Towards an Open Source Fully Modular Multi Unmanned Aerial Vehicle Simulation Framework

Tobias Hardes*†‡, Dalisha Logan†‡, Touhid Hossain Pritom†‡, and Christoph Sommer*
*TU Dresden, Faculty of Computer Science, Germany
†Paderborn University, Dept. of Computer Science, Germany
‡Software Innovation Campus Paderborn, Germany
https://www.cms-labs.org/people/ { hardes , dalisha.logan , touhid.pritom , sommer }

***Abstract*—With the trend towards smart cities, the use of Unmanned Aerial Vehicles (UAVs) is becoming more and more diverse. An economical and insightful way to study the use of UAVs in cities is through the utilization of computer simulations. However, currently available simulators are only of limited use to investigate such scenarios: often they have been developed for a dedicated use case, can only handle single UAVs, or the software was not made available with an open license. In this paper, we present AirMobiSim, a modular and microscopic UAV simulation framework, which is available under a GPLv2 license. AirMobiSim provides the basis for the creation of kinematic and energy models for different UAV types. It can support an arbitrary number of UAVs and can be coupled with other simulators via open interfaces, for example to examine systems using wireless communication between UAVs. We furthermore present our approach for model-building and show that AirMobiSim can already accurately reproduce existing work. We also highlight current limitations and show prospects for future work.**

I. Introduction

There are already several different possible applications today. These include Unmanned Aerial Vehicles (UAVs) as mobile data carriers [1] or relays [2], parcel delivery services [3], medical supply [4], monitoring tasks [5], or precision agriculture [6]. The increasing number of civil applications for UAVs is leading to more attention being paid to this area in research.

This trend is likely to be amplified by the smart city and Internet of Things (IoT) trends. However, a real-world UAV deployment must be planned or evaluated in advance appropriately.

This is often done as part of field tests, such as the one DHL conducted for its own parcel delivery UAV. However, such field tests are usually very complex, expensive, limited to a certain size, and do usually only cover a very specific use case in a specific scenario. If one of its conditions is changed, a field test must be repeated which often causes the costs to rise sharply. In combination with possible legal barriers, field tests are not feasible in a lot of cases.

Computer simulations offer an alternative to field testing by analyzing the real system through software models. In doing so, input parameters (e.g., number of UAVs, physical properties, etc.) can be quickly changed and simulations can then be repeated at low cost. Depending on the abstraction of the software model, conclusions from simulations can be drawn about the effects in the real world.

The topics of possible UAV simulation studies are manifold. For example, different positions for UAV takeoff and landing locations could be evaluated through simulations. For example, these locations can be evaluated in terms of various metrics such as cost, flight time, etc. Furthermore, a pure evaluation of algorithms is possible, for example for collision avoidance using different flight maneuvers or parameters.

However, to make all this possible, software models are required to approximate reality sufficiently well. This includes sufficiently realistic modeling of flight behavior of various flight maneuvers, but also different types of UAVs.

For many use cases, pure mobility studies are often already sufficient. Especially in the environment of IoT and smart cities, however, UAVs often coexist with other nodes such as road vehicles. Thus, a pure consideration of air traffic may not be sufficient here. Previous work [2], [7] has considered the coexistence of UAVs in the air and vehicles on the ground. In addition to UAVs and road vehicles, these studies also consider aspects regarding wireless communication. A wide range of applications therefore need simulations of road traffic, air traffic, and communication.

In this paper, we introduce concepts and considerations for such a simulator and present *AirMobiSim*, our Open Source unmanned aerial vehicle simulation framework. The simulator is available under the terms of a GPL license[1].

AirMobiSim is a microscopic simulation framework for UAVs that can be used stand-alone and it provides a basis for the creation of kinematic and energy models for different UAVs. In the current version, we provide an implementation for the Crazyflie 2.1 platform. We furthermore enable coupling AirMobiSim with any other simulators via a generic gRPC interface. The current version of AirMobiSim provides interfaces to OMNeT++ frameworks like Veins [8] to enable wireless communication between UAVs and to couple AirMobiSim with SUMO for hybrid simulations of air and road traffic mobility. In doing so, we also implemented the coupling with the popular INET Framework for further usage of existing models such as the ones provided by simuLTE [9] or simu5G [10]. However, the coupling capabilities of AirMobiSim are not limited to OMNeT++; e.g., simulators like NS-3 can be integrated using the same interface and methodology.

[1] http://airmobisim.org

15th International Workshop on Wireless Sensor, Robot and UAV Networks (IEEE WiSARN 2022)

In brief, the key contributions of this paper are:

- We present AirMobiSim, an Open Source simulation framework for UAV kinematics and energy aspects[1].
- We describe the current structure of AirMobiSim and explain how it can be extended through generic interfaces.
- We demonstrate the flexibility and applicability for simulation studies with multiple UAVs communicating with road vehicles via IEEE 802.11p.

II. Use Cases

In recent years new use cases for single and multi-UAV systems appeared.

One research area is collision avoidance studies [11]. With AirMobiSim, algorithms can be evaluated in different scenarios without additional frameworks before being deployed in the real world. The available gRPC interfaces of AirMobiSim also allow the evaluation of cooperative approaches that use wireless communication for cooperative collision avoidance [12].

Another use case can be the evaluation of UAVs as data carriers in disaster scenarios [1]. UAVs acting as flying cell stations could thus provide connectivity in such situations. AirMobiSim can help here to evaluate an optimal deployment of such flying cells within a geographic area. This requires a coupling of AirMobiSim with a mobile broadband simulator like simu5G [10] or any other framework to simulate communication aspects.

A third use case can may be the coexistence of road vehicles and UAVs. Since road vehicles and UAVs are very likely to exist together in smart cities, their interaction (for example, in terms of communication) could be investigated here. Delivery UAVs could be launched directly from a delivery truck. While a UAV delivers a package, the vehicle continues the delivery route. After delivery, the UAV flies back to the truck, which requires communication, e.g., to exchange data regarding the vehicle's position. This combination of road traffic, air traffic, and communication can be realized with the help of AirMobiSim.

For all the use cases presented, precise planning and evaluation of the algorithms is necessary. Simulations offer a good middle ground here, which can greatly reduce the required effort and financial investment. There are already a large number of simulations available today, but almost all of them are not suitable for the investigation of such questions.

III. Related Work

The literature provides a variety of different simulator implementations for UAVs. Some of the simulators are community driven and developed, others are purposed built for scientific publications [13].

The aim of Microsoft AirSim [14] is to support the development and evaluation of autonomous vehicle applications. The focus is not only on UAVs but also on different road vehicles. A special focus is on the development of reinforcement learning algorithms. The kinematic properties of UAVs are derived from the Unreal Engine and it is therefore not clear how far these models are directly transferable to the real world. Furthermore, AirSim is not modularly extensible and is primarily designed to simulate a single UAV.

Gazebo [15] was developed back in 2002. It is one of the most popular multi-robot simulators, it includes several physics engines, and allows the evaluation of algorithms, but also of robot/UAV designs. However, its kinematic model is not purpose built for UAVs [16]. It also operates on a low level of abstraction, potentially leading to long simulation runtimes [17].

Morse [18] is a simulator for the evaluation of robotics and UAVs. It is based on Blender for physics simulation and can simulate multiple UAVs at the same time. Like for Microsoft AirSim, the kinematic properties are derived from the physics engine and applicability to the real world is not ensured. Morse is furthermore not providing any opportunities for coupling or to use wireless communication.

There are also simulation frameworks available that are integrated into the OMNeT++ ecosystem.

The INET Framework is an Open Source OMNeT++ model suite for wired, wireless, and mobile networks. The framework has various mobility models. This allows various movements to be mapped, but it is not guaranteed that these also correspond to the real flight behavior of a UAV. However, the large number of available models for wireless communication are certainly beneficial for simulation studies regarding UAVs.

LIMoSim [19] is an open simulation framework that can be used with OMNeT++ and NS-3. It provides mobility and energy models. Although the simulator is versatile, core functionality is provided in NS-3, meaning NS-3 is required for any kind of simulation.

FlyNetSim [20] is another example, which however forces hardware-in-the-loop. Thus it works on a low level of abstraction, potentially resulting in long simulation runtimes.

UAV Sim [21] is an OMNeT++ based simulator with the purpose of cyber security analysis. It is based on the INET Framework to enable communication between UAVs. There are also mobility models, but they are not described in more detail. Finally, this simulator is available as an Open Source project, but it is not maintained anymore.

Dietrich et al. [22] present a time discrete simulator with a focus on energy maintenance and node replacement strategies. The aim of the simulators is to analyze deployment planning strategies and management processes for multiple UAVs which can be optimized with the help of the simulator. The development is based on OMNeT++. However, the simulator focuses strongly on mobility aspects while communication is only considered in an abstract way. Additionally, the simulator itself is not made available under an open license.

Lieser et al. [23] propose a simulation platform that combines UAV mobility, ad hoc communication on the ground, and the corresponding human mobility models. The authors include communication between UAVs and support strategies for post-disaster scenarios. The simulator uses external libraries to enable communication within different nodes in the simulation. Thus, the authors are able to study support strategies for post-disaster communication networks using their simulator. However, the implementation is not Open Source and the libraries used for communication continue to have not been further developed for several years.

The simulators and frameworks shown here have all been able to handle their individual use case. However, many approaches do not allow further use since they are not available as an Open Source solution with an open license. Others do not provide dedicated models for UAVs that are appropriately validated by real systems. Still other simulators model custom aspects of wireless communication in addition to mobility without relying on established models here.

In this paper, we close this gap with AirMobiSim, an Open Source unmanned aerial vehicle simulation framework that is available under the terms of a GPL license[1]. AirMobiSim provides a basis for the creation of kinematic and energy models for different UAVs and enables coupling with any other simulator via generic gRPC interfaces. In the current version, simultaneous simulations of UAVs, road traffic, and wireless communication are possible.

IV. AirMobiSim Concept

AirMobiSim is a microscopic simulator for UAV kinematics and energy aspects. It is a stand-alone command-line-based simulator that is developed in Python 3. We implement AirMobiSim as a time-discrete simulation, meaning we model the behavior of UAVs as a sequence of discrete states in time.

AirMobiSim is licensed with GPLv2 or any later version and an Open Source project[1].

The idea of AirMobiSim is to provide a framework and an infrastructure with which other researchers and end-users can develop their own models for individual UAVs that are available as hardware. Models in AirMobiSim primarily represent the kinematic behavior of a UAV, i.e. the motion characteristics.

Currently, AirMobiSim includes first models for the Bitcraze Crazyflie 2.1 platform, which we use for several reasons.

First, Crazyflie is an Open Source flying development platform with open APIs and a modular setup. The hardware can be programmed with Python and provides libraries for automated control of UAVs.

Second, the Crazyflie platform is expandable by using different expansion decks. These decks include capabilities for storing large amounts of data directly on the UAV, but also decks for localizing of UAVs in 3D space.

Third, the Crazyflie platform is often used in academic settings [24], [25]. Several publications use the Crazyflie platform for a wide variety of use cases. These include cooperative strategies for different use cases [26], vehicle tracking [27], or swarm formation [28], [29].

To allow a straightforward evaluation of simulation results, AirMobiSim records statistics as a CSV file. These results are recorded for each UAV in the simulation. They include time-dependent information regarding the position, velocity, etc., but also scalar data.

AirMobiSim provides generic gRPC interfaces that can be used to couple AirMobiSim with other simulation frameworks. For example, coupling with simulators that provide simulation models for wireless communication is possible.

Figure 1. Measurement setup using HTC Vive base stations. The base stations are mounted on tripods and at a height of 1.6 m. The total flight space has a size of 3 m x 3 m.

Currently, AirMobiSim possesses two mobility models, which we call *Linear Mobility* and *Spline Mobility*. Regardless, however, the models are arbitrarily extensible and can be developed for other UAV platforms or with a different level of abstraction.

A. Kinematic Model

Our model development relies heavily on real-world measurements, for example on the Crazyflie 2.1 platform. The idea of the development is to let a UAV fly different maneuvers first. During the flight, the position of the UAV is recorded with very small time intervals of 0.01 s to be able to record the trajectory. Accordingly, precise localization and a stable flight behavior of the UAV in 3D space is necessary. There are several approaches to this for the Crazyflie 2.1 platform.

Our first experiments built on the Bitcraze *Loco Positioning System*. The system is based on ultra wideband radio to measure the position of an object in 3D space. Following the instructions of the available tutorial, we positioned a total of 8 *Anchor Nodes* in our lab and equipped the UAV with a Crazyflie 2.1 *Loco Tag*. Using it, the UAV can locate its position based on the measured distance to the anchor nodes. Our experiments showed that the flight characteristics of the UAV are more stable than with relative positioning. However, the UAV is not stable enough to derive model properties and the measurement series became inaccurate. A derivation of the flight characteristics was therefore not possible using the Loco Positioning System.

In a second experiment, we used *HTC Vive base stations* (Lighthouse) together with the Crazyflie 2.1 *Lighthouse deck*. The Lighthouse system generates a measurement error, which is, however, negligible [30]. In fact, this measurement setup allowed a very stable flight behavior, which was also confirmed with different maneuvers. Figure 1 shows the setup of our hardware experiments.

We use the Motion Commander from Crazyflie for the development of autonomous flights that are used for our measurement series. This has the advantage that influences by a human pilot are excluded.

We started with very simple maneuvers, such as flying a straight line. This kind of maneuver allows a first implementation of a model with low complexity. Furthermore, we could evaluate the integration of the models into the AirMobiSim ecosystem. We call this first model the *Linear Mobility* model. Although the model itself is not complex, it approximates the flight behavior of point-to-point flying UAVs often used in related work [2], [31]. In this mobility model, a UAV starts at a position and moves toward the target position. The flight angle and speed are constant. The acceleration is set to zero. Accordingly, during the hardware measurements, we only recorded data when the desired speed was reached. This reduces side effects due to acceleration or deceleration.

For a second model, we consider more complex movements, such as curvy trajectories. This model considers a list of consecutive waypoints, e.g., a set of positions (x, y, z) a UAV has to reach during simulation and the hardware experiment.

To minimize the complexity of this mobility pattern in simulation, we use a method to interpolate the trajectory between two defined waypoints. This approach is based on cubic spline interpolation. This means that the connection between two waypoints is defined by a cubic parabola. We call this model *Spline Mobility*.

We repeat all experiments 10 times for statistical confidence and cut take-off and landing phases. Our measurements show that the differences between different measurement series are negligible.

We analyze both mobility models by comparing the results of the simulations with those obtained by the hardware measurements. We do this through graphical validation [32].

We record the position of the UAV every 0.01 s in the hardware measurement using the Lighthouse setup. Since the kinematic model is based on waypoints, we transfer waypoints from the hardware measurement to the simulation. The path shown here consists of 10 waypoints, each acquired with a time distance of 1 s at a constant velocity. The path between the waypoints is calculated by cubic spline interpolation. Our simulation also uses a step length of 0.01 s.

Figure 2 shows the data for the Spline Mobility model (data for Linear Mobility not shown).

The graphical validation shows that the model approximates the hardware experiment very well. Visible differences between the two experiments result from the interpolation between the waypoints, but also from errors with the positioning and thus in an inaccurate flight trajectory of the hardware. Regarding the scale of the axes of the plot, however, the differences are very small, so the interpolation proves to be very useful and sufficiently accurate here.

B. Interfaces

AirMobiSim can be used standalone for pure mobility studies. However, as smart city initiatives in general and the deployment of 5G in particular continue to be driven forward, pure mobility studies cover only a small part of the possible application spectrum of UAVs. Consideration of wireless communication between UAVs, but also between UAVs and other nodes in the network is therefore often necessary to investigate real-world problems and situations.

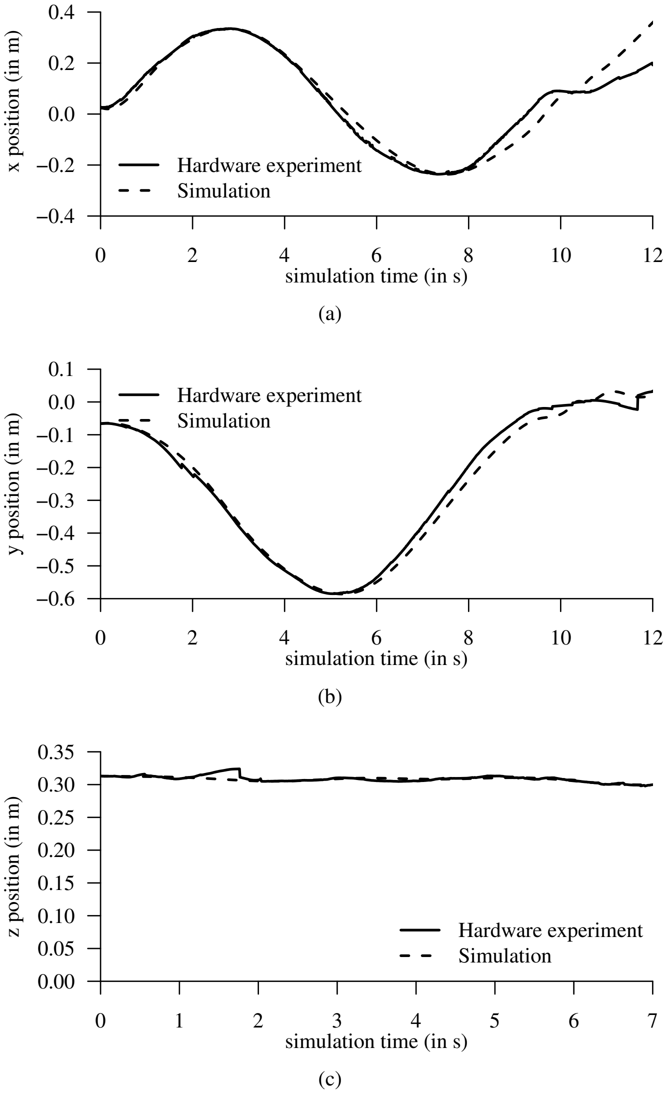

Figure 2. Position of UAV in simulation and in hardware measurement as a function of time.

Since there are specialized and established simulators for road traffic or wireless communication, we follow a coupling approach using Google Protobuf and gRPC interfaces to connect any external simulator. We use gRPC as a Remote Procedure Call (RPC) framework to connect different services or applications. It is offered in various programming languages like C++, Python, Java, or GO. Through gRPC we enable coupling of AirMobiSim with any external software. This means that other models, for example for wireless communication, do not have to be developed from scratch, but can simply be integrated via this interface.

To use the gRPC interfaces, AirMobiSim provides the *Drone Command Interface Bridge (DroCI-Bridge)* module. This interface can be used by any simulator or framework. For this, the API definitions must be implemented only on the

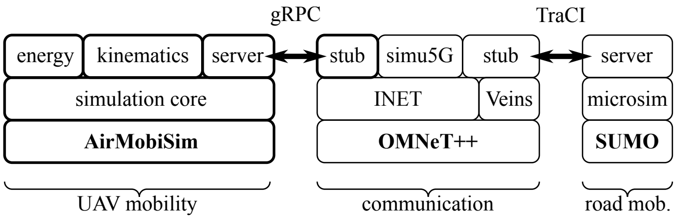

Figure 3. Example of a coupling of AirMobiSim with OMNeT++ and SUMO.

opposite side likewise. Due to the platform independence and a large number of supported programming languages, wide use is possible.

V. Proof of Concept

We show an example use case of AirMobiSim by re-simulating a previous work [33] which used the *Linear Mobility* model from the INET Framework, now using a kinematic model from AirMobiSim. The study includes UAV movements, simulations of road traffic, and communication using IEEE 802.11p.

To enable wireless communication within AirMobiSim, our current version couples AirMobiSim with Veins [8]. Veins is an extension to the OMNeT++ network simulator and provides fully-detailed models of the IEEE 802.11p network layers. It additionally offers coupling with the simulator SUMO for road traffic mobility. This allows, first, the coexistence of road vehicles on the ground and UAVs in the air and, second, communication between UAVs and between UAVs and road vehicles. All that is needed is a stub in Veins for the gRPC connection to the server module in AirMobiSim. Like AirMobiSim we make these OMNeT++ modules available as Open Source.

Coupling with Veins would also allow coupling with all extensions of Veins. This includes for example the platooning extension Plexe [34]. A coupling with the INET Framework and all extensions of INET is also possible. INET itself is the standard communication protocol library for OMNeT++ and provides different models, e.g., for IEEE 802.11, MAC protocols, or QoS mechanisms. Known extensions of INET include, e.g., simuLTE [9] or simu5G [10].

Figure 3 is showing the coupling of AirMobiSim and the OMNeT++ ecosystem using Veins. In the same way a coupling with any other simulations like NS-3 is possible.

We investigate the effects of using randomly passing UAVs at an urban intersection to improve the perception of vehicles on the ground. We use AirMobiSim to simulate point-to-point flying UAVs using the Linear Mobility model. We couple Veins to enable the use of the OMNeT++ INET framework for wireless network modeling and SUMO for road mobility. We use a symmetric intersection with four legs of 500 m each from our previous study. The intersection is surrounded by buildings that are fully opaque to radio transmissions.

UAVs are flying at a constant speed (20 m/s) and height (70 m). Furthermore, UAVs spawn at the outer edge of the scenario with a mean inter-arrival time of 30 s.

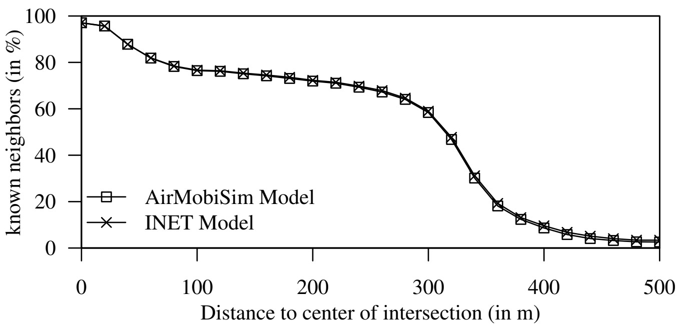

Figure 4. Data for the experiment using AirMobiSim and the models from the INET Framework. Both results are almost identical.

Figure 4 shows the results of this work. The data shows the relative share of detected vehicles within an area of 100 m (euclidean distance) around the center of the intersection. The data proves that the values from the previous work can be reproduced almost identical. As shown in the data, however, there is a deviation that can be traced back to localization errors. Based on this, more extensive parameter studies in AirMobiSim could cover a complete evaluation of the value space of different UAV parameters in the next step.

Our simulation results show that AirMobiSim provides a suitable environment for the presented simulation study. This opens the door to investigations with more extensive kinematic models with more information value.

VI. Conclusion and Future Work

We presented AirMobiSim, a modular and microscopic Unmanned Aerial Vehicle (UAV) simulation framework for multiple UAVs, which is available under a GPLv2 license. AirMobiSim provides a basis for the creation of kinematic and energy models for different UAV types. Unlike related work it is Open Source, it can support an arbitrary number of UAVs, and can be coupled with other simulators like OMNeT++ or NS-3 via open interfaces, for example to enable wireless communication between UAVs. We furthermore presented our approach for model-building and showed that AirMobiSim can accurately reproduce existing work.

Building on the current state of AirMobiSim, there are two straightforward directions for future work.

First, the current kinematic models have been developed specifically for the Crazyflie 2.1 platform. Also, these models are not yet fully comprehensive. For example, the acceleration behavior of UAVs is currently not yet fully considered. For more complex studies regarding collision avoidance, however, this consideration is certainly advantageous. Yet, AirMobiSim itself is built in a way such that these models can be easily adapted or extended, but also completely replaced. For future work we will expand and further refine the models for the Crazyflie 2.1. However, it is also possible to replace the models of the Crazyflie 2.1 with those of a real parcel UAV based on experiments with the hardware.

Second, the existing energy model is very abstract so far and not suitable for simulation studies. This is mainly because

Crazyflie 2.1 cannot provide the required data for a realistic energy model. We therefore plan to use a second hardware platform for future work. This will probably be based on the established Pixhawk series. The Pixhawk series is an independent Open Source hardware platform project [13]. Pixhawk collaborates with several partners, e.g., the DroneCode project, and it is suitable for different kinds of UAVs or frames like multirotor, fixed-wing, gliders, and others.

ACKNOWLEDGMENTS

This work was supported by a grant from Software Campus through the German Federal Ministry of Education and Research (Project Number: 01IS17046). The authors gratefully acknowledge the funding of this project by computing time provided by the Paderborn Center for Parallel Computing (PC^2).

REFERENCES

[1] J. Zobel, P. Lieser, B. Drescher, B. Freisleben, and R. Steinmetz, "Optimizing Inter-Cluster Flights of Post-Disaster Communication Support UAVs," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, Osnabrück, Germany: IEEE, Oct. 2019, pp. 364–371.

[2] T. Hardes and C. Sommer, "Opportunistic UAV Relaying for Urban Vehicular Networks," in *17th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2022)*, Virtual Conference: IEEE, Mar. 2022.

[3] G. Cavone, N. Epicoco, R. Carli, A. Del Zotti, J. Paulo Ribeiro Pereira, and M. Dotoli, "Parcel Delivery with Drones: Multi-criteria Analysis of Trendy System Architectures," in *29th Mediterranean Conference on Control and Automation (MED)*, Puglia, Italy: IEEE, Jun. 2021.

[4] Y.-H. Ma, T.-H. Shieh, L.-C. Hung, and L.-C. Tsai, "Research on The New Drones Design to Improve Transportation of Supplies and Medical Care for Remote Mountain Area in Taiwan," in *9th International Conference on Orange Technology (ICOT)*, Tainan, Taiwan: IEEE, Dec. 2021.

[5] Q. Zhou and J. Yang, "Research on Reliability Modeling of Image Transmission Task Based on UAV Avionics System," in *15th International Wireless Communications Mobile Computing Conference (IWCMC)*, Tangier, Morocco: IEEE, Jun. 2019, pp. 1504–1507.

[6] V. P. Subba Rao and G. S. Rao, "Design and Modelling of an Affordable UAV Based Pesticide Sprayer in Agriculture Applications," in *5th International Conference on Electrical Energy Systems (ICEES)*, Chennai, India: IEEE, Feb. 2019.

[7] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone Assisted Vehicular Networks: Architecture, Challenges and Opportunities," *IEEE Network*, vol. 32, no. 3, pp. 130–137, May 2018.

[8] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, Jan. 2011.

[9] A. Virdis, G. Stea, and G. Nardini, "Simulating LTE/LTE-Advanced Networks with SimuLTE," in *Simulation and Modeling Methodologies, Technologies and Applications*, Springer, 2016, pp. 83–105.

[10] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G – An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks," *IEEE Access*, vol. 8, pp. 181 176–181 191, Jan. 2020.

[11] V. Casas and A. Mitschele-Thiel, "From simulation to reality: a implementable self-organized collision avoidance algorithm for autonomous UAVs," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece: IEEE, Sep. 2020, pp. 822–831.

[12] J. Sun, J. Tang, and S. Lao, "Collision Avoidance for Cooperative UAVs With Optimized Artificial Potential Field Algorithm," *IEEE Access*, vol. 5, pp. 18 382–18 390, 2017.

[13] E. Ebeid, M. Skriver, K. H. Terkildsen, K. Jensen, and U. P. Schultz, "A survey of Open-Source UAV flight controllers and flight simulators," *Microprocessors and Microsystems*, vol. 61, pp. 11–20, Sep. 2018.

[14] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *11th Conference on Field and Service Robotics (FSR 2017)*, Springer, 2017, pp. 621–635.

[15] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Sendai, Japan: IEEE, Sep. 2004, 2149–2154 vol.3.

[16] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, "Comprehensive simulation of quadrotor uavs using ros and gazebo," in *International conference on simulation, modeling, and programming for autonomous robots*, Springer, 2012, pp. 400–411.

[17] L. Pitonakova, M. Giuliani, A. Pipe, and A. Winfield, "Feature and performance comparison of the V-REP, Gazebo and ARGoS robot simulators," in *Annual Conference Towards Autonomous Robotic Systems*, Springer, 2018, pp. 357–368.

[18] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, "Modular open robots simulation engine: MORSE," in *IEEE International Conference on Robotics and Automation*, Shanghai, China: IEEE, May 2011.

[19] B. Sliwa, M. Patchou, K. Heimann, and C. Wietfeld, "Simulating Hybrid Aerial- and Ground-Based Vehicular Networks with Ns-3 and LIMoSim," in *Proceedings of the 2020 Workshop on Ns-3*, ser. WNS3 2020, Gaithersburg, MD, USA: ACM, 2020.

[20] S. Baidya, Z. Shaikh, and M. Levorato, "FlyNetSim: An Open Source Synchronized UAV Network Simulator based on ns-3 and Ardupilot," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ACM, 2018.

[21] A. Y. Javaid, W. Sun, and M. Alam, "UAVSim: A simulation testbed for unmanned aerial vehicle network cyber security analysis," in *IEEE Globecom Workshops (GC Wkshps)*, Atlanta, GA, USA: IEEE, Dec. 2013, pp. 1432–1436.

[22] T. Dietrich, S. Krug, and A. Zimmermann, "A discrete event simulation and evaluation framework for multi UAV system maintenance processes," in *2017 IEEE International Systems Engineering Symposium (ISSE)*, IEEE, Oct. 2017.

[23] P. Lieser, J. Zobel, B. Richerzhagen, and R. Steinmetz, "Simulation Platform for Unmanned Aerial Systems in Emergency Ad Hoc Networks," in *ISCRAM*, 2019.

[24] C. Budaciu, N. Botezatu, M. Kloetzer, and A. Burlacu, "On the Evaluation of the Crazyflie Modular Quadcopter System," in *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain: IEEE, Sep. 2019, pp. 1189–1195.

[25] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland: IEEE, Aug. 2017, pp. 37–42.

[26] J. Kim, C. D. Eggleton, S. A. Wilkerson, and S. A. Gadsden, "Cooperative Sensor-based Selective Graph Exploration Strategy for a Team of Quadrotors," *Journal of Intelligent & Robotic Systems, Springer*, vol. 103, no. 2, p. 24, Sep. 2021.

[27] J. Lee, J. Moon, and S. Kim, "UWB-based Multiple UAV Control System for Indoor Ground Vehicle Tracking," in *IEEE VTS 17th Asia Pacific Wireless Communications Symposium (APWCS)*, Osaka, Japan: IEEE, Aug. 2021.

[28] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore: IEEE, Jun. 2017.

[29] B. Rinner, C. Bettstetter, H. Hellwagner, and S. Weiss, "Multidrone Systems: More Than the Sum of the Parts," *IEEE Computer*, vol. 54, no. 5, pp. 34–43, 2021.

[30] P. Bauer, W. Lienhart, and S. Jost, "Accuracy Investigation of the Pose Determination of a VR System," *Sensors*, vol. 21, no. 5, Feb. 2021.

[31] D. Liu et al., "Opportunistic UAV Utilization in Wireless Networks: Motivations, Applications, and Challenges," *IEEE Communications Magazine*, vol. 58, no. 5, pp. 62–68, 2020.

[32] R. G. Sargent, "Verification and validation of simulation models," in *39th Winter Simulation Conference (WSC 2007)*, Piscataway, NJ: IEEE, Dec. 2007, pp. 124–137.

[33] T. Hardes, C. Boos, and C. Sommer, "Towards opportunistic UAV relaying for smart cities," in *International Conference on Networked Systems (NetSys 2021)*, Virtual Conference, Sep. 2021.

[34] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *6th IEEE Vehicular Networking Conference (VNC 2014)*, Paderborn, Germany: IEEE, Dec. 2014, pp. 53–60.