SUMO User Conference 2025

Conference paper

https://doi.org/10.52825/scp.v6i.2610

Published: 15 Jul. 2025

SUMO-UAV-Py: A SUMO Plugin for UAV-Based Road Traffic Sensing

Charalambos Tsioutis[1,*], Christos Makridis[1,2], and Stelios Timotheou[1,2]

[1]KIOS Center of Excellence, University of Cyprus, Cyprus

[2]Department of Electrical and Computer Engineering, University of Cyprus, Cyprus

*Correspondence: Charalambos Tsioutis

Abstract. In recent years, Unmanned Aerial Vehicles (UAVs) have emerged as effective tools for traffic monitoring and control by offering high-resolution, aerial observations of vehicular movement. Although UAV simulation is well established, tools to capture microscopic traffic measurements from UAV-based observations remain limited. This paper introduces SUMO-UAV-Py, an open-source SUMO plugin that integrates UAV-based sensing into microscopic traffic simulations in Python. SUMO-UAV-Py captures detailed vehicle observations by dynamically employing multiple UAVs to observe traffic measurements based on their position and field-of-view (FoV). Performance evaluations on a mid-sized network demonstrate that SUMO-UAV-Py maintains simulation performance comparable to standard post-processing methods, confirming its suitability for large-scale traffic monitoring research.

Keywords: UAV-Based Sensing, Open-Source Tools, Traffic Monitoring

1. Introduction

Traffic simulation is an essential tool for analyzing and improving modern transportation systems. As urban environments grow more complex, the need for accurate and scalable traffic monitoring solutions has become increasingly evident. Among emerging sensing technologies, Unmanned Aerial Vehicles (UAVs) have gained significant attention for their ability to monitor road networks from an aerial perspective using onboard cameras. UAV-based sensing has been widely explored for a variety of transportation-related applications, including traffic monitoring, control, and data collection [1], [2], [3]. However, despite their potential, conducting real-world UAV experiments can be challenging due to operational constrains and regulatory restrictions, making simulation-based approaches an attractive alternative for the development and validation of UAV-based traffic sensing.

UAV-based microscopic simulation can provide fine-grained traffic data that many existing studies currently lack. For instance, some works on UAV-based traffic state estimation rely solely on macroscopic measurements (e.g., average density and speed) [4], [5], thus missing the opportunity to capture individual vehicle interactions. Such

efforts could benefit from a UAV-based microscopic framework, which enables higher-resolution data collection for more accurate validation in traffic monitoring approaches.

Despite the growing body of work on UAV-based traffic estimation, few traffic simulators frameworks explicitly focus on integrating UAV sensing at a microscopic level within large-scale transportation simulations. SUMO (*Simulation of Urban MObility*)[6] is a well-established, open-source traffic simulator designed for microscopic modeling of complex road networks. Its open-source nature is potentially suitable for UAV research, as it enables customizable data extraction and manipulation through tools like the TraCI interface.

To effectively enable UAV operations, in SUMO, we present a new open-source plugin, SUMO-UAV-Py, that extends SUMO's capabilities to incorporate UAV-based sensing. The plugin's goal is to simulate the aerial vantage point and data collection processes of UAV traffic monitoring in an efficient and flexible manner. While SUMO-UAV-Py does not explicitly focus on high-fidelity UAV flight physics, its modular architecture allows the integration of external modules or extended libraries to accommodate advanced aerodynamic and environmental models. The plugin enables multiple UAVs to be freely positioned above the road network, collecting detailed vehicle-level information and thus allowing researchers to evaluate the performance of UAV-based algorithms before deploying them in real-world scenarios.

Our contributions in this paper include the development of SUMO-UAV-Py, an open-source and lightweight plugin for SUMO that enables UAV-based microscopic traffic observations. SUMO-UAV-Py offers a flexible, modular framework that supports dynamic UAV positioning while capturing detailed vehicle-level information during simulations, providing a robust platform for aerial traffic monitoring.

The remainder of this paper is organized as follows. Section 2 reviews existing software and simulation frameworks that combine UAVs and traffic networks, highlighting the gap addressed by our proposed plugin. Section 3 details the architecture and design principles of SUMO-UAV-Py, while Section 4 presents the plugin's user options and features. Section 5 presents an experimental use case and performance results. Finally, Section 6 concludes the paper and outlines future work directions.

2. Related Work

A variety of simulation platforms and co-simulation strategies have been proposed to study UAVs in either autonomous navigation or broader traffic management contexts. High-fidelity environments like AirSim [7] leverage the Unreal Engine to simulate realistic visual and physical conditions for autonomous vehicles, while other environments couple SUMO and AirSim to explore lane-based Unmanned Traffic Management (UTM) concepts [8]. Similarly, UTSim [9] targets UAV air traffic control and communication aspects, and commercial traffic simulators such as PTV Vissim have been adapted for 3D UAV visualization [10].

