Advanced Co-Simulation Platform for UAV Simulations Under Virtual Wireless Network Environments

WAN JOO CHO¹, SEONGHEON KIM², YOONSOO KIM³, (Member, IEEE), AND YONG HO MOON³

¹Engineering Research Institute, Gyeongsang National University, Jinju 52828, Republic of Korea
²School of Mechanical and Aerospace Engineering, Gyeongsang National University, Jinju 52828, Republic of Korea
³Division of Aerospace and Software Engineering, Gyeongsang National University, Jinju 52828, Republic of Korea

Corresponding author: Yong Ho Moon (yhmoon5@gnu.ac.kr)

ABSTRACT In this paper, we propose an advanced co-simulation platform that performs the unmanned aerial vehicle (UAV) simulation and the wireless network simulation concurrently. As UAV-based applications become more advanced and complicated and the commands and data transmitted between UAV and ground control station significantly increase, it is vital to safely and robustly transmit the packets carrying commands and data in a wireless network environment. However, the conventional UAV simulation platforms do not take into account the wireless network conditions encountered in the real-world flight so they cannot sufficiently simulate packet transmission. To address this issue, co-simulation platforms have been developed to verify UAV-based applications under a virtual wireless network environment. However, they still cannot effectively support the co-simulation needed to develop the UAV-based applications. In this paper, we have derived a list of improvements to the conventional co-simulation platform based on analysis of the conventional ones. Based on these improvements, we have developed an advanced co-simulation platform consisting of a UAV simulation part, a wireless network simulation part, and an integration part. In this platform, a two-channel approach was introduced to parallelize the synchronization required for performing the UAV simulation and the wireless network simulation simultaneously. The experimental results carried out on three test scenarios demonstrate that the proposed platform accurately performs co-simulation by efficient interoperation of its three parts and by remarkable reduction of the processing time delay, and that it is extendible due to its modularized software architecture.

INDEX TERMS Unmanned aerial vehicle, co-simulation, UAV simulation, wireless network environment, synchronization, software platform.

I. INTRODUCTION

In recent years, unmanned aerial vehicle (UAV) has drawn great attention as an innovative technology driving the industrial revolution. UAV has been employed for a broad range of applications including surveillance, search and rescue, cargo transport, precision agriculture, inspection, environmental monitoring, disaster relief and management, etc [1].

Furthermore, UAV-based applications have become more advanced and complicated by incorporating multi-UAVs with the technologies of big-data, artificial intelligence, and virtual reality [2], [3], [4], [5], [6], [7]. When developing these advanced UAV-based applications, UAV simulation platforms based on virtual UAVs and virtual environments have been utilized to verify and evaluate the mission planning, the flight of UAV, the performance of application, etc. The verifications and evaluations performed on simulation platforms ultimately reduce the risk of accidents during actual
flights as well as prevent loss of cost and time consumption. Thus, an efficient UAV simulation platform is essential for developing UAV-based applications.

In general, the amount of commands and data transmitted between UAV and ground control station (GCS) significantly increase as UAV-based applications become more advanced and complicated. In order to ensure that these applications are reliable and usable, the packets carrying commands and data between UAV and GCS must be safely and robustly transmitted in a wireless network environment. However, packet transmission has not been sufficiently simulated in the conventional UAV simulation platform, although functional behaviors of UAV have been rigorously tested in the conventional one. This clearly shows that the conventional simulation platforms cannot fully test UAV-based applications because they do not effectively take into account the wireless network conditions encountered in real-world flight. As a result, even the functional behaviors of UAV confirmed on the conventional platform cannot be guaranteed to work successfully in the real-world environment. Therefore, a new type of simulation platform capable of testing the functionalities and performances of an application under a virtual wireless network environment is required to develop reliable and practical UAV-based applications.

In order to overcome the limitations of the conventional UAV simulation platform, co-simulation platforms have been suggested in these days [8], [9], [10], [11]. In these simulation platforms, a synchronization scheme has been introduced to provide co-simulation in which UAV simulation and wireless network simulation are performed concurrently. However, these platforms only cover the basic UAV movements such as take-off, hovering, and moving forwards or backwards, and do not support the visualization functions required for observing and investigating the flights of UAV. In addition, they only deal with the packets delivered from UAV to GCS, not vice versa. It means that various types of missions and associated UAV flights cannot be supported on these platforms. Moreover, the simulation results cannot be accurate due to the time delay caused by sequential processing of packet during synchronization. For these reasons, co-simulation platforms are still not suitable for verifying and evaluating UAV-based applications completely. Therefore, they must be further improved to simulate the functionalities and performances of applications as well as the functional behaviors of UAV under a virtual wireless network environment.

In this paper, we propose an advanced co-simulation platform that performs UAV simulation under a virtual wireless network environment. The proposed co-simulation platform consists of three parts which are the UAV simulation (UA VSIM) part, the wireless network simulation (WNETSIM) part, and the integration (ITG) part. The UA VSIM part provides a virtual UAV, a virtual environment, and a GCS to validate functionalities of application and to visualize the flights of UAV. In addition, this part has two channels to separately transmit the location data and the other flight data of UAV. This makes it possible to transmit packets in an up-down link between UAV and GCS and to parallelize synchronization. The WNETSIM part carries out a wireless network simulation with packet of location data under a virtual network environment. In this simulation, packet transmission from UAV to GCS and its communication status are examined and transferred to the ITG part. Both the UA VSIM and the WNETSIM parts are interconnected with the ITG part. The ITG part efficiently manages packet transmission between UAV and GCS by a synchronization scheme based on parallel processing of packets. Through the interoperability of these three parts, the proposed co-simulation platform can simultaneously execute UAV simulation and wireless network simulation. The experimental results demonstrate that the proposed platform effectively performs the co-simulation under various conditions without simulation errors and time delays.

II. BACKGROUND AND RELATED WORKS

UAV simulation platform has been diversely used from modeling the aerodynamics and functional aspects of UAV control system to validating the mission fulfillment and related performance of multi-UAVs. Typically, UAV simulation platform must support virtual environments and virtual UAVs that reflect the characteristics of the specific UAV-based application to be developed. For this reason, open-source software packages such as PX4, Gazebo, X-Plane, and AