceedings

MDPI

Proceeding Paper

Noise-Aware UAV Path Planning in Urban Environment with Reinforcement Learning †

Shahin Sarhan, Marco Rinaldi *, Stefano Primatesta and Giorgio Guglieri

Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

* Correspondence: marco_rinaldi@polito.it

† Presented at the 14th EASN International Conference on "Innovation in Aviation & Space towards sustainability today & tomorrow", Thessaloniki, Greece, 8–11 October 2024.

Abstract: This research presents a comprehensive approach for mitigating noise pollution from Unmanned Aerial Vehicles (UAVs) in urban environment by using Reinforcement Learning (RL) for flight path planning. Focusing on the city of Turin, Italy, the study utilizes its diverse urban architecture to develop a detailed 3D occupancy grid map, and a population density map. A dynamic noise source model adjusts noise emissions based on the UAV velocity, while acoustic ray tracing simulates noise propagation in the environment. The Deep Deterministic Policy Gradient (DDPG) algorithm optimizes flight paths, minimizing the noise impact, and balancing both the path length and the population density located under the UAV path. The simulation results demonstrate significant noise reduction, suggesting scalability and adaptability for global urban environments, contributing to sustainable urban air mobility by addressing noise pollution.

Keywords: UAV; noise mitigation; reinforcement learning; path planning; DDPG; UAM; acoustic ray tracing; drone noise; aerial robotics; sustainable drones; quiet drones; drones and environment

check for updates

Academic Editors: Spiros Pantelakis, Andreas Strohmayer and Nikolaos Michailidis

Published: 7 March 2025

Citation: Sarhan, S.; Rinaldi, M.; Primatesta, S.; Guglieri, G. Noise-Aware UAV Path Planning in Urban Environment with Reinforcement Learning. *Eng. Proc.* **2025**, *90*, 3. https://doi.org/10.3390/engproc2025090003

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The advent of Unmanned Aerial Vehicles (UAVs) has revolutionized many sectors including surveillance, search and rescue, delivery [1,2], environmental monitoring, etc. [3]. Urban Air Mobility (UAM) aims to utilize the urban airspace for efficient transportation of goods and passengers, thereby reducing ground traffic congestion. However, public acceptance of UAVs is hindered by concerns over privacy, safety, and noise pollution [4]. Addressing UAV noise is crucial for the successful implementation and acceptance of UAM services. Mitigation strategies encompass structural modifications, such as sound-proofing, the use of porous materials [5] and biologically inspired blade designs, as well as active noise cancellation techniques [6] employing digital signal processing.

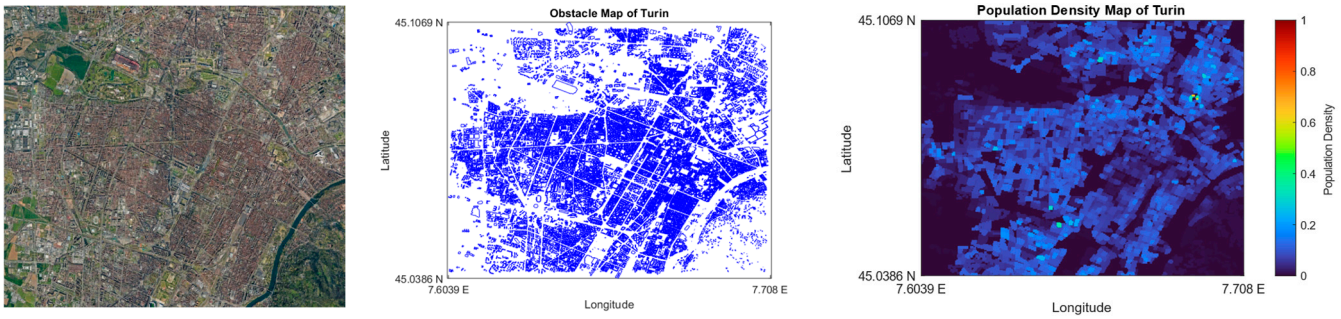
Optimizing UAV flight paths is another strategy for reducing the UAV noise impact. One study utilizes the "iNoise" software to simulate and calculate noise, proposing flight paths that minimize the noise impact [7]. Another study uses the A* algorithm to compute various paths with different levels of "quietness", overlaying a grid to designate obstacles and "quiet zones" [8]. Agent-based modelling is employed in [9] to guide the UAV away from noisy, heavily used paths. The work in [10] integrates Gaussian beam tracing and the A* algorithm with a noise-based objective function to mitigate urban noise pollution from UAVs. A method introduced in [11] leverages a noise assessment platform to generate noise maps and uses an improved cost-based A* algorithm to find optimal UAV flight paths