In the context of open-source UAV simulation, UavSim [11] provides a lightweight platform for comparing multi-UAV path planning algorithms, and [12] introduces a modular UAV simulation framework that provides kinematic and energy models of multiple UAVs. Another study [13], explores UAV swarm navigation in urban environments, using sensing for velocity and position estimation within the swarm while [14] introduces a co-simulation framework for UAV physics and wireless communication modeling. In contrast, the work in [15] aims to simulate a swarm of drones for traffic monitoring in the context of a Smart City developed in Unity game engine.

While these tools and frameworks provide robust solutions for UAV flight dynamics, swarm coordination, or communication modeling, they generally do not prioritize or fully implement UAV-based ground traffic sensing at a vehicle-level scale. In contrast, SUMO-UAV-Py aims to fill this gap by integrating a lightweight, open-source extension for a microscopic UAV-based traffic observation.

In addition to the UAV simulation frameworks discussed above, several established microscopic traffic simulators are widely used for modeling urban traffic. Commercial products such as Aimsun provide detailed microscopic simulation and analysis, including support for pedestrians, bicycles, and vehicles exhibiting non-lane-based behavior [16]. Also, VISSIM offers a multimodal traffic simulation environment with a user-friendly interface and high-quality 3D visualization [17]. Among the open-source options, MITSIMLab is a microscopic traffic simulation model that assesses the effects of different traffic management system designs on operational performance [18], and MATSim supports large-scale, agent-based simulations [19]. SUMO is an open-source microscopic traffic simulator under active development that is highly configurable and provides direct data access via the TraCI interface, making it particularly well-suited for integrating our UAV-based sensing plugin and achieving detailed, real-time microscopic traffic observations.

Through direct access to SUMO's internal states via TraCI, SUMO-UAV-Py aims to support real-time vehicle-level data collection, while allowing users to modify UAV trajectories dynamically. Moreover, its modular design facilitates coupling with external simulators or advanced UAV models, ensuring flexibility for researchers who require refined aerodynamics, path planning, or environmental conditions.

3. Plugin Architecture

SUMO-UAV-Py is a Python-based plugin designed to integrate SUMO's traffic simulation with UAV-based sensing. It accepts user-defined drone parameters, launches SUMO with the specified network configuration, and generates detailed aerial observations of vehicular traffic at each simulation step. In this section, we examine SUMO-UAV-Py's implementation, including its communication framework, the implemented UAV motion model, and our visualization technique.

3.1 Implementation Overview

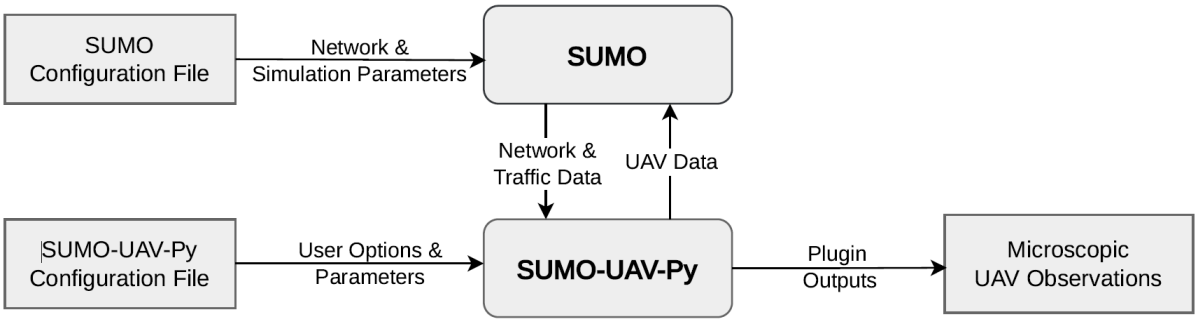

Figure 1. *High-level overview of SUMO-UAV-Py: initial parameters from configuration files are used to initialize SUMO and SUMO-UAV-Py, which communicates with SUMO-UAV-Py, to allow the communication between the two entities. The output of SUMO-UAV-Py is stored as a file containing Microscopic UAV Observations.*

Figure 1 presents a high-level diagram of SUMO-UAV-Py's architecture. Two configuration files serve as the initial inputs: the SUMO configuration file, which provides the network topology and simulation parameters, and the SUMO-UAV-Py configuration file, which specifies all UAV-related settings, including the total number of UAVs, their speed, rotation speed, and field-of-view (FoV) angles. SUMO supplies network and traffic data to SUMO-UAV-Py, while SUMO-UAV-Py sends updated UAV positions and orientations back to SUMO. The plugin then records all drone-based observations in an output file.

The SUMO-UAV-Py configuration file includes a list of 5D waypoints for each UAV, given by (t, x, y, z, ϕ). Here, t is the simulation time in which the UAV starts its movement to a position $(x,\ y\, z)$, with a yaw orientation ϕ.

The output file contains attributes that mimic real-world UAV sensor data for each simulation step, as detailed in Table 1.

Table 1. *Output File Attributes.*

Attribute Name	**Type**	**Description**
Step	Number	Current simulation step number.
Seconds	Number	Simulation time in seconds.
UAV_ID	String	Unique identifier of each UAV.
UAV_Pos	(x, y, z, ϕ)	4D coordinates of each UAV.
VehicleID	String	IDs of the detected vehicle.
Veh_Pos	(x, y)	2D coordinates of each detected vehicle.
Veh_Speed	Number	Speed of each detected vehicle.

