SIMULATION FRAMEWORK FOR TESTING PERFORMANCE OF MULTI-UAV MISSIONS

Ernesto Santana[(a)], Romualdo Moreno[(b)], Christian Sánchez[(c)], Miquel Àngel Piera[(d)]

[(a),(b),(c),(d)] Dept. Of Telecommunication and Systems Engineering,
Autonomous University of Barcelona, Spain

[(a)]ernestoemmanuel.santana@uab.cat, [(b)]romualdo.moreno@uab.cat, [(c)]sanchezcsa@gmail.com
[(d)]miquelangel.piera@uab.cat

ABSTRACT

In this paper, we present a Software-in-the-Loop simulation framework for multi-UAV systems which allows test the implementation of cooperation algorithms, sense & avoid algorithms, etc., without taking any risk. In our proposed framework, we can test the correct execution of algorithms, and since our setup is inherently multi-UAV, we can also test the communication flow among the vehicles. The simulation framework has been developed as a functional extension of Mission Planner, open source software aimed to plan, control and analyze data from missions of a single UAV driven by Ardupilot flight controllers.

Keywords: SIL, multi-UAV, Mission Planner, Simulation Framework

1. INTRODUCTION

In recent years the research in the field of multi-UAV systems has received increased attention, due to that this option allows to perform missions set with greater efficiency. This new technologies offer many potential applications in multiple fields such as infrastructure inspection, monitoring coastal zones, traffic and disaster management, agriculture and forestry among others. The coordination of a team of autonomous vehicles allows accomplish missions that no individual autonomous vehicles can accomplish on its own when certain time constraints have to be satisfied. Team members can exchange sensor information, collaborate to track and identify targets, perform detection and monitoring activities (Ollero and Maza 2007), or even actuate cooperatively in tasks such as the transportation of loads.

However, multi-UAV missions are much more complex. The development of algorithms for multi-UAV systems requires performing tests which implies a high level of risk for UAVs. Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) simulations are a good way to test these aspects without a need of real UAV experiments.

Hardware-in-the-Loop is a well established methodology for testing of automotive software under simulated conditions. One advantage of HiL simulation is that the validation of control results is very straightforward. In industrial control, hardware-in-the-loop simulation techniques can significantly reduce the time required to design controllers and can increase the reliability of the systems (Xue and Chen 2013). However, it requires the software to run on an electronic control unit (ECU), and the vehicle components (e.g. sensors and loads) to be either available physically or simulated accurately. The tests based on HiL simulation are limited: occurs relatively late, it is expensive, has slow turn around times, limited scalability, and provides in practice only quite limited coverage. On the other hand, the Software-in-the-Loop methodology has emerged as an alternative approach to embedded software testing. A SiL system of simulation requires no physical hardware and can reduce the time and costs of new software development considerably.

In this paper we present a Software-in-the-Loop simulation framework for multi-UAV systems, which consists of two main parts. The first one is formed by a group of virtual machines that simulate the fleet of UAV Software-in-the-Loop. And the second one is the open source Ground *Control* Station (*GCS*) software *Mission Planner* for planning and executing UAV missions. The source code of Mission Planner is modified to handle multi-UAV systems.

2. BACKGROUND

One of the main research topics involved in the study of multi-UAV systems focuses on modeling the problem as a multi-agent system. Therefore, most multi-UAV simulators are used only as testbeds for cooperative models and algorithms.

The commercially available X-Plane flight simulator, together with MATLAB, are used by Garcia and Barnes (2009) to create a simulator framework for studying multi-UAV control algorithms. Pujol, Cerquides, and Meseguer (2014), proposed a multi-agent simulation environment to investigate decentralized coordination for teams of UAVs. Odelga, Stegagno, and Ahmad (2015) present a setup for HIL multi-UAV simulations which is based on Gazebo, a popular open source ROS-enabled.

Other research lines around the field of multi-UAV simulations include the study of the best interface or set of interfaces for the operator to monitor the status of all UAVs. Related to this, the company Silicon Valley Simulation, specialized in real time visual simulation since 1996, has developed MUSIM (Multiple UAV Simulation) [30]), a flexible and modular UAV simulation environment used for research into the operator interface.