with minimal sound exposure. Additionally, authors in [12] exploit a simulated annealing algorithm to balance noise reduction and energy efficiency, prioritizing noise abatement in residential zones by incorporating noise sensitivity levels. This work focuses on the development of a noise-aware UAV path planning strategy in urban environment by means of a Reinforcement Learning (RL). This paper contributes to the development of sustainable UAM by addressing the critical issue of noise pollution in populated environments, enhancing the potential for UAV operations to be publicly acceptable and regulatory compliant. Section 2 describes the methodology employed in the study, divided into several key components: creation of obstacles and population density maps for the study area, modelling of the noise source, modeling of noise propagation using ray tracing, and development of a RL-based approach for noise-aware UAV flight path planning. The implementation details are presented in Section 3. In Section 4, the model is validated by means of simulation results with maps unseen during training, and a discussion is proposed.

2. Noise-Aware UAV Path Planning in Urban Environment

2.1. Obstacles and Population Density Maps

Buildings' location and dimension data is sourced from OpenStreetMap (OSM) for constructing a 3D occupancy grid of Turin, Italy. The study area is defined within the longitude range of [7.6039°, 7.708°] E, and the latitude range of [45.0386°, 45.1069°] N. Building heights are randomly assigned values between 12 m and 18 m. The area was represented by a 3D tensor, with each point indicating occupancy (1) or vacancy (0). Population density data is obtained from Meta, filtered to match the coordinates of the obstacles map. Figure 1 shows the satellite picture of the area as well as the obstacle map and the population density map. Both the maps were subdivided into manageable maps, each 200 × 200 cells, enabling consistent training and simulation. Each cell has a length of 2 m.



(**a**) (**b**) (**c**)

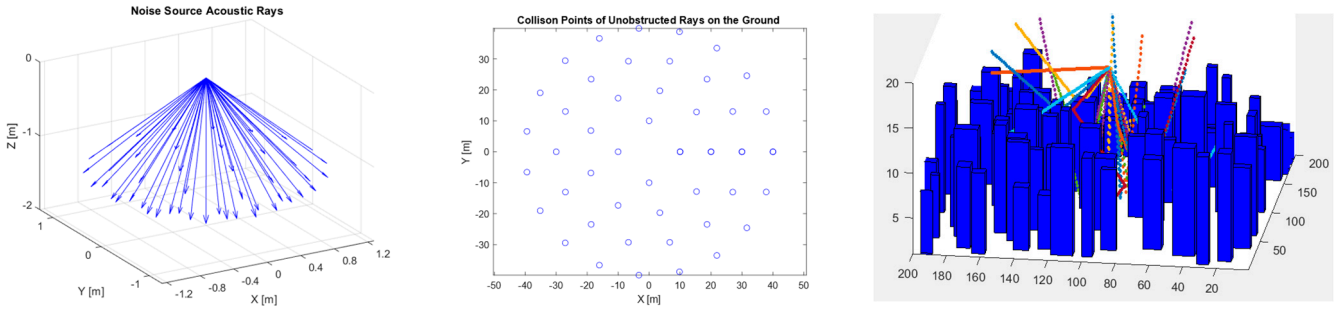
Figure 1. The geographical location of the area of study (Turin, Italy): (**a**) satellite image of the study area; (**b**) obstacle map of the study area; (**c**) population density map of the study area.

2.2. Noise Source Modelling

A point-source spherical propagation model is employed to simulate the UAV noise. This model simplifies calculations while remaining representative of established findings. Although multi-rotor UAVs produce highly directional noise in their immediate vicinity, it's assumed that the noise propagates isotropically at the distances considered in this study. Consequently, as the UAV traverse its routes, the noise pattern generated on the ground does not distort when turning.

Typically, a constant noise source is assumed in such models; however, the noise levels of the drone is correlated with its velocity, and since the RL agent can adapt its velocity, it is beneficial to incorporate this variable into the noise model. Since Sound Pressure Level

(SPL) at a distance of 1 m from the 'DJI Inspire 2' quadrotor was measured at speeds of 5, 10, and 20 km/h in [13], an exponential curve is fit to these data points. This results in $L_0 = 60.24 \cdot \exp(0.003379 \cdot v)$ which allows dynamic noise level adjustment based on UAV velocity during flight. L_0 denotes the source SPL (in dB), and v denotes the velocity (in km/h). The apparently overly simplified drone's noise model introduced by the relation between L_0 and v can be re-addressed quite easily in the future by interpolating data resulting from extensive experimental results and/or introducing new parameters and functions for the assessment of the impact of drone-generated noise on people based on the operating conditions [14]. In the context of minimizing UAV noise impact, accurately modelling noise propagation is critical. This process involves generating acoustic rays emitted from a noise source, which are utilized in the ray tracing algorithm. The source altitude is set to 40 m and direction vectors from the source to 46 predefined ground points are computed and normalized to unit vectors, as illustrated in Figure 2.



(**a**) (**b**) (**c**)

Figure 2. The noise source: (**a**) the acoustic rays emitted from the source at an altitude of 40 m; (**b**) collision points of the unobstructed rays on the ground. (**c**) Synthetic representation of the scenario with buildings and the consequent reflection of acoustic rays.

2.3. Acoustic Ray Tracing

The ray tracing algorithm models UAV noise propagation considering geometric divergence, atmospheric absorption, and reflections. The assumptions include homogeneous medium properties such as temperature, pressure and relative humidity, linear propagation, specular reflection, and negligible diffraction. The algorithm initializes arrays for each ray, propagating them through the medium and updating their positions based on the direction. Reflections off surfaces are assumed to follow the law of specular reflection, where the angle of incidence equals the angle of reflection. The SPL attenuations due to geometric divergence A_{div} and atmospheric absorption A_{atm} are aligned with ISO 9613-2 standards [15], and these attenuations are subtracted from the source SPL. The overall SPL at a height of 2 m, L_{tot}, is then computed by summing contributions from all the rays, providing a comprehensive method for evaluating ground-level noise impact. A_{div}, A_{atm}, and L_{tot} are compued as $A_{div} = 20\log\left(\frac{d}{d_0}\right) + 11$, $A_{atm} = \frac{\alpha d}{1000}$, and $L_{tot} = 10\log\left(\sum_i^N 10^{\frac{L_i}{10}}\right)$. In particular, d is the distance travelled from the source (in m), d_0 is the reference distance (d_0 = 1 m); L_i is the attenuated SPL at the ground for each ray, and α is the coefficient of atmospheric attenuation (in dB/km). Given a temperature of 10 degrees Celsius, Relative Humidity (RH) of 70%, and assuming a nominal median frequency of 200 Hz (drones typically operate between 100–300 Hz), the atmospheric absorption coefficient, α, is set to 0.76 dB/km as per ISO 9613-2. The effect of wind on sound propagation is neglected since it depends on frequencies and distances, and established models currently lack in the literature [16]. This issue may be addressed in future works.

2.4. Reinforcement Learning

RL is a paradigm of Machine Learning (ML) that trains an agent to accomplish tasks in uncertain environments. At each discrete time step, the agent receives observations and rewards from the environment, and sends actions back. The reward provides immediate feedback on the success of the agent's previous action relative to the task's goal. An RL agent consists of two main components: a policy and a learning algorithm. The policy maps current environment observations to a probability distribution over possible actions, implemented by a function approximator with tuneable parameters. The learning algorithm continuously updates the policy parameters based on the agent's actions, observations, and received rewards. RL is typically modelled as a Markov Decision Process (MDP). The primary objective of the agent is to maximize the return, G, which is defined as the expected sum of the discounted future rewards: $G(t) = E\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\right]$; where γ is the discount factor, and R is the reward. The agent gradually updates its policy, π, towards the optimal policy, π^*, via action-value methods or policy gradient methods, which are the two principal approaches. The detriment of sample efficiency of policy-based approaches can be improved by combination with value-based methods. For this reason, the DDPG algorithm [17] has been implemented in this work.

2.5. Deep Deterministic Policy Gradient (DDPG)

The DDPG algorithm is a model-free, off-policy RL method that concurrently learns a Q-function and a policy, optimizing both using off-policy data and the Bellman equation. DDPG utilizes an actor and a critic. The actor represents a deterministic policy that maps states to actions, and the critic evaluates the action-value function.

During training, the agent stores past experiences in a replay buffer, and mini-batches of experiences are sampled from this buffer to update the critic by minimizing the mean squared error between the predicted and target Q-values. The actor is updated using the policy gradient, which involves calculating the gradient of the Q-value with respect to the action, and adjusting the policy parameters to increase the expected return. The DDPG algorithm maintains four function approximators to estimate the policy and the value functions: Actor $\pi(S;\theta)$ takes observation S as input and returns the corresponding action that maximizes the long-term reward; Critic $Q(S, A;\phi)$ takes observation S and action A as inputs, and returns the expected long-term reward; Target Actor $\pi_t(S;\theta_t)$ and Target Critic $Q_t(S, A;\phi_t)$ are both periodically updated using the latest actor and critic parameters (θ and ϕ) to maintain stability.

DDPG trains a deterministic policy using an off-policy approach. Since the policy is deterministic, on-policy exploration at the start may not cover a sufficiently diverse range of actions to obtain useful learning signals. To enhance exploration, noise is added to actions during training using the Ornstein-Uhlenbeck noise model. The noise scale can be reduced over the course of training using a decaying factor to balance the exploration-exploitation tradeoff. The noise value $v(t)$ at each time step t is updated as $v(t+1) = v(t) + Mean_{ac} \cdot (Mean - v(t)) \cdot T + std(t) \cdot \mathcal{N}(0,1) \cdot \sqrt{T}$, the standard deviation decays at each time step as $std(t+1) = std(t) \cdot (1 - Decay\ Rate)$. $Mean_{ac}$ is the mean attraction constant which specifies how quickly the noise model output is attracted to the mean, $Mean$ is the mean of the noise value, std is the standard deviation, T is the sampling time and $\mathcal{N}(0,1)$ represents a normal random variable.

3. Implementing the RL Framework

The MATLAB's 2024 Deep Learning and Reinforcement Learning Toolboxes facilitated the development, training, and testing of the DDPG agent in a structured environment. The training setup involves configuring the environment, and initializing both the actor

and the critic neural networks. Some of the main hyperparameters meticulously tuned to optimize the training process are reported in Table 1. Also, $Mean_{ac} = 0.15$, $Mean = 0$, $std = 0.85$, and $Decay\ Rate = 10^{-4}$. The Adam algorithm is used for optimization, with gradient clipping to mitigate the exploding gradient problem.

Table 1. Key tuned hyperparameters associated with the algorithm, and the scaling factors of the reward function of the RL model.

	Hyperparameter	Value
DDPG Algorithm	Mini batch size	128
	Discount factor γ	0.99
	Smoothing factor τ	10^{-3}
Actor/Critic	Learning rate α	10^{-5}
	Number of hidden layers	3
	Number of nodes in each hidden layer	256
Reward Function	$\lambda_1,\ \lambda_2,\ \lambda_3,\ \lambda_4,\ \lambda_5,\ \lambda_6$	0.02, 0.07, 0.43, 0.01, 0.47, 1

During training, a random obstacle and a population density map are selected at the beginning of each episode, and the agent's starting and target positions are specified randomly. Training proceeds with the episode termination condition being either the agent reaching the target within a radius of 2 m or exhausting the maximum number of episode steps (280). The agent is trained until converging to the optimal policy by monitoring the learning curves.

3.1. State Representation and Action Space

The state representation is a critical component in the RL approach, providing a detailed snapshot of the drone's current situation to the neural network. The state is represented as a 24-dimensional vector, encapsulating various features of the drone and its environment such as: cartesian coordinates of the drone and the target in the XY plane, distance from the drone to the target, orientation of the target with respect to the drone represented as an angle in the XY plane, population density at the drone's current location, mean population density across two sub grids (15×15 and 51×51 cells) centred on the drone, orientation of the maximum population density in the sub grid with respect to the drone, overall SPL value on the ground, distances and orientations of the three closest and three furthest obstacles. To ensure stability during training, each of these values is normalized to the range [0, 1]. Normalizing input features ensures balanced gradients, and improves the generalization capability of the RL agent.

The agent operates within a fixed height airspace, navigating in any direction across the continuous XY plane. The sampling time, T, for the agent is set to 0.3 s, balancing decision frequency and computational efficiency. The range of movement at each time step, calibrated to match the maximum velocity of commercial drones (approximately 101 km/h), is set to 6 m for both X and Y directions.

3.2. Reward Function

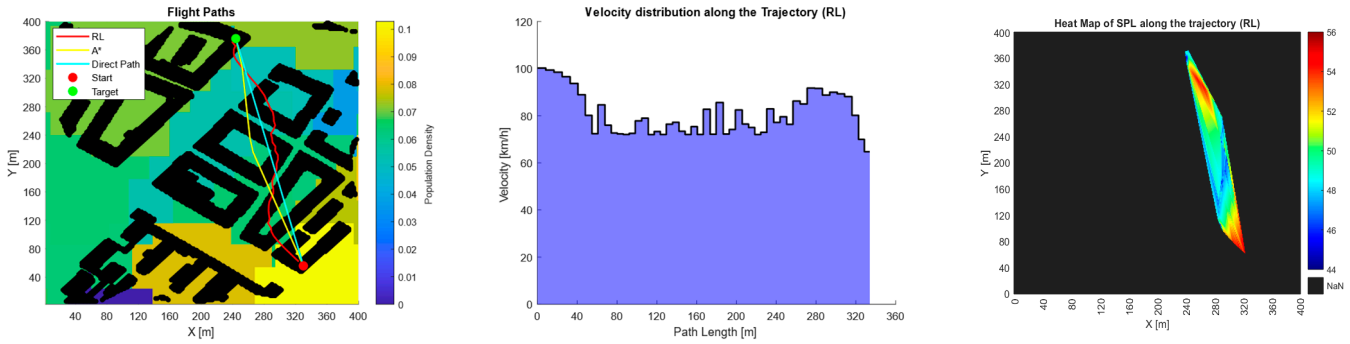
The reward function plays a pivotal role in achieving the desired behavior from the model. It is crucial to set the components of the reward function to guide it towards achieving optimal path and policy. At any timestep, if the episode terminates (the agent reaches the target), a reward of 2 is given. Otherwise, the reward is calculated by the weighted sum of the following terms, which are all individually normalized in the range of $[0, -1]$:

- Idle penalty $P_{Idle}(t)$: penalizes the drone for remaining stationary. A penalty of -1 is applied if the drone's position has not changed in between timesteps.
- Distance penalty $P_{Distance}(t)$: encourages the drone to reduce the distance to the target. The penalty is based on the change in Euclidean distance to the target.
- Population density penalty $P_{Density}(t)$: discourages navigation through high-density areas. A penalty of -1 is applied if the population density increases from the previous timestep; otherwise, a smaller penalty of -0.1 is applied.
- Noise penalty $P_{Noise}(t)$: penalizes high noise levels. If the SPL exceeds the threshold (maximum possible SPL on the ground), a penalty of -1 is applied. Below the threshold, the penalty decreases exponentially to 0 as the SPL decreases to 0.
- Cumulative noise penalty $P_{Cumu_Noise}(t)$: encourages lower cumulative noise and, indirectly, shorter flight time. The penalty increases based on the cumulative noise up to the current timestep.
- Smoothness penalty $P_{Smooth}(t)$: prevents erratic behaviour and encourages smooth navigation. The penalty is based on the angular difference in the movement vector between consecutive timesteps.

The total reward at each timestep, $R(t)$, is calculated as $R(t) = \lambda_1 P_{Idle}(t) + \lambda_2 P_{Distance}(t) + \lambda_3 P_{Density}(t) + \lambda_4 P_{Noise}(t) + \lambda_5 P_{Cumu_noise}(t) + \lambda_6 P_{Smooth}(t)$, where λ is the scaling factor.

4. Results and Discussion

In order to validate the accuracy of the agent's learned policy, the trained agent is tested on a map different from the testing dataset. To compare its performance, this trajectory is compared with the direct path to the target and the optimal path found from the A* algorithm. The heuristic function in the A* algorithm is configured to account for both distance (40%) and population density (60%). Since the noise model is velocity-dependent, defining a velocity for the alternative methods is essential to evaluate their noise impact accurately. For a fair comparison, the average velocity of the RL path is used as a constant velocity along the entire path computed by the other methods. The flight paths are depicted in Figure 3, and the results are tabulated in Table 2.