3.2 Communication

The core of SUMO-UAV-Py's implementation relies on a TCP-based communication loop between SUMO, TraCI, and the SUMO-UAV-Py Python script, as illustrated in Figure 2. At each simulation step, SUMO-UAV-Py computes the UAV's new position, sends it to SUMO via TraCI, and retrieves the list of vehicles that fall within the UAV's calculated FoV. This connection enables continuous monitoring of both UAV motion and network-wide vehicle states. To optimize performance, SUMO-UAV-Py uses TraCI's object variable subscription to specify a set of variables (e.g., position, speed) that should be automatically returned in each step, avoiding the overhead of individually querying every vehicle.

3.3 Motion Model

To define each UAV's trajectory, SUMO-UAV-Py reads an initial 5D waypoint $(t_0, x_0, y_0, z_0, \phi_0)$ and the next 5D waypoint (t, x, y, z, ϕ). At simulation time t, the UAV rotates from ϕ_0 to align with the direction of (x, y, z), then moves from (x_0, y_0, z_0) toward (x, y, z). Upon arrival, the UAV performs a final rotation to match the yaw angle ϕ. This sequence of moves "rotate-move-rotate" provides a basic yet flexible approach for most UAV path scenarios.

In SUMO-UAV-Py the transitions between waypoints follow a lightweight linear approach model. Given two 3D points (x_1, y_1, z_1) and (x_2, y_2, z_2), the travel distance is computed as the Euclidean distance

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \quad (1)$$

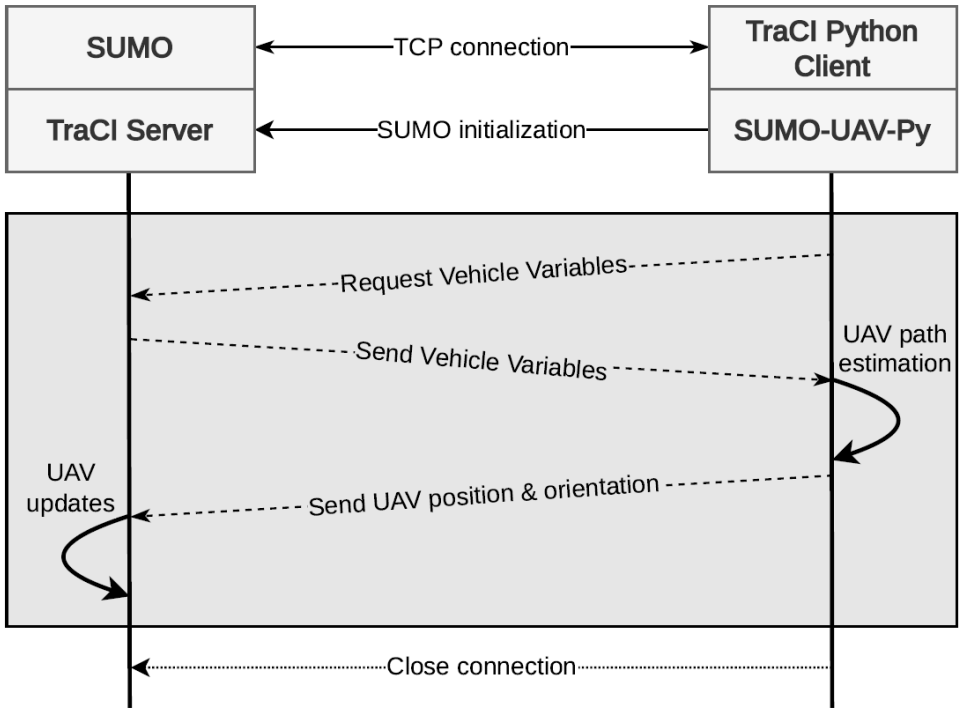

Figure 2. *Sequence diagram of TCP communication among SUMO and SUMO-UAV-Py, using TraCI.*

The number of steps required to reach the next waypoint, with a simulation step length Δt and horizontal speed v, is calculated using

$$N_{\text{move}} = \frac{d}{v\,\Delta t}. \tag{2}$$

Similarly, to calculate the number of rotation steps needed to cover a yaw-angle difference $\Delta\phi$ with yaw rate ω we use

$$N_{\text{rotate}} = \frac{\Delta\phi}{\omega\,\Delta t}. \tag{3}$$

For more complex flight paths the plugin is integrated with a discrete movement option that overrides the built-in dynamics, in which users manually supply 5D waypoints (t, x, y, z, ϕ), enabling seamless coupling with external trajectory planners.

Furthermore, SUMO-UAV-Py is able to determine the vehicles observable to each deployed UAV by calculating their camera FoV. The camera is fixed at a 90-degree angle relative to the ground, yielding a rectangular projection whose width and height scale linearly with altitude h. As illustrated in Fig. 3, for camera angles α (horizontal) and β (vertical), the half-view tangent geometry gives

$$FoV_x = 2\,h\,\tan\!\Big(\tfrac{\alpha}{2}\Big), \quad FoV_y = 2\,h\,\tan\!\Big(\tfrac{\beta}{2}\Big), \tag{4}$$

which are used to compute the FoV rectangle.