3. ARCHITECTURE FRAMEWORK

This section provides a brief overview of the configurations of the architecture framework *proposed in this paper*. A block diagram of the simulation framework is depicted in figure 1.

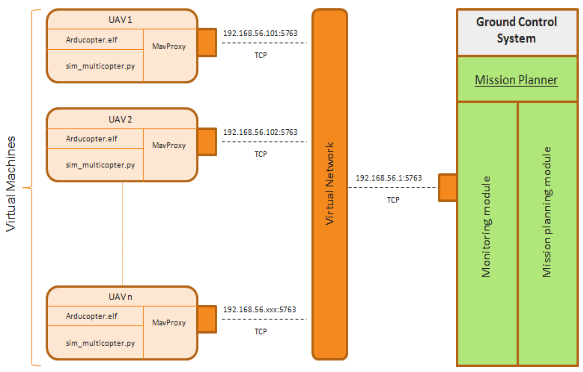

Figure 1: Scheme of the Software-in-the-Loop simulation framework for multi-UAV

3.1. Simulated UAVs

The fleet of UAVs is created from virtual machines that run the software-in-the-loop simulation. The SiL simulator used was developed by the community of APM as a tool for analysis of APM firmware: Copter and other variants. It allows execution of source code without connecting hardware, emulating all modules involved (low-level drivers of different sensors) [see figure 2]. In addition, it works by performing a compilation of the complete code, which runs in conjunction with a simulation of the physics of the ship.

The source code is compiled into an executable file ArduCopter.elf, which is invoked in conjunction with the sim_multicopter.py simulator that is written in Python language. These two programs interact with data packets sent through the UDP ports 5501-5502. On the other hand, the main program also interacts through the communication protocol Mavlink, with MAVProxy program. MAVProxy is a powerful command-line based “developer” ground station software and is responsible for establishing communication between the simulated UAVs and GCS software, using the TCP / UDP protocol [see figure 3].

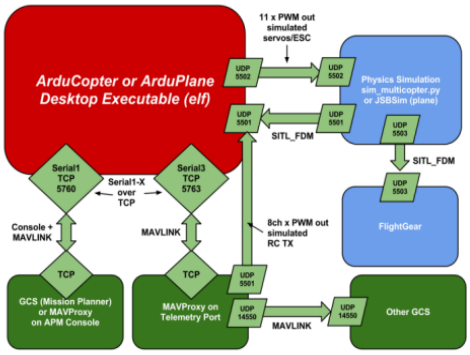

Figure 2: Architecture of the SiL simulator

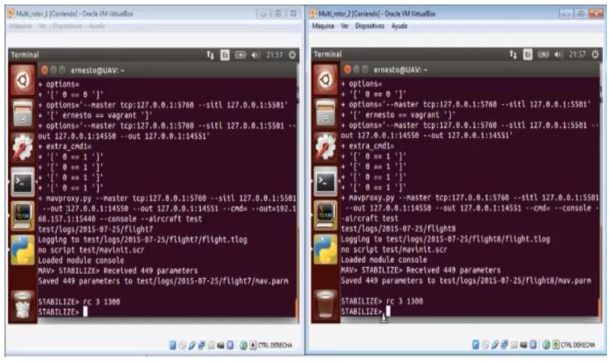

Figure 3: *MAVProxy running* under Ubuntu Virtual Machines

3.2. The Ground Control Station (GCS)

On the other hand, we have selected the open source software Mission Planner to perform the functions of control, planning and monitoring. This package software is, by far, the most used open source software for configuring and planning autonomous missions. This results in a large online community with many resources for developing an autonomous vehicle.

Mission Planner, created by Michael Oborne, is a software package that allows us to configure the APM’s settings for our particular airframe setup, so we can achieve the best stability and control for manned and unmanned flights. This software package also supports calibration of the compass and our accelerometers. Mission Planner also allows us to test our motors as well as our sensors such as the sonar module. Mission Planner allows us to plan missions with waypoints planned using a Google Maps interface within the software. Mission Planner makes this very easy with point-and-click waypoint entry. We also have the possibility to download mission log files for analysis after our missions are completed and the copters have safely returned. This Ground Control Station software can be used as a configuration utility or as a dynamic control supplement for your autonomous vehicle.

As this software is designed to operate with a single UAV, we have modified your source code for providing it with the ability to control multiple UAVs. We have created a multiple connection module, which allows simultaneous communication with several UAVs using TCP / UDP and Serial protocols.

To establish the connection between the Mission Planner and each UAV, this module performs the following steps [see figure 4]:

1. Receives the connection request.
2. Identifies the communication protocol used.
3. A MAVLink interface for this UAV is created (this interface allows you to interact with the UAV using the MAVLink protocol). For identification a name according to the order of connection, UAVi, is assigned.
4. The interface is stored in a list called "Comports".
5. All Mission Planner modules (Monitoring, control and planning) point to the "Comports" list to interact with each of the stored MAVLink Interfaces.

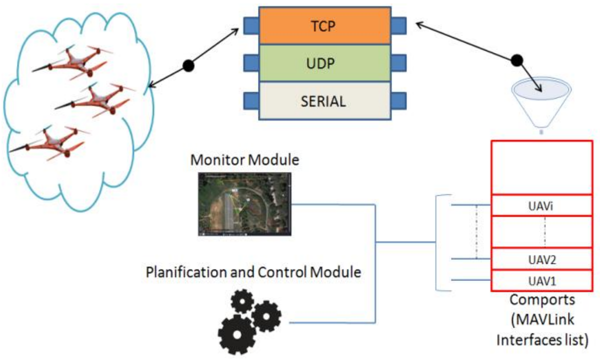

Figure 4: Multiple connection scheme

In figure 5 the graphical user interface of the Multiple Connection module is shown.

A great advantage of using the Mission Planner software in this simulation framework is that all the implemented algorithms can be easily tested in real missions with only replacing virtual machines by real UAVs.

3.3. Communication

To establish communication between simulated UAVs and the GCS we used the VirtualBox Manager to make a Virtual Network. VirtualBox provides several virtual PCI Ethernet cards for each virtual machine. For each card, you can individually select the hardware that will be virtualized as well as the virtualization mode that the virtual card will be operating in, with respect to your physical networking hardware on the host.

Figure 5: Graphical user interface of the Multiple Connection module

In our case, we selected the Intel PRO/1000 MT Desktop adapter and the Host-Only networking mode for each simulated UAV, as shown in figure 6.

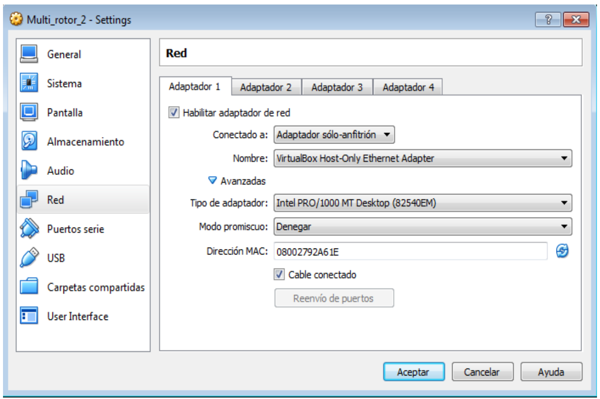

Figure 6: Virtual PCI Ethernet card setting

This configuration is particularly useful for preconfigured virtual applications, where multiple virtual machines are shipped together and designed to cooperate. Instead, a virtual network interface (similar to a loopback interface) is created on the host, providing connectivity among virtual machines and the host [see figure 7].

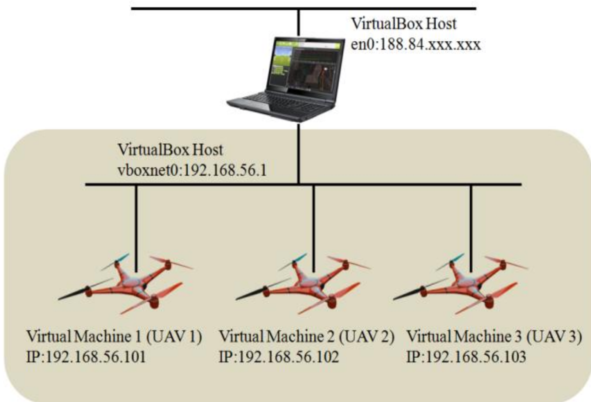

Figure 7: Software in The Loop Simulation Network

4. APPLICATIONS

The proposed framework has been used to assess the feasibility and performance of some flight missions for multi-UAV systems.

Also, this simulation framework has proven useful in the development and implementation of various types of algorithms for multi-UAV systems. As it is based on open source software, it is possible to access each of the features of the environment and modify existing features or create new ones. A simple way to implement algorithms for multi-UAV systems is by using Python scripts directly in Mission Planner.

One of the algorithms implemented using this framework has been a Sense & Avoid algorithm for multirotor UAVs based on TCAS II (traffic collision avoidance system II). For this implementation, we have relied on the TCAS-II Resolution Advisory Detection Algorithm presented in Munoz and Chamberlain 2013 [see figure 8].

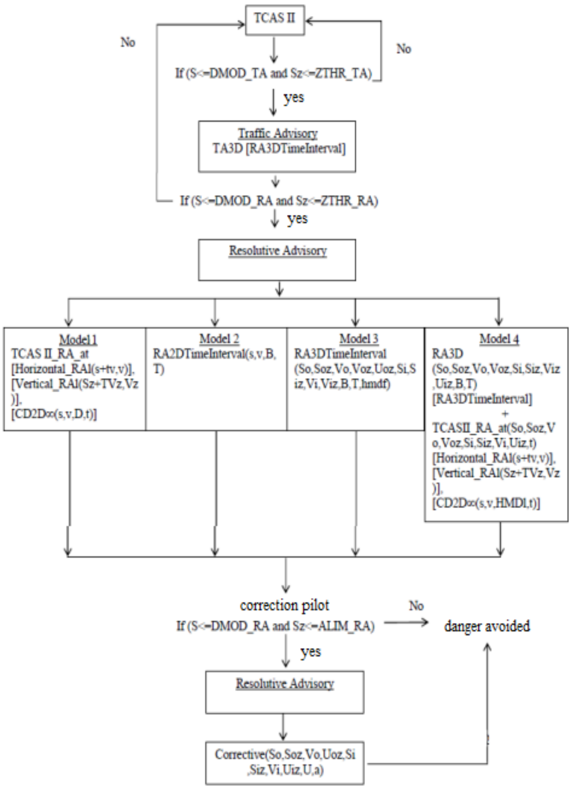

Figure 8: Flowchart of the TCAS-II Resolution Advisory Detection Algorithm

The TCAS II system provides that in the case of a potential collision, a sounding alert is emitted by the "Traffic Advisory" (TA). This system warns the pilot about every intruder aircraft by a "traffic, traffic" vocal announcement. It does not provide any suggestion avoidance maneuver. Whenever the conflict situation gets worst after a TA alert and the collision seems impending, an audio message and a visual alert are generated by the "Resolution Advisory" (RA). This alert indicates the concerned airplane and suggests an avoidance action to be immediately executed by the pilot.

To implement the TCAS II in UAVs we need to modify those processes involving interaction with the pilot. This is due to the fact that the UAV does not require notice to the pilot to perform evasive action. The aircraft itself can perform the maneuver after receiving the alert. Also, the ranges have been modified for TA and RA alerts due to low speed of multirotors UAVs.

The TA will have a horizontal range of 150 meters and 70 meters for the RA. On the other hand, the minimum vertical separation will not be taken into account for the TA, while for the RA will 5m.

In the case of vertical separation of less than 10 meters the avoidance maneuver is a displacement of 60 meters in the horizontal plane and a descent or ascent of 10 meters.

Figure 9 shows a scenario of conflict implemented in our simulation framework. In this case there is a vertical separation of 2 meters between the two UAVs. In figure 10 the avoidance maneuver is shown. More details about the algorithm can obtained in Sánchez (2016).

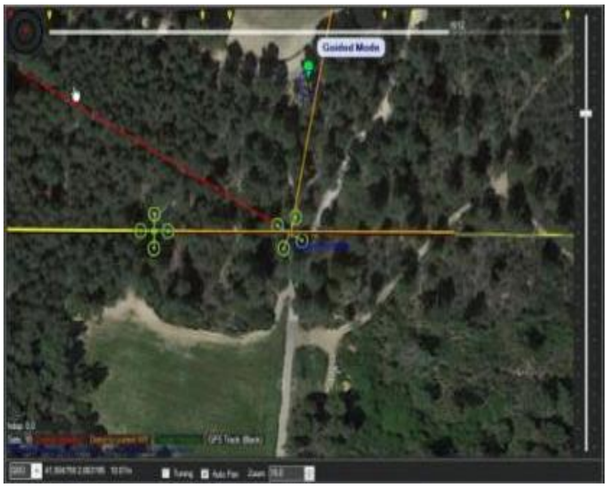

Figure 9: Vertical separation of less than 10 meters

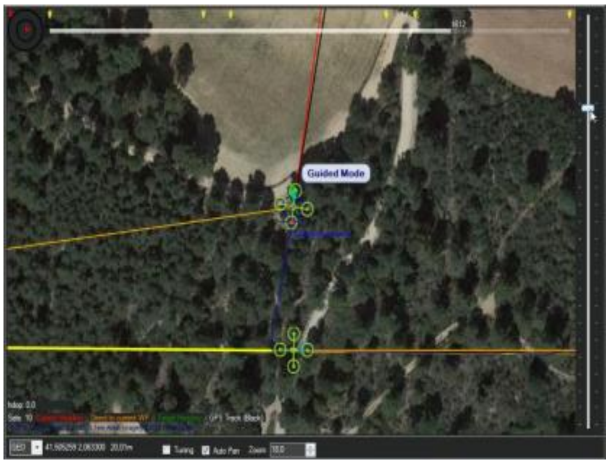

Figure 10: Avoidance maneuver

In addition to this, we have test the simulation framework in other applications. One of them consists in developing and implementing a multi-UAV routing for area coverage in the shortest time possible [see figure 11].

.

Figure 11: Multi-UAV Routing for area coverage with minimum time

This algorithm has been especially designed for applications where it is necessary to cover an area in the shortest time possible, such as to look for hot spots after a fire. The target area is subdivided into subregions it. Each subregion is allocated to a UAV following a strategy that minimizes the overall time of the mission. The algorithms discussed above were programmed directly into the source code of "Mission Planner" code.

Currently we are developing some cooperative algorithms for multi-UAV systems where the simulation framework is also showing its usefulness.

CONCLUSIONS

In this paper, a Software in The Loop Simulation Framework has been presented which allows testing performance of missions and developing different types of algorithms for multi-UAV systems. Additionally, this framework allows implement the communication among UAVs for developing cooperation algorithms.

The proposal has been successfully used for simulating several flight tests including basic flight motions, full-envelope flight and multiple UAV formation flight. Results obtained show that the constructed software-in-the-loop simulation framework is highly effective and useful. This implementation has allowed us to make significant progress in the development and study of task allocation algorithms, sense & avoid algorithms and cooperative algorithms.

FUNDING

This work was supported by the Ministry of Economy and Competitivity in the project ‘‘Fire Guided Unmanned Aircrafts and Resources Distribution - 2 (FireGUARD-2)’’, CICYT Spanish program (TIN2014-56919-C3-1-R)

REFERENCES

Chandler, P. R., & Pachter, M. 1998. “Research issues in autonomous control of tactical UAVs”. Proceedings of American Control Conference, pp. 394-398). Philadelphia (Pennsylvania, USA).

Garcia, R., & Barnes, L. 2009. “Multi-UAV simulator utilizing X-Plane”. Proceedings of the 2nd International Symposium on UAVs, pp. 393-406. June 8–10. Reno (Nevada, USA).

Marlow, T. and Welsh, T. MUSIM: “Multiple UAV simulation (MUSIM) source code documentation”. Retrieved May 13, 2016, from http://www.siliconvalleysimulation.com/muDoxyDocs/main.html

Muñoz, C., Narkawicz, A., & Chamberlain, J. 2013. “A TCAS-II resolution advisory detection algorithm”. Proceedings of the AIAA Guidance Navigation, and Control Conference and Exhibit. Boston, USA.

Odelga, M., Stegagno, P., & Ahmad, A. 2015. “A Setup for multi-UAV hardware-in-the-loop simulations”. Proceedings of Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), pp. 204-210. November 23-25. Cancún, Mexico.

Ollero, A., & Maza, I. 2007. Multiple heterogeneous unmanned aerial vehicles. Berlin Heidelberg. Springer.

Pujol-Gonzalez, M., Cerquides, J., & Meseguer, P. 2014. “MAS-planes: a multi-agent simulation environment to investigate decentralised coordination for teams of UAVs”. Proceedings of the international conference on Autonomous agents and multi-agent systems, pp. 1695-1696. May 05 – 09. Paris, France.

Sánchez, C. 2016 “Adaptation and implementation of TCAS II in unmanned aerial vehicles” (in Spanish). Final B.Sc. thesis: Degree in Aeronautical Management, Universitat Autònoma de Barcelona.

Xue, D., & Chen, Y. 2013. *System simulation techniques with Matlab and Simulink*. United Kingdom. John Wiley & Sons.

3DRobotics Wiki. “Setting up SITL on Linux”. Available from: http://ardupilot.org/dev/docs/setting-up-sitl-on-linux.html [accessed June 25, 2016]

AUTHORS BIOGRAPHY

Ernesto Santana received the B.S. degree in Electronics and Communications from APEC University, Santo Domingo, Dominican Republic, in 2005 and the M.S. degree in Information Technologies and Communi-cations in Mobile Networks from University of The Basque Country, Vizcaya, Spain, in 2011. Currently, he is a PhD student in the Department of Telecommunication and Systems Engineering of the Universitat Autònoma de Barcelona. His current research interest area is the development of unmanned aerial systems, focused in the development and implementation of cooperative systems.

Dr. Romualdo Moreno received his B.Sc. degree in Computer Science from Universitat Autònoma de Barcelona in 1989 and the Ph.D. from the same university in 1994. He has held various positions in the UAB: Vice Dean of the Faculty of Science between 1996 and 1999, Head of the Department of Telecommunication and Systems Engineering between 2003 and 2010, Deputy Director of the Engineering School between 2010 and 2011, among others. Since 2011 he is Coordinator of the Master in Aeronautical Management and since 2014 Deputy director for Sabadell Campus of the Engineering School. For more than 15 years he worked in the development and implementation of advanced control systems and optimized production systems both in the research and in the industry area. More recently, he has developed some research and technological solutions in the air traffic control area. His current research interest area is the development of unmanned aerial systems or RPAS (Remotely Piloted Aircraft Systems), focused in the development and implementation of multi-RPA cooperative systems. He leads the RPAS Laboratory of the UAB.

Christian Sánchez received his B.Sc. degree in Aeronautical Management from Universitat Autònoma de Barcelona in 2016. He developed his final Bachelor thesis entitled “Adaptation and implementation of TCAS II in unmanned aerial vehicles”. Aviation is one of his passions. By the influence of his father who is pilot, he has always been linked to it. He works at Virtual Drone Company, developing unmanned aerial system applications.

Dr. Miquel Àngel Piera is the director of LogiSim, a recognized research group on Modeling and Simulation of Complex Systems and former director for Technical Innovation Cluster. He is Full time associate professor in the Telecomunication and System Engineering Department at Universitat Autònoma de Barcelona. Graduated with excellence from UAB in Computer Engineering (1988), Msc from University of Manchester Institute of Science and Technology in Control Engineering (1991), and he got his Phd in 1993. He is member of the Editorial Board of 3 international journals, has been General Chair of 5 International conferences, and invited speaker in 8 international conferences, has published more than 30 journal and book chapters and more than 100 conference papers.. Dr. Piera has been nominated for the Outstanding Professional Contribution Award 2013 from the Society for Computer Simulation and the William Sweet Smith Prize 2015 by the Institution of Mechanical Engineers (UK).