Simulation Tools, Environments and Frameworks for UAV Systems Performance Analysis

Aicha Idriss Hentati, Lobna Krichen, Mohamed Fourati and Lamia Chaari Fourati
Laboratory of Technology and Smart Systems (LT2S),
Digital Research Center of SFAX (CRNS),
University of Sfax, Tunisia
aicha.idriss@gmail.com, lobnaakrichen@gmail.com, fourati@tunet.tn, lamia.chaari@enis.rnu.tn

***Abstract*—Nowadays, Unmanned Aerial Vehicles (UAVs) are widely deployed in many contexts (civilian and military). Real experiments using UAVs are costly; therefore, the performances of UAVs systems should be analyzed before their deployments. Accordingly, researchers and software engineers developed several simulations tools, environments and frameworks for UAV systems evaluation. In this context, this study highlights and identifies the most suitable simulators for UAVs flights and UAVs managements. This paper details the requirements, the goals, the strengths and the weakness of each studied tools. This investigation helps researchers to identify and to select the adequate UAVs performances analysis tools that satisfy their needs.**

***Index Terms*—Unmanned Aerial Vehicle; UAV system; Flight Simulators; Ground Control Station.**

I. Introduction

Recently, Unmanned Aerial Vehicles (UAVs) [1] are widely serving because their capability of performing complex tasks such as surveillance [2], disaster monitoring [3] [4], managing wildfire [5], tracking [6] [7], etc. UAV systems consist of a number of sub-systems which comprehend the UAV, the ground control station (GCS) and the communication links as shown in Fig.1.

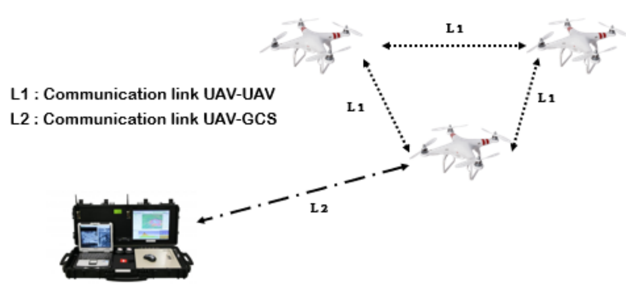

Fig. 1. UAVs system architecture

UAV is a flight vehicle which operates without any human intervention. Table I. presents UAVs types and their characteristics. Whereas, real experiments using UAVs are expensive; therefore, the performances of UAVs systems should be analyzed before their deployments [8] [9]. For this reason, many researchers [10] [11] utilize the Software In The Loop (SITL), which is a simulation that gives the opportunity to operate Plane, Copter or Rover, without the need for any hardware. It consists of a build of the autopilot code, using C++ language, which serves to run an autopilot on your computer directly for testing, consequently, the simulation is self-contained. SITL becomes an exceedingly practical tool since end-product can misbehave in-flight. It influences on avoiding hazard situations and preserving costly equipment being damaged. To connect to the SITL, two ports can be used. The port 14550 using the UDP protocol or the port 5760 using the TCP protocol.

The second component of the UAV systems is the GCS, it has several features that will be detailed as follows.

TABLE I
UAVs CLASSIFICATION

Category	Weight of UAV	Radius of Mission	Endurance	Use case
Micro	<2 kg	5 km (LOS)	A few hours	Reconnaissance Inspection Surveillance
Mini	2-20 kg	25 km (LOS)	Up to 2 days	Surveillance Data gathering
Small	20 -150 kg	50 km (LOS)	Up to 2 days	Surveillance Data gathering

Mission planning: GCS prepares the mission plans and paths for the UAV, according to the environment and mission requirements, then, UAV has to achieve the mission depending on the planned trajectories.

Navigation and position control: During the mission, UAVs are placed in several positions at different altitudes, in order to check out the target area. For that, GCS has to display and control the movements of the UAVs to succeed the mission.

Payload control: UAVs are mounted by devices such as cameras, sensors etc. GCS must supervise the payload parameters during the mission.

Communication and data exchange: GCS and UAVs should have a direct and bi-directional communication between them. The GCS sends commands and orders for the UAV according to the mission and the UAV sends telemetry and data (images, videos, etc.) to the control station.

The communication links between the various nodes is a necessary component in the flight systems. In fact, there are two different types of links: UAV-UAV and UAV-GCS. UAV-UAV communication link ensure the collaboration and the coordination between UAVs to improve the performance of the system. UAV-GCS communication links ensure the transmission of signals between the GCS and UAVs which are classified as: data signals and control signals. The data signals are collected from onboard sensors of the flight system including cameras, GPS, microphones, lidars, gas sensors, etc. Whereas, the control signals are required for take-off, landing and piloting the UAV in general. Control signals are mainly sent from the ground control station (GCS) to the UAVs and the data signals are transmitted from the UAVs to the GCS. In order to have a smooth transition from the simulated system to the the real world system, several researchers has performed flexible hardware-in-the-loop (HITL) architectures [16], [17]. To evaluate and analyze the performance of the UAV systems, we need to integrate flight simulator with GCS and UAVs emulators.

To the best of our knowledge, this is the first report that address and analyze the performance of both flight simulators and GCS. However, many reviews have been performed for flight simulators. Authors in [18] studied and evaluated UAVs simulators including several types UAVs on the basis of many considerations. Researchers in [19] analyzed simulators which were designed specifically for the Autonomous Underwater Vehicles simulation, but many of the reviewed simulators have been used in internal academic projects and they are unavailable for public use.

This study determines and highlights the most adequate simulator for UAVs flights and UAVs managements. Moreover, it details the requirements, the goals, the strengths and the weaknesses of each studied tools on the basis of several factors.

The remainder of this paper is structured as follows. Section II identifies the most popular UAVs simulators and Section III examines some performing ground controls station. Section IV presents a case study using the best UAVs simulator and GCS. Section V gives some concluding remarks.

II. UAVs SIMULATORS

The UAVs simulator should create an environment for UAVs to fly in, allowing a multitude of sensors such as cameras, lidars, GPS, microphones, gas sensors, etc. There is no unique simulator suitable for all aims. The choice of the adequate simulator depends on both the purposes and the different features offered by the simulator. In many applications, it is necessary to include physical aspects like accurate turbulence, air density, wind shear, cloud, precipitation, and other fluid mechanics constraints. Indeed, several flight simulators were created to test hardware in the loop (HITL).

Several UAVs simulators (e.g. XPlane [12], Flightgear [14], Gazebo [21], JMavSim [20])were focused on simulating the UAV physics in order to train pilots of the right controller. Whereas, they used limited assets and textures, and did not support the motion of capture (MOCAP) which allow simulating the UAVs natural movement. Recently, Microsoft Airsim [27], and UE4Sim [28] are created for supporting MOCAP showing UAVs motion planning. They are based on the Unreal Engine 4 (UE4). UE4 is an open-source tool for simulating the UAVs natural movement using modern high-quality engine and realistic physics library. Fig. 2. demonstrates a classification for the aforementioned available UAVs simulators.

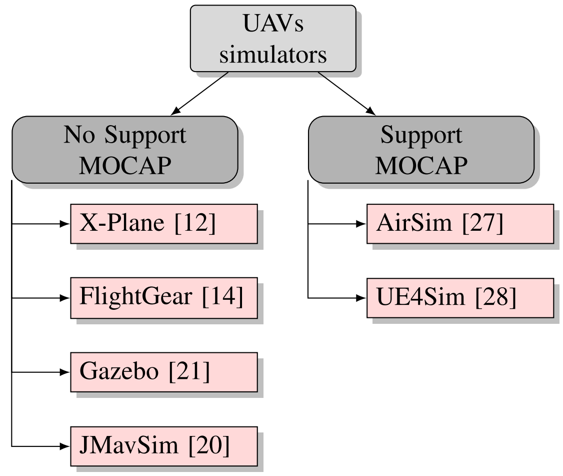

Fig. 2. Classification Flight Simulators

A. X-Plane

X-Plane [12] is developed by Laminar Research. This simulator is commercially available and it has specific builds for a diversity of operating systems including Windows, Linux, and macOS, and there is a mobile version for Android, iOS, and WebOS. X-Plane is packaged with many commercial, military, and other aircraft, using basic global scenery that covers most of the Earth. It has a plug-in architecture which allows users to create others modules, extending the functionality of the simulator. It received FAA certification and it is used with some hardware configurations to increase its fidelity to the flight model. For multiplayers flight simulation, X-Plane can connect other X-Plane instances via TCP/IP or UDP/IP protocols, using multi-monitor X-Plane configurations such as Pilot Edge [13].

The X-Plane employs the geometric shape created in Plane Maker to show how the UAVs will fly. It uses an engineering mechanism that means to divide the UAVs into small elements and find the forces acting for each element. This simulator allow the visualization of different forces acting on the UAVs, the UAVs path traveled and, also, allows the establishment of the flight failures. The native communication protocol considered by X-Plane simulator is the UDP protocol.

B. Flight Gear

FlightGear Flight Simulator [14] is a free, open source multi-platform flight simulator implemented by the FlightGear project since 1997 and continued in development. This simulator has specific builds for a diversity of operating systems such as Windows, Mac, OS, Solaris and various distributions of Linux such as Ubuntu and Debian. etc. FlightGear code is released under the GNU General Public License. UAVs models must be created by an external 3D modelling application with an XML file illustrating the various UAVs features. FlightGear

can be used on the local network thanks to the multiplayer protocols developed for the multi-aircraft environment. Moreover, different multiple UAVs initialized in various instances for multiple mode can coordinate and communicate in the ambient environment. This feature can be used for air traffic control or formation flight simulation.
FlightGear permits to choose between three Flight Dynamics Models (FDMs). In fact, an FDM contains some mathematical equations which are used to compute the physical forces acting in a simulated aircraft. Forces could be drag, thrust, and lift. Every simulated aircraft must use one type of the FDMs models. Examples of the popular FDMs are YASim, JSBSim, and UIUC. FlightGear can run SITL and HITL and it is used for various academic projects [15].

C. JMavSim

JMavSim [20] is a multirotor simulator and it is developed by PX4. It is a simple and lightweight quadrotor simulator and it can run SITL and HITL. As shown in Fig. 3. JMavSim has a Micro Aerial Vehicle Link [37](MAVlink) interface and is continuously incorporated with the PX4 firmware. It can be integrated with both ROS and flight controller firmware. JMAVSim graphical aspect is created by Java3d. It uses the UDP protocol for communication. Whereas there is no easy way to integrate extra sensors or obstacles in the simulation, and there is no possibility of simulating a visual inspection.

Fig. 3. Flying a multirotor in JMavSim

D. Gazebo

Gazebo [21] is a simulator implemented at the University of Southern Californa since 2002; since 2009 it has been used with ROS [22]. Gazebo is developed for quadrotor and any robot. It uses the Ogre3D visualisation tool and permits the choice between four physics engines: ODE [23], Simbody [24], DART [25], and Bullet [26] giving it great flexibility. Gazebo support Ardupilot and PX4 with the possibility to run SITL and HITL. Moreover, it is easy to create worlds and models in Gazebo which are implemented in a custom XML format named Simulation Description Format (SDF). Besides, models can be imported from meshes developed in other programs likes Inkscape or Solidworks, with the need some extra information about inertia and mass. It is possible to import Digital elevation models (DEM) or heightmaps.

E. Microsoft AirSim

Microsoft AirSim [27] is a simulator developed by Microsoft for the machine learning development. The simulator is revealed in February 2017. The simulator uses Unreal Engine 4 to render the simulation more photo-realistic as illustrated in Fig. 4. The simulator gives the possibility of interfacing with Mavlink, and it is possible to run SITL and HITL with the pixhawk firmware such as Ardupilot and PX4. There exist APIs for both Python and C++. Airsim has implemented monocular and depth cameras, whereas, other sensors, like lidars, are not developed. The Airsim simulator is only a quadrotor simulator, other UAVs are not supported. The great visuals can be a downside because it demands powerful GPUs to run.

Fig. 4. Flying a multirotor in Microsoft AirSim. The three boxes at the bottom are depth camera, depth segmentation, and First Person View (FPV)

F. UE4Sim simulator

UE4Sim [28] simulator is developed in 2017 at the King Abdullah University of Science and Technology. It is used in various fields of computer vision such as object tracking, object detection, autonomous navigation, multi-agent collaboration, etc. UE4Sim is built on the Unreal Engine 4 and it generates photo-realistic simulation for autonomous cars and UAVs as shown in Fig. 5. UE4Sim allows access not only to visual data captured from sensors mounted in the simulated environment but also semantic information.

III. Ground control station

In fact, there is a large variety of GCS software application which runs on a ground-based computer such as QGroundControl [29], Mission Planner [30], MAVProxy [33], UGCS [32], that will be studied in this article.

A. QgroundControl

QgroundControl is an open source ground control station (GCS) developed by Lorenz Meier [29] and written in C++ using the Qt libraries . This GCS operate on different platforms

TABLE II
COMPARISON OF AVAILABLE UAVS SIMULATORS

Simulator	FlightGear	X-Plane	JMavSim	Gazebo	Air Sim	UE4Sim
Commercial / free	Free	Commercial	Free	Free	Free	Free
Vehicles	Airplanes UAVs Some multirotor	Airplanes UAVs Some multirotor	Multirotor	Multirotor and Any robots	Multirotor	Multirotor, cars
Interface ROS	No	No	Yes	Yes	No	No
Sensors	Diversity of sensors	Easy incorporation of sensors	No incorporation of sensors	Easy modification of sensors	Monocular, depth cameras No lidar	Easy modification of sensors
Motion capture	No	No	No	No	Yes	Yes
Obstacles	Yes	Yes	No	Yes	Yes	Yes
SITL-HITL	Yes	Yes	Yes	Yes	Yes	No
MAVLink	Yes	Yes	Yes	Yes	Yes	No
Ease of Development	Medium	Medium	High	High	Medium	Medium

Fig. 5. Flying a multirotor in UE4Sim

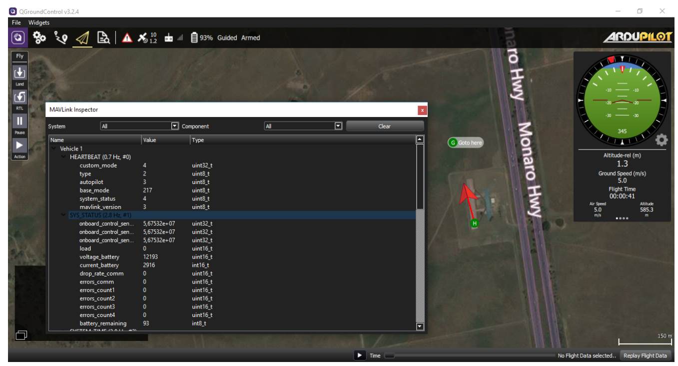

Fig. 6. Graphical interface of QgroundControl station

such as Windows, Mac OS X, Linux, Android and iO. It supply configuration for both PX4 Pro and ArduPilot firmware supports the MAVlink protocol (which employed to communicate with micro air vehicle) and offers the opportunity to visualize details of the MAVLink protocol messages, exchanged between it and the flying vehicle. Moreover, QGroundControl proposes a graphical interface, which includes 2 D map, to facilitate the management of one or multiple UAVs and to control the location of drones. Additionally, it provides video streaming, displays the vehicle position, waypoints, etc; and offering the possibility to create a mission for autonomous flight .

B. Mission Planner (MP)

This GCS is developed by Michael Oborne using Python Programming Language. Unlike QgroundControl which is compatible with all platforms, MP is compatible with Windows only [30]. it provides a graphical interface which displays information about the UAV like GPS status, battery, airspeed, video etc [31], additionally, MP allows to download the log files of a mission and examine them [30].

C. Universal Ground Control Software (UGCS)

UGCS [32] is a desktop software with a graphical interface that allows users to create a mission, to supervise UAVs, to be informed about the state of the vehicle and to follow the telemetry. The efficiency of the UGCS is that it supports various autopilots such as DJI, Ardupilot, Px4, Micropilot, Microdrones, and other MAVLink compatible, etc. This product has a solid support which contains articles, tutorials and videos. The free version of UGCS is limited and it is dedicated to beginners for a simple test or a simple mission within line of sight. For the proper functioning of UGCS, it is recommended to operate it in an environment with the following configurations: 2 GB RAM minimum, 2 GB of free space in the hard disk and a processor Core 2 Duo or Athlon X2 at 2.4 GHz.

D. MAVProxy

MAVProxy is an extensible command-line ground station written in python [33] which can be run on different OS such as Linux, OS X, Windows, and others. MAVProxy manages any UAV supporting the MAVLink protocol by using Micro Air Vehicle Marshaling/Communication library [34]. Furthermore, it provides different modules, like console module that displays information about the UAV's current state and map

TABLE III
COMPARISON BETWEEN GCS SOFTWARE

GCS software	QGroundControl	Mission Planner	UGCS	MAVProxy
Interface	Graphical	Graphical	Graphical	Command
Commercial / Free	Free	Free	Free version with limited capabilities	Free
Support MAVLink	Yes	No	Yes	Yes
Platform for Android	Yes	No	Yes	No
Pilot	PX4 Pro ArduPilot(APM) MAVLink compatible	ArdupilotPX4	Dji, Innoflight Micropilot Mikrokopter Microdrones Parrot MAVLink compatible	Ardupilot MAVLink compatible

module that shows the UAV's current position and waypoints. MAVProxy is commonly employed by developers to interact with SITL. It has the ability to forward messages from a UAV via UDP protocol to others GCS on other devices [35]. MAVProxy used in [36] to implement attacks on UAV system in order to exploit gaps and vulnerabilities of the MAVLink protocol.

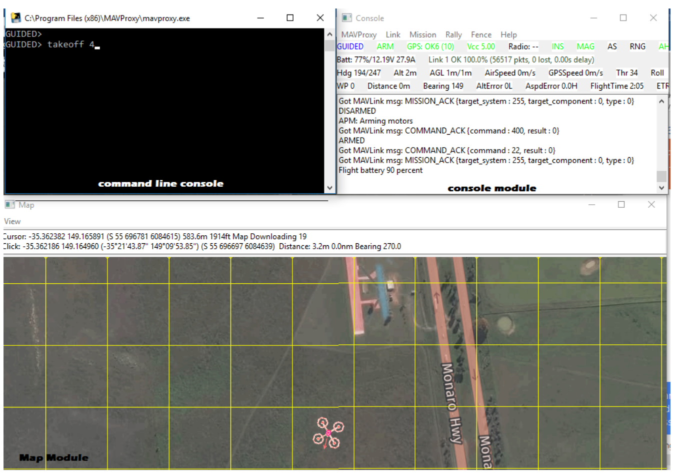

Fig. 7. MAVProxy command-line with console module and map module

IV. CASE STUDY

The aforementioned GCSs and simulators have been evaluated and compared previously, in order to choose the most suitable for our use case which consists on exchanging MAVLink messages. The MAVLink protocol is used as it is supported by all system components. It is applied for the exchange of data between the UAV and the GCS.

We have chosen QgroundControl station for the reason that it has a graphical interface which contains many features. Moreover, it is the unique one which allows recording video captured by the UAV. Equally, we have selected FlightGear as UAV simulator in view of its flexibility in modeling the UAV features described in an XML file. Furthermore, its code source is accessible for modification.

We used the ArduPilot, more specifically an UAV copter as a SITL to testing. Afterwards, we simulated an environment (Fig. 8.) and choose to use the FlightGear to visualize the simulation in real time and in a real environment, with its opportunity to display the scene in 3D. And for the management and the supervision of the UAV, we employ the QgroundControl station.

Several studies [38] prove that The MAVLink protocol has security gaps. For this purpose, this simulation is performed in order to send and receive MAVLink messages, like in the real case. Thus, we will sniff the network traffic, decode the messages exchanged to recognize the weaknesses of this protocol, in order to modify and secure it.