The computed dimensions FoV_x and FoV_y define a rectangle centered at the UAV's current position (x_u, y_u), oriented according to the UAV's yaw angle ϕ. To evaluate which vehicles are visible during each simulation step, SUMO-UAV-Py computes the global coordinates of the rectangle's corners after applying a planar rotation around the UAV center.

To compute the rotated position of a FoV corner, the point (x, y), defined in the local axis-aligned FoV frame, is first translated so that the UAV center (x_u, y_u) is moved

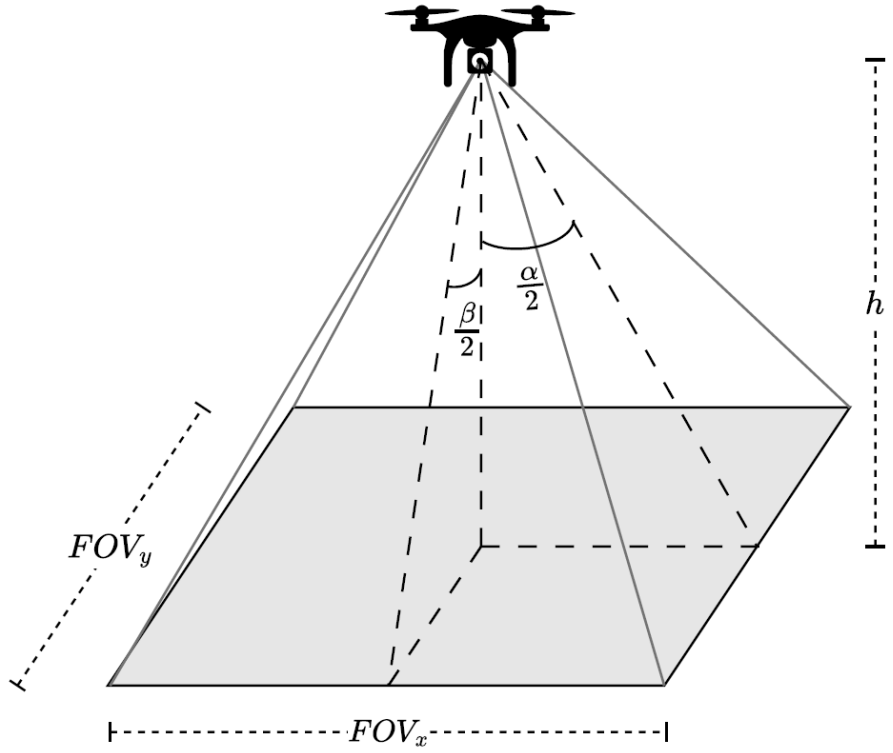

Figure 3. *Rectangular FoV representation for a $90°$ camera angle at height h.*

to the origin. A rotation by the yaw angle ϕ is then applied using a 2D rotation matrix. After the rotation, the point is translated back to its position relative to the UAV in the SUMO coordinate frame.

This results in the following transformation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = R(\phi) \left[\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_u \\ y_u \end{pmatrix} \right] + \begin{pmatrix} x_u \\ y_u \end{pmatrix}, \tag{5}$$

where $R(\phi)$ is the 2D rotation matrix defined as:

$$R(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}. \tag{6}$$

Here, (x, y) are the local coordinates of a FoV corner, and (x', y') are the corresponding coordinates in the global SUMO frame. This procedure is applied to all four corners of the FoV rectangle to compute its rotated projection on the plane. Vehicles whose positions fall within this transformed rectangle are marked as visible and recorded, based on data retrieved via TraCI subscriptions at each simulation step.

3.4 Visualization

When SUMO runs in graphical mode, SUMO-UAV-Py visually represents each UAV and its FoV by drawing polygons and points-of-interest (POIs). The rectangular FoV polygon moves and rotates with the UAV's position (x, y, z) and yaw angle ϕ. Also, it changes size relative to the UAV's altitude. The UAV position is marked with a POI icon. Figure 4 shows an example screenshot of these polygons and POIs in SUMO's GUI.

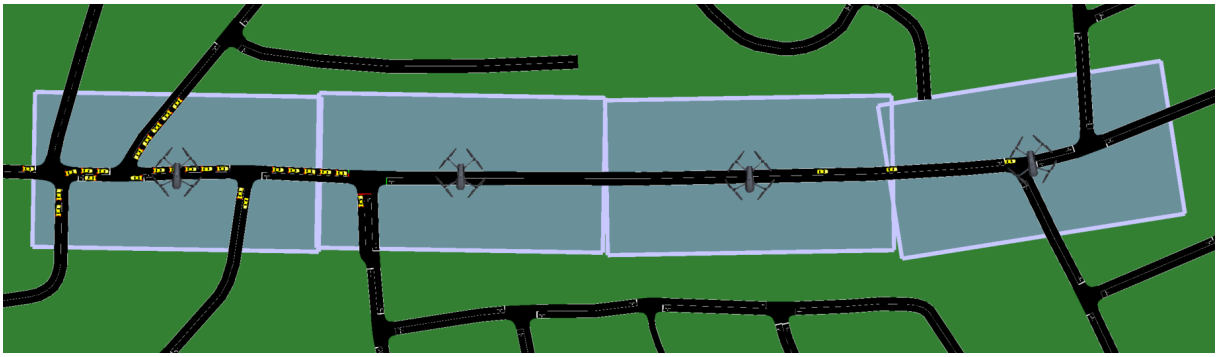

Figure 4. *Example screenshot of SUMO-UAV-Py's polygon- and POI-based visualization in SUMO's GUI.*

