2018 9th ACM/IEEE International Conference on Cyber-Physical Systems

Formation Control and Persistent Monitoring in the OpenUAV Swarm Simulator on the NSF CPS-VO

Anna Lukina[1], Arjun Kumar[2], Matt Schmittle[3], Abhijeet Singh[2], Jnaneshwar Das[2], Stephen Rees[4], Christopher P. Buskirk[4], Janos Sztipanovits[4], Radu Grosu[1], and Vijay Kumar[2]

***Abstract*— Simulation tools offer a low barrier to entry and enable testing and validation before field trials. However, most of the well-known simulators today are challenging to use at scale due to the need for powerful computers and the time required for initial set up. The OpenUAV Swarm Simulator was developed to address these challenges, enabling multi-UAV simulations on the cloud through the NSF CPS-VO. We leverage the Containers as a Service (CaaS) technology to enable students and researchers carry out simulations on the cloud on demand. We have based our framework on open-source tools including ROS, Gazebo, Docker, and the PX4 flight stack, and we designed the simulation framework so that it has no special hardware requirements. The demo and poster will showcase UAV swarm trajectory optimization, and multi-UAV persistent monitoring on the CPS-VO. The code for the simulator is available on GitHub: https://github.com/Open-UAV.**

I. INTRODUCTION

Unmanned aerial vehicles (UAVs), specifically the multirotor platform, have been rapidly growing in popularity in robotics and cyber-physical systems research. Multirotors are UAVs with four or more rotors enabling hovering and maneuvering similar to a helicopter, but with added stability and a simplified electro-mechanical configuration [1]. While fixed wing and helicopter UAVs do not have the precision or footprint necessary for tasks such as close-range inspection, or smooth videography, multirotors are compact, allow high precision control, and can hover, enabling execution of such tasks. With improved on board computational and sensing capabilities, these platforms are being used for increasingly complex missions, and hardware abstraction and end-to-end simulation tools will accelerate innovation and education as they allow for more time to be spent designing the algorithms than on implementation.[†]

Currently, to develop UAV autonomy or conduct UAV experiments, the developer or researcher starts with simulation and then moves to real robots in Fig. 1 (Top). Fig. 1 (Bottom) shows an example simulation environment. For simulation, the user (i.e., developer or researcher) usually works with the popular tools, ROS and Gazebo, with ROS to communicate with the simulated robot(s) in Gazebo. To run visualization, the user could use Gazebo or rviz (http://wiki.ros.org/rviz). ROS is ideal for simulation as it can port directly to a real robot, requiring little change to run real life testing. The complexity of the simulated environment (e.g., number of objects, lighting, collision checking) and the number of UAVs used, will determine the power of the computer necessary for simulation.

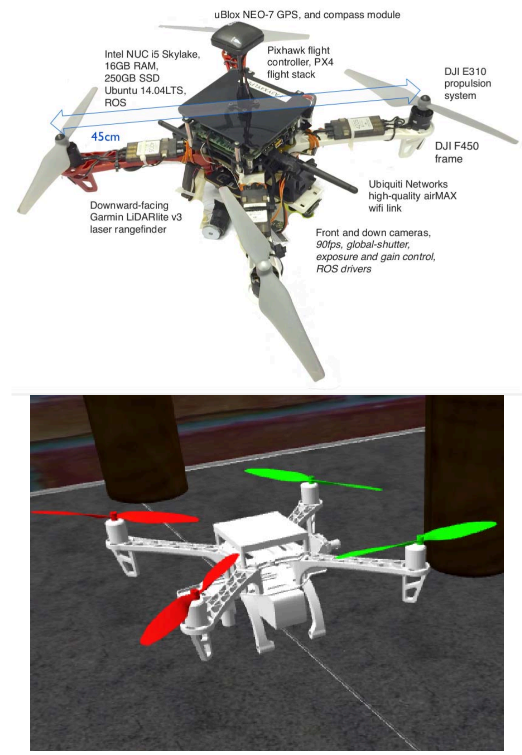

Fig. 1: Top: The DJI F450 airframe and Intel NUC i5 based multirotor UAV model used in the OpenUAV simulator. Bottom: A simulated DJI F450 multirotor as seen on a browser, enabled through Gzweb on the OpenUAV simulator.

The process of setting up these tools requires proficiency in UNIX (or Linux) systems and access to a powerful desktop computer. This barrier to entry can inhibit researchers and stagnate innovation of field systems. In education, these barriers are more prevalent. Students interested in learning about UAVs may lack Linux knowledge, and there may be limited access to a powerful computer for an entire class. The above barriers slow down research and limit education.

In the demo, we will showcase the OpenUAV simulator

[1]A. Lukina and R. Grosu, Cyber-Physical Systems, TU Wien

[2]A. Kumar, A. Singh, J. Das, L. Vacek, and V. Kumar, GRASP Laboratory, University of Pennsylvania

[3]M. Schmittle, Department of Computer Science, University of Delaware

[4]C. P. Buskirk, S. Rees,and J. Sztipanovits, Vanderbilt University

[†]http://usblogs.pwc.com/emerging-technology/lowering-the-barrier-to-innovation-in-robotics/

DOI 10.1109/ICCPS.2018.00050

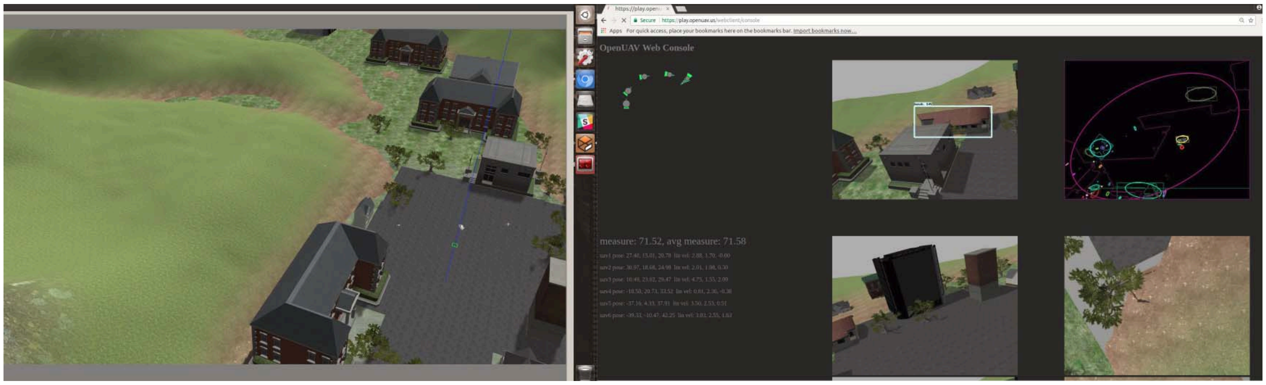

Fig. 2: The OpenUAV simulator showing six UAVs implementing leader-follower behavior.

on the NSF CPS-VO, an open-source, web-based simulation testbed designed specifically for UAVs.

II. Swarm Case Studies

We will showcase two applications benefiting from a cloud-based UAV swarm simulator. The first demo is for plan optimization with multiple UAVs. In this application, a plan is synthesized to bring n agents from an arbitrary initial configuration to a so-called V-formation [2]. This formation has been proven to be energy efficient for birds on long-distance flights. The vortex generated near the wingtips of a bird offers an upwash benefit for the other following birds but only given they avoid downwash formed directly behind the bird as depicted in Fig. 3 (Bottom). The same principle has been exploited by aircraft for fuel conservation on military flight missions. The second demo will showcase formation flights with multiple UAVs for persistent monitoring (Fig. 2).

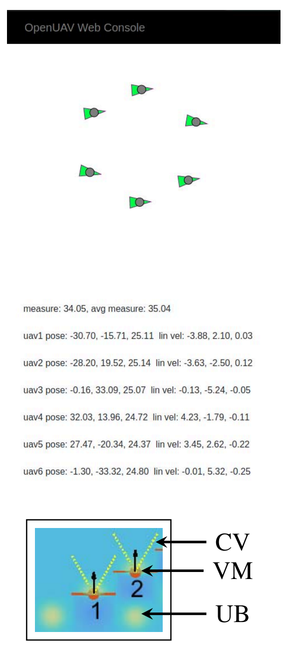

Fig. 3: Top: Screenshot of a simplified 2D visualization for six UAVs in formation on the OpenUAV simulator. Bottom: Optimal positions in V-formation: clear view (CV), velocity matching (VM), and upwash benefit (UB).

Achieving a formation of drones requires emulating the dynamics as close as possible to reality to identify the risks of collisions or interferences between multiple UAVs. We chose V-formation for the multirotors as planning of the close formation flights for them should be downwash-aware as well as for winged vehicles. The current state of the art relied heavily on the environmental conditions and smoothness of the trajectories, which limited the experiments to three drones and wide formations. In contrast, OpenUAV allowed us to simulate larger flocks and study the performance of the algorithm on more challenging starting configurations. Naturally, since aerodynamic considerations are not currently addressed in the OpenUAV stack, it serves as a way for testing and debugging the system before actual field trials.

III. Conclusions

In this paper, we described two proposed demos on the OpenUAV Swarm Simulator using the NSF CPS-Vthe first cloud-enabled, open source simulation testbed for UAV research and education. Based on open source software, we offer this simulator free for public use towards reducing the cost of research and education, and to promote further development of the simulator. By being cloud enabled, we have lowered the barrier to entry to UAV development and research by not requiring specific hardware or complicated setup. The demonstration of multi-UAV formation control and persistent monitoring will showcase to potential users the effective use of this testbed for computationally challenging testing done by a user not involved with development.

Acknowledgment

We gratefully acknowledge NSF grant CNS-1521617, USDA grant 2015-67021-23857 under the National Robotics Initiative, a gift from Microsoft Research, the Doctoral Program Logical Methods in Computer Science and the Austrian National Research Network RiSE/SHiNE (S11412-N23) project funded by the Austrian Science Fund (FWF) project W1255-N23 for supporting this work.

References

[1] G. Shweta, P. Infant Teenu Mohandas, and J. Conrad, "A survey of quadrotor Unmanned Aerial Vehicles," pp. 1–6, 03 2012.

[2] A. Lukina, L. Esterle, C. Hirsch, E. Bartocci, J. Yang, A. Tiwari, S. A. Smolka, and R. Grosu, "ARES: adaptive receding-horizon synthesis of optimal plans," in *Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017*, ser. LNCS, vol. 10206, 2017, pp. 286–302. [Online]. Available: https://doi.org/10.1007/978-3-662-54580-5_17