(**a**) (**b**) (**c**)

Figure 3. (**a**) Flight paths for the RL, A* and direct paths. (**b**)Velocity distribution along the RL flight path. (**c**) Heat Map of the SPL distribution along the RL path.

The RL agent follows a trajectory of 356.2 m, which is 6.98% and 6.02% longer than the direct and the A* paths, respectively. Despite its longer path, the RL agent achieves the lowest SPL per unit length, indicating a prioritization of noise reduction over travel efficiency. The RL agent records the lowest minimum SPL level among the paths at 44.76 dB. This ~2.7 dB reduction demonstrates the agent's ability to effectively reduce noise in specific segments of its path, which is valuable for noise-sensitive environments. However,

the RL agent's maximum SPL is the highest at 56.13 dB. This suggests that the agent dynamically adjusts its speed, accelerating in less sensitive areas and slowing down in noise-sensitive ones, balancing time and noise constraints. The total SPL for the RL agent is the highest at 17,652.2 dB, reflecting its longer path. Nonetheless, the RL agent excels in noise management with the lowest SPL per unit length at 49.55 dB/m, indicating an efficient noise distribution along its trajectory. The SPL distribution along the flight paths and the noise impact in the environment for the 3 methods are shown in Figure 4.

Table 2. Performance metrics of the test for the 3 methods along with the comparison of the RL model's performance with respect to both the A* and the direct paths.

Path	Path Length (m)	Average Velocity (km/h)	Minimum SPL (dB)	Maximum SPL (dB)	Total SPL (dB)	SPL per Unit Length (dB)
Direct	332.97	84.04	47.89	52.87	16,782.8	50.41
A*	335.99	84.04	47.47	52.8	16,974.2	50.52
RL	356.2	84.04	44.76	56.13	17,652.2	49.55
RL vs. Direct	6.98%	0%	−6.54%	6.17%	5.18%	−1.71%
RL vs. A*	6.02%	0%	−6.02%	6.31%	5.11%	−1.92%