4. User Options and Features

The plugin offers a wide range of user-oriented features that maximize flexibility and ease of configuration. In addition to the ability to bypass UAV graphics for performance-oriented experiments, users can tailor multiple aspects of the simulation to fit specific research scenarios. These options can be selected via the configuration file.

4.1 UAV Model Parameters

The UAV dynamics generated by SUMO-UAV-Py are described by a lightweight linear motion model as stated in Section 3.3. However in respect to realistic modeling of speed, FoV characteristics and battery life approximations SUMO-UAV-Py comes with two preset options of well-known commercial UAV models: *Mavic 2e* [20] and *Mini 3 pro* [21]. For further customization, the plugin offers a *Manual* option, which allows users to specify custom values for maximum speed, yaw rate, and FoV dimensions.

The optional battery mode can be enabled, to assign a finite flight duration to each UAV. When battery levels drop to five minutes remaining, a warning is issued, and if depleted, the UAV ceases operation. This feature is essential for simulating realistic mission constraints.

4.2 UAV Observation Mode

SUMO-UAV-Py supports three observation modes:

- *Hovering*: The UAV rotates and moves toward the next waypoint, moving with a uniform speed. When it reaches the desired waypoint, it hovers until its time to move to the next waypoint. Hovering mode collects data continuously regardless of being stationary or in motion.
- *Sampling*: The UAV follows the same motion pattern as in Hovering mode, but records data only when stationary, thus capturing isolated sampling points.
- *Spinning*: Upon reaching a waypoint, the UAV rotates in place to maximize its FoV by covering a full or partial rotation. This mode is particularly useful when a wide-angle observation is desired as illustrated in figure 5.

4.3 Real-time interactions

In parallel with the TraCI loop, SUMO-UAV-Py implements a separate threaded mechanism to handle real-time interactions to modify the existing UAV flight paths or add new

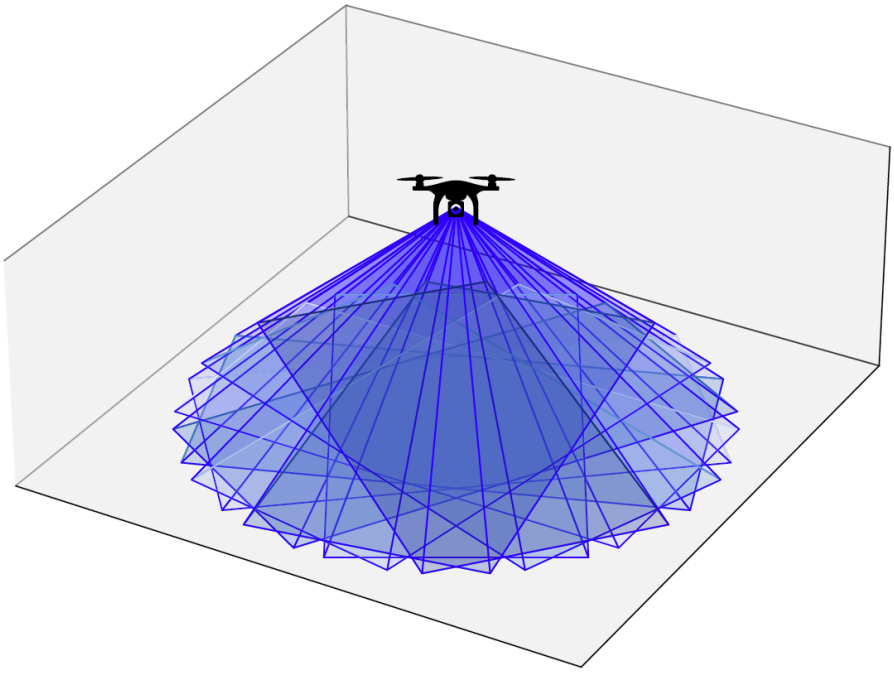

Figure 5. *FoV coverage visualization in the 2D plane in the Spinning UAV Observation Mode for different simulation steps.*

ones. This thread allows users to submit new waypoints in real-time (either through a local interface or an optional remote connection). Such interactions immediately update the drone's trajectory without pausing the simulation, allowing the users of SUMO-UAV-Py to implement their own open-loop or closed-loop logic, to dynamically adjust UAV positions as the simulation proceeds. The local interface for real-time modifications its shown in figure 6.