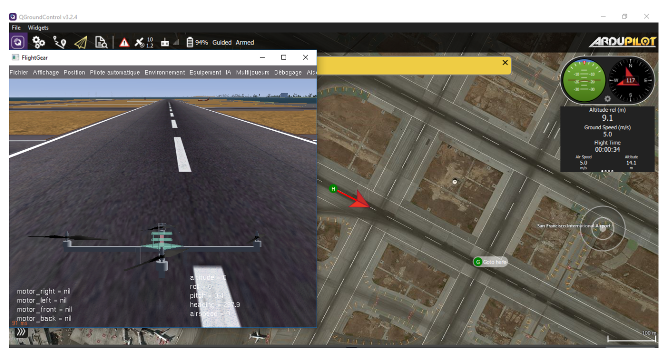

Fig. 8. Use case: Exchange of data between FlightGear and QgroundControl

A test was performed using the ardupilot SITL. A command 'takeoff' was sent from the QgroundControl to the UAV, thus, FlightGear simulate immediately.

At the same time, we looked at the MAVLink inspector, which is a feature in the GCS, and we perceived the details of exchanged messages (Fig.9). The MAVLink messages are detailed in [39], they must be understood in order to make the correspondence.

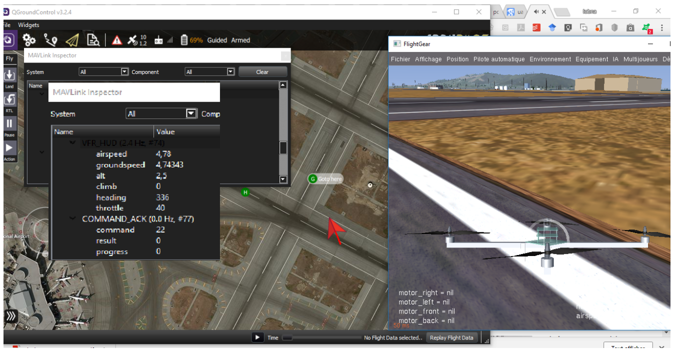

Fig. 9. Exchange of command between FlightGear and QgroundControl

V. Conclusion

In this paper, we presented a study about the most interesting flight simulators and GCS that can be employed to modelling and managing UAVs system, in order to facilitate the choice for the researchers and developers in their works. For our future researches, our choice will be focusing on using the QgroundControl station as GCS and FlightGear as flight simulator.

Acknowledgment

This work is supported by the Tunisian program "Tunisian Federated Research Project" within the framework of the project Supervision Sensitive de lieux Sensibles multicapteurs: Super-Sens .

References

[1] He, Daojing, Sammy Chan, and Mohsen Guizani. "Communication security of unmanned aerial vehicles." IEEE Wireless Communications 24.4 (2017): 134-139.

[2] Acevedo, Jose Joaquin, et al. "Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions." Journal of Intelligent & Robotic Systems (2013): 1-17.

[3] Maza, Ivn, et al. "Experimental results in multi-UAV coordination for disaster management and civil security applications." Journal of intelligent & robotic systems 61.1 (2011): 563-585.

[4] Bruzzone, Agostino, et al. "Disasters and Emergency Management in Chemical and Industrial Plants: Drones Simulation for Education and Training." International Workshop on Modelling and Simulation for Autonomous Systems. Springer, Cham, 2016.

[5] Barrado, Cristina, et al. "Wildfire monitoring using a mixed air-ground mobile network." IEEE Pervasive Computing 9.4 (2010): 24-32.

[6] Albert, Anders, and Lars Imsland. "Performance bounds for tracking multiple objects using a single UAV." Unmanned Aircraft Systems (ICUAS), 2017 International Conference on. IEEE, 2017.

[7] Shrinivasan, Lakshmi. "Monocular Imaging-based Autonomous Tracking for Low-cost Quad-rotor Design-TraQuad." arXiv preprint arXiv:1801.06847 (2018).

[8] Hodicky, Jan. "Autonomous systems operationalization gaps overcome by modelling and simulation." International Workshop on Modelling and Simulation for Autonomous Systems. Springer, Cham, 2016.

[9] Yapp, Justin, Remzi Seker, and Radu Babiceanu. "UAV as a Service: A Network Simulation Environment to Identify Performance and Security Issues for Commercial UAVs in a Coordinated, Cooperative Environment." International Workshop on Modelling and Simulation for Autonomous Systems. Springer, Cham, 2016.