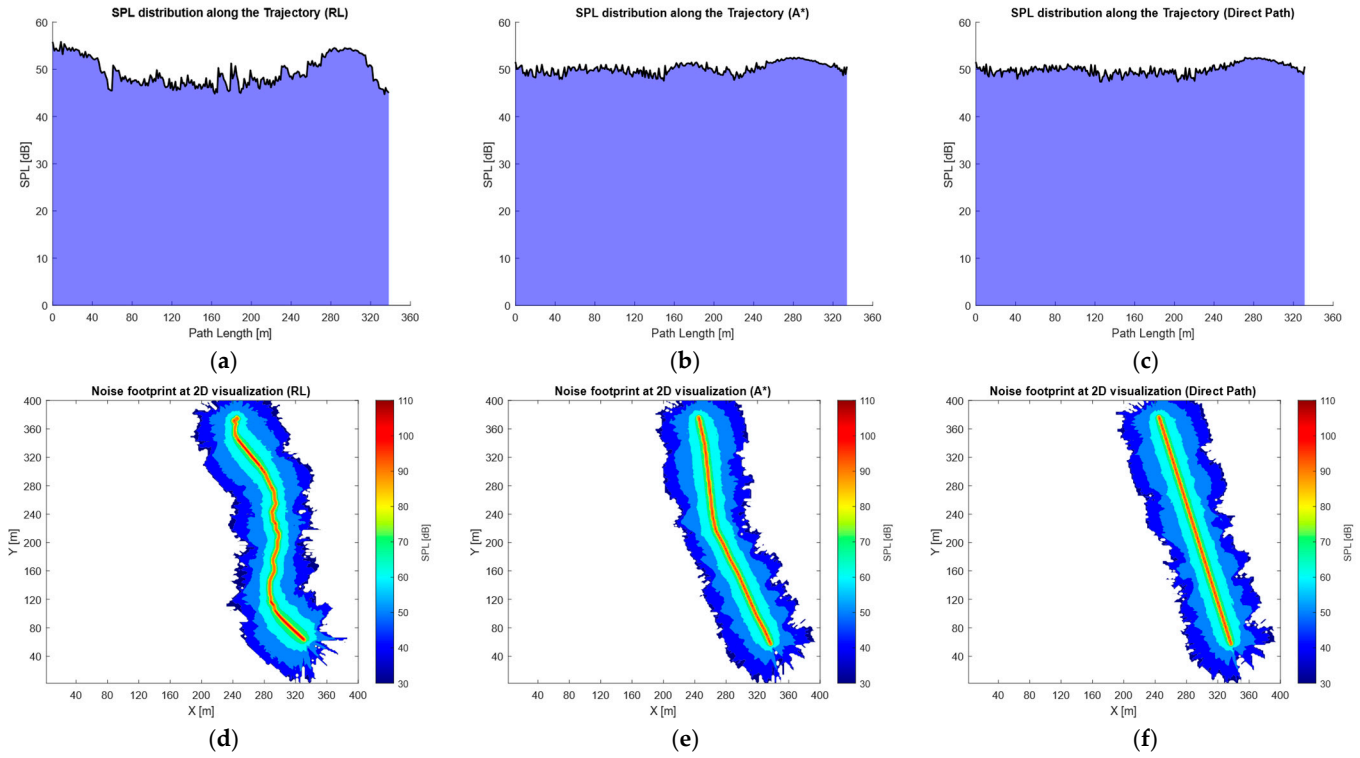


Figure 4. (**a**) SPL distribution along the RL path; (**b**) SPL distribution along the A* path; (**c**) SPL distribution along the direct path; (**d**) Top view of noise impact in the environment for RL path; (**e**) Top view of noise impact in the environment for A* path; (**f**) Top view of noise impact in the environment for direct path.

The comparative analysis reveals that while the RL model generates a longer path and higher overall noise levels, it minimizes SPL per unit length and achieves lower minimum SPL values. The increased path length suggests a deliberate trade-off to prioritize noise reduction. The higher maximum SPL indicates occasional noise peaks due to dynamic velocity adjustments. These findings highlight the RL model's potential for minimizing noise pollution in noise-sensitive environments, despite not always producing the shortest

or fastest path. To rigorously evaluate the generalization capability of the RL model, 20 tests were conducted on ten different maps, each featuring random start and target locations. The statistical analysis, presented as the median and Standard Deviation (Std) values for each performance metric are reported in the Table 3. The validation results confirm that the RL model demonstrates strong generalization capability as well as accuracy across diverse map layouts.

Table 3. Median and standard deviations of the performance metrics for the 3 methods.

Path	Path Length (m)		Minimum SPL (dB)		Maximum SPL (dB)		Total SPL (dB)		SPL per Unit Length (dB)	
	Median	*Std*	*Median*	*Std*	*Median*	*Std*	*Median*	*Std*	*Median*	*Std*
Direct	295.44	122.1	47.89	3.82	53.19	2.58	14,925.4	6446	52.9	2.91
A*	336	128.28	47.47	3.65	54.19	2.7	16,588.8	6794.9	52.78	2.73
RL	333.18	138.8	45.57	3.88	56.61	1.91	17,625.4	7363.5	51.77	2.97
RL vs. Direct	12.78%	13.68%	−4.84%	1.68%	6.44%	−25.97%	18.09%	14.2%	−2.14%	2.13%
RL vs. A*	−0.8%	8.2%	−4.01%	6.24%	4.49%	−29.26%	6.25%	8.37%	−1.92%	8.84%