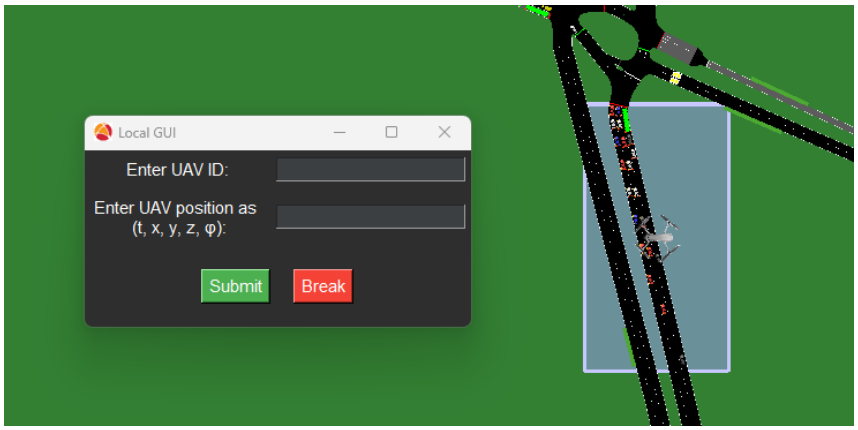

Figure 6. *SUMO-UAV-Py interface for real-time UAV updates.*

4.4 GUI Dialog

SUMO-UAV-Py provides a GUI dialog that includes all user-defined parameters in a single interface. This interface allows users to configure the total number of UAVs, select flight modes, set battery options, and specify the network and SUMO configuration file among others. Figure 7 shows the dialog window, illustrating how various simulation and UAV settings are easily accessible in one place.

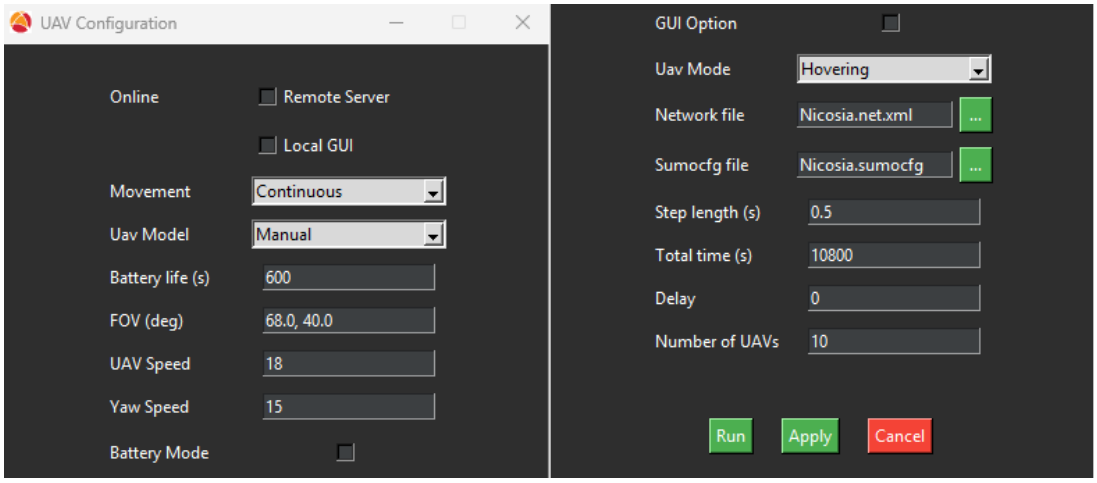

Figure 7. *GUI dialog for SUMO-UAV-Py configuration and user options selection.*

5. Performance Evaluation

To evaluate SUMO-UAV-Py's computational overhead, we conducted a benchmark using a real-world traffic scenario from the city of Bologna [22] (A.Costa scenario from iTetris). The scenario includes 8,600 vehicles, 179 edges, 112 nodes, 267 lanes and a total lane length of 33.52 km. All simulations covered three hours of virtual time and were executed on a machine with a 12th Gen Intel® Core™ i9-12900K processor and 32 GB of RAM.

We compared two approaches for capturing vehicle-level measurements of traffic:

- **SUMO-UAV-Py Logging (No GUI):** SUMO-UAV-Py records all vehicles within each UAV's FoV in real time. The number of UAVs was varied from 0 to 32 across different runs.
- **Post-Analysis Parsing:** SUMO was run in an identical setup, generating floating car data (`fcd-output`) file instead of the SUMO-UAV-Py implementation. A Python script using the library `xml.etree.ElementTree` then parsed the file to identify the same vehicles that would have been observed by the UAV(s).

Each scenario was executed five times for each UAV count (0 to 32) under identical conditions. We recorded the total simulation time, including post-processing in the second approach, and then computed average run times and standard errors. Figure 8 shows the results, where the "Plugin" line corresponds to aerial measurements logged by SUMO-UAV-Py and the "Post-Analysis" line represents offline parsing method.

Importantly, the UAV=0 case reflects the base simulation performance of SUMO without any UAV logging. In this case, SUMO-UAV-Py simply initializes but performs no additional computations. This allows us to measure the plugin's overhead relative to SUMO's base performance. In the figure, a horizontal line indicates this base SUMO timing as a reference.