[10] Shastry, Abhishek K., Mangal Kothari, and Abhishek Abhishek. "Development of Flight Dynamics Model of Quadrotor." 2018 AIAA Guidance, Navigation, and Control Conference. 2018.

[11] Coopmans, Calvin, Michal Podhradsk, and Nathan V. Hoffer. "Software- and hardware-in-the-loop verification of flight dynamics model and flight control simulation of a fixed-wing unmanned aerial vehicle." Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2015 Workshop on. IEEE, 2015.

[12] A. Meyer, X-plane. [Online]. Available: http://www.x-plane.com.

[13] Thurber, Matt. "Technology Advances Flight Training Field." Aviation International News (2017).

[14] Perry, Alexander R. "The flightgear flight simulator." Proceedings of the USENIX Annual Technical Conference. 2004.

[15] http://www.flightgear.org/Projects/747-JW/

[16] Lepej, Peter, Angel Santamaria-Navarro, and Joan Sola. "A flexible hardware-in-the-loop architecture for UAVs." Unmanned Aircraft Systems (ICUAS), 2017 International Conference on. IEEE, 2017.

[17] Worth, Derek B., Brian G. Woolley, and Douglas D. Hodson. "SwarmSim: a framework for modeling swarming unmanned aerial vehicles using hardware-in-the-loop." The Journal of Defense Modeling and Simulation (2017): 1548512917715154.

[18] J. Craighead, et al. A survey of commercial & open source unmanned vehicle simulators. Robotics and Automation, 2007 IEEE International Conference on. IEEE, 2007.

[19] O. Matsebe, C. M. Kumile, and N.S. Tlale. A review of virtual simulators for autonomous underwater vehicles (auvs). NGCUV, Killaloe, Ireland (2008).

[20] Available : https://pixhawk.org/dev/hil/jmavsim.

[21] Available : http://gazebosim.org.

[22] http://www.ros.org

[23] Smith, Russell. "Open dynamics engine." (2005): 3-2.

[24] Sherman, Michael A., Ajay Seth, and Scott L. Delp. "Simbody: multibody dynamics for biomedical research." Procedia Iutam 2 (2011): 241-261.

[25] Wang, Danyao, Natalie Enright Jerger, and J. Gregory Steffan. "DART: Fast and flexible noc simulation using FPGAs." WARP-5th Annual Workshop on Architectural Research Prototyping. 2010.

[26] Coumans, Erwin. "Bullet physics engine." Open Source Software: http://bulletphysics. org 1 (2010): 3.

[27] Shah, Shital, et al. "Airsim: High-fidelity visual and physical simulation for autonomous vehicles." Field and Service Robotics. Springer, Cham, 2018.

[28] Mueller, Matthias, et al. "UE4Sim: A Photo-Realistic Simulator for Computer Vision Applications." arXiv preprint arXiv:1708.05869(2017).

[29] https://docs.qgroundcontrol.com/en/

[30] http://ardupilot.org/planner/docs/mission-planner-overview.html

[31] http://ardupilot.org/planner/docs/mission-planner-ground-control-station.html

[32] https://www.ugcs.com/en/supported_drones_autopilots.

[33] Andreescu, A. M. T., Dima, M., Istrate, A., Muresan, C., Visoiu, C., & Prvu, P. Autonomous System For Image Geo-tagging And Target Recognition.

[34] Mitra, S., & Land, B. (2013). Autonomous quadcopter docking system. Cornell University.

[35] http://ardupilot.org/dev/docs/mavproxy-developer-gcs.html

[36] Butcher, N., Stewart, A., & Biaz, S. (2013). Securing the mavlink communication protocol for unmanned aircraft systems. Appalachian State University, Auburn University, USA

[37] http://qgroundcontrol.org/mavlink/start

[38] Marty, J. A. (2013). Vulnerability analysis of the mavlink protocol for command and control of unmanned aircraft (No. AFIT-ENG-14-M-50). Air Force Institute Of Technology Wright-Patterson AFB Oh Graduate School Of Engineering And Management.

[39] http://mavlink.org/messages/common