Author Contributions: Conceptualization, M.R. and S.P.; methodology, M.R. and S.P.; software, S.S.; validation, S.S., M.R. and S.P.; formal analysis, S.S. and M.R.; investigation, S.S. and M.R.; resources, S.S.; data curation, S.S.; writing—original draft preparation, S.S. and M.R.; writing—review and editing, S.P. and G.G.; visualization, S.S.; supervision, G.G.; project administration, G.G.; funding acquisition, G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in the current study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors wish to thank the Piedmont Aerospace Cluster for supporting the activity of Marco Rinaldi.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rinaldi, M.; Primatesta, S. Comprehensive Task Optimization Architecture for Urban UAV-Based Intelligent Transportation System. *Drones* **2024**, *8*, 473. [CrossRef]
2. Rinaldi, M.; Primatesta, S.; Bugaj, M.; Rostáš, J.; Guglieri, G. Urban Air Logistics with Unmanned Aerial Vehicles (UAVs): Double-Chromosome Genetic Task Scheduling with Safe Route Planning. *Smart Cities* **2024**, *7*, 2842–2860. [CrossRef]
3. Rinaldi, M.; Wang, S.; Geronel, R.S.; Primatesta, S. Application of Task Allocation Algorithms in Multi-UAV Intelligent Transportation Systems: A Critical Review. *Big Data Cogn. Comput.* **2024**, *8*, 177. [CrossRef]
4. Panov, I.; Ul Haq, A. A Critical Review of Information Provision for U-Space Traffic Autonomous Guidance. *Aerospace* **2024**, *11*, 471. [CrossRef]
5. Candeloro, P.; Ragni, D.; Pagliaroli, T. Small-Scale Rotor Aeroacoustics for Drone Propulsion: A Review of Noise Sources and Control Strategies. *Fluids* **2022**, *7*, 279. [CrossRef]
6. Gupta, P. Different Techniques of Secondary Path Modeling for Active Noise Control System: A Review. *Int. J. Eng. Res. Technol.* **2016**, *5*, 611–616.
7. Kennedy, J.; Garruccio, S.; Cussen, K. Modelling and mitigation of drone noise. *Vibroengineering Procedia* **2021**, *37*, 60–65. [CrossRef]
8. Adlakha, R.; Liu, W.; Chowdhury, S.; Zheng, M.; Nouh, M. Integration of acoustic compliance and noise mitigation in path planning for drones in human–robot collaborative environments. *J. Vib. Control* **2023**, *29*, 4757–4771. [CrossRef]

9. Šiljak, H.; Kennedy, J.; Byrne, S.; Einicke, K. Noise mitigation of UAV operations through a Complex Networks approach. In Proceedings of the 51st International Congress and Exposition on Noise Control Engineering (Inter·Noise 2022), Glasgow, UK, 21–24 August 2022.
10. Tan, Q.; Li, Y.; Wu, H.; Zhou, P.; Lo, H.K.; Zhong, S.; Zhang, X. Enhancing sustainable urban air transportation: Low-noise UAS flight planning using noise assessment simulator. *Aerosp. Sci. Technol.* **2024**, *147*, 109071. [CrossRef]
11. Tan, Q.; Zhong, S.; Qu, R.; Li, Y.; Zhou, P.; Lo, H.K.; Zhang, X. Low-Noise Flight Path Planning of Drones Based on a Virtual Flight Noise Simulator: A Vehicle Routing Problem. *IEEE Intell. Transp. Syst. Mag.* **2024**, *16*, 56–71. [CrossRef]
12. Scozzaro, G.; Delahaye, D.; Vela, A.E. Noise Abatement Trajectories for a UAV Delivery Fleet. In Proceedings of the 9th SESAR Innovation Days (SID 2019), Athenes, Greece, 2–6 December 2019.
13. Škultéty, F.; Bujna, E.; Janovec, M.; Kandera, B. Noise Impact Assessment of UAS Operation in Urbanised Areas: Field Measurements and a Simulation. *Drones* **2023**, *7*, 314. [CrossRef]
14. Kawai, C.; Jäggi, J.; Georgiou, F.; Meister, J.; Pieren, R.; Schäffer, B. How annoying are drones? A laboratory study with different drone sizes, maneuvers, and speeds. In Proceedings of the 2024 Quiet Drones Conference, Manchester, UK, 8–11 September 2024.
15. *ISO 9613-2:2024*; Acoustics—Attenuation of Sound during Propagation Outdoors—Part 2: Engineering Method for the Prediction of Sound Pressure Levels Outdoors. International Standards Organisation: Geneva, Switzerland, 2024.
16. Trikootam, S.C.; Hornikx, M. The wind effect on sound propagation over urban areas: Experimental approach with an uncontrolled sound source. *Build. Environ.* **2019**, *149*, 561–570. [CrossRef]
17. Silver, D.; Lever, G.; Heess, N.M.; Degris, T.; Wierstra, D.; Riedmiller, M.A. Deterministic Policy Gradient Algorithms. In Proceedings of the 31st International Conference on Machine Learning (ICML 2014), Beijing, China, 21–26 June 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.