For the case of zero UAVs, SUMO-UAV-Py runs faster than the post-analysis approach, because the latter requires additional processing to write and export the floating car data file. As the number of UAVs increases, the run times for both methods remain comparable, and there are instances in which SUMO-UAV-Py outperforms the post-analysis script. This result is partly due to SUMO-UAV-Py logging only those vehicles within each UAV's FoV, while the post-analysis method processes the entire network file. Overall, the extra computational cost of SUMO-UAV-Py's real-time data

extraction is minimal. These results confirm that SUMO-UAV-Py is well-suited for large-scale simulations where both detailed observations and computational efficiency are essential.

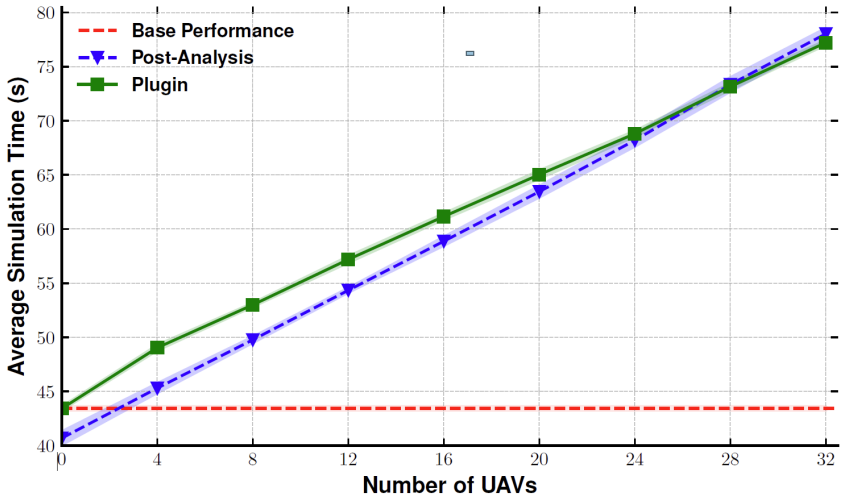

Figure 8. *Benchmarking of the SUMO-UAV-Py plugin versus a post-analysis script as described in Section 5. The results are the averaged timing over 5 executions for different number of UAVs. The shaded regions represent the standard errors. The horizontal line represents the SUMO base performance.*

6. Conclusion

This paper introduced SUMO-UAV-Py, an open-source plugin that integrates UAV-based sensing into SUMO's microscopic traffic simulations. By allowing researchers to specify UAV properties such as speed, FoV, and flight mode, SUMO-UAV-Py generates high-resolution aerial observations of traffic in a fully customizable manner. Its modular design enables users to incorporate custom external modules for UAV dynamics or path planning, offering a flexible framework for diverse research applications.

Performance evaluations showed that SUMO-UAV-Py's real-time data extraction imposes only a minimal additional computational cost compared with standard post-analysis approaches, and we demonstrate how the plugin scales up with the number of UAVs relative to SUMO's base performance. These results underscore the plugin's suitability for large-scale simulations where both detail and efficiency are critical. In future works, we plan to extend the basic motion model by introducing additional FoV constraints and variable camera angles, as well as to incorporate macroscopic UAV measurements such as density, flow, and average speed. We also welcome community feedback to drive further performance improvements and refinements. Overall, SUMO-UAV-Py provides a robust platform for advancing UAV-based traffic monitoring research.

Data availability statement

The code of SUMO-UAV-Py is open source and available on github: https://github.com/TsioutisCh/SUMO-UAV-Py. The referred network data used for the performance evaluation are also included. Video and an example output dataset of the plugin are included as well.

Author contributions

Charalambos Tsioutis: Conceptualization, Writing – original draft, Software, Methodology. Christos Makridis: Conceptualization, Writing – review & editing, Methodology. Stelios Timotheou: Conceptualization, Writing – review & editing, Supervision.

Competing interests

The authors declare that they have no competing interests.

Funding

This work is supported by the European Union (i. ERC, URANUS, No. 101088124 and ii. Horizon 2020 Teaming, KIOS CoE, No. 739551), and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation, and Digital Strategy. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "A survey of unmanned aerial vehicles (uavs) for traffic monitoring," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 221–234. DOI: 10.1109/ICUAS.2013.6564694.

[2] E. N. Barmpounakis, E. I. Vlahogianni, J. C. Golias, and A. Babinec, "How accurate are small drones for measuring microscopic traffic parameters?" *Transportation Letters*, vol. 11, no. 6, pp. 332–340, 2019, ISSN: 1942-7867. DOI: https://doi.org/10.1080/19427867.2017.1354433. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1942786722001369.

[3] I. Bisio, C. Garibotto, H. Haleem, F. Lavagetto, and A. Sciarrone, "A systematic review of drone based road traffic monitoring system," *IEEE Access*, vol. 10, pp. 101 537–101 555, 2022. DOI: 10.1109/ACCESS.2022.3207282.

[4] Y. Englezou, S. Timotheou, and C. Panayiotou, "Enhancing traffic state estimation using uav-based measurements," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 413–420. DOI: 10.1109/ICUAS60882.2024.10556942.

[5] K. Theocharides, C. Menelaou, Y. Englezou, and S. Timotheou, "Real-time unmanned aerial vehicle-based traffic state estimation for multi-regional traffic networks," *Transportation Research Record*, vol. 2678, no. 8, pp. 1–12, 2024. DOI: 10.1177/03611981231213079. eprint: https://doi.org/10.1177/03611981231213079. [Online]. Available: https://doi.org/10.1177/03611981231213079.

[6] P. A. Lopez et al., "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/.

[7] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds., Cham: Springer International Publishing, 2018, pp. 621–635, ISBN: 978-3-319-67361-5.

[8] Z. Wen, J. Zhao, Y. Xu, and A. Tsourdos, "A co-simulation digital twin with sumo and airsim for testing lane-based utm system concept," in *2024 IEEE Aerospace Conference*, 2024, pp. 1–11. DOI: 10.1109/AERO58975.2024.10521156.

[9] A. Al-Mousa, B. H. Sababha, N. Al-Madi, A. Barghouthi, and R. Younisse, "Utsim: A framework and simulator for uav air traffic integration, control, and communication," *International Journal of Advanced Robotic Systems*, vol. 16, no. 5, p. 1 729 881 419 870 937, 2019. DOI: 10.1177/1729881419870937. eprint: https://doi.org/10.1177/1729881419870937. [Online]. Available: https://doi.org/10.1177/1729881419870937.

[10] G. Dowd, O. Kavas-Torris, L. Guvenc, and B. Aksun-Guvenc, "Simulation environment for visualizing connected ground and air traffic," *Transportation Research Record*, vol. 2675, no. 6, pp. 15–22, 2021. DOI: 10.1177/0361198120962485. eprint: https://doi.org/10.1177/0361198120962485. [Online]. Available: https://doi.org/10.1177/0361198120962485.

[11] K. Thompson, F. J. Kurfess, D. Walter, R. Maksymiuk, R. Mevorach, and G. Joshi, "Uavsim: An open-source simulator for multiple uav path planning," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2022, pp. 229–236. DOI: 10.1109/DCOSS54816.2022.00047.

[12] T. Hardes, D. Logan, T. H. Pritom, and C. Sommer, "Towards an open source fully modular multi unmanned aerial vehicle simulation framework," in *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2022, pp. 284–289. DOI: 10.1109/ICDCSW56584.2022.00060.

[13] M. Uijt de Haag, S. Huschbeck, and C. Berth, "Modelling assured navigation of suas swarms in urban environments," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–8. DOI: 10.1109/DASC52595.2021.9594465.

[14] J. Wubben, C. T. Calafate, F. Granelli, J.-C. Cano, and P. Manzoni, "A real-time co-simulation framework for multi-uav environments offering detailed wireless channel models," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 2649–2654. DOI: 10.1109/ICC45041.2023.10278997.

[15] P. Garcia-Aunon, J. J. Roldán, and A. Barrientos, "Monitoring traffic in future cities with aerial swarms: Developing and optimizing a behavior-based surveillance algorithm," *Cognitive Systems Research*, vol. 54, pp. 273–286, 2019, ISSN: 1389-0417. DOI: https://doi.org/10.1016/j.cogsys.2018.10.031. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389041718303279.

[16] Aimsun, *Aimsun next 24 user's manual*, Aimsun Next 24.0.0, Barcelona, Spain, Accessed on: Month, Day, Year 2024. [Online]. [Online]. Available: https://docs.aimsun.com/next/24.0.0.

[17] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator vissim," in Jun. 2011, pp. 63–93, ISBN: 978-1-4419-6141-9. DOI: 10.1007/978-1-4419-6142-6_2.

[18] M. Ben-Akiva, H. Koutsopoulos, T. Toledo, Q. Yang, C. Choudhury, and R. Balakrishna, "Traffic simulation with mitsimlab," *Fundamentals of Traffic Simulation*, vol. 145, p. 233, Jun. 2010. DOI: 10.1007/978-1-4419-6142-6_6.

[19] A. Horni, K. Nagel, and K. Axhausen, Eds., *Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press, Aug. 2016, p. 618, ISBN: 978-1-909188-75-4, 978-1-909188-76-1, 978-1-909188-77-8, 978-1-909188-78-5. DOI: 10.5334/baw.

[20] M. 2e, *Dji mavic 2e specs*, https://www.dji.com/global/support/product/mavic-2-enterprise, Accessed: April 03, 2025.

[21] M. 3. Pro, *Dji mini 3 pro specs*, https://www.dji.com/support/product/mini-3-pro, Accessed: April 03, 2025.

[22] L. Bieker, D. Krajzewicz, A. Morra, C.Michelacci, and F.Cartolano, "Traffic simulation for all: A Real World traffic scenario from the city of bologna," in *Proceedings of the SUMO Conference 2014*, 2014, pp. 19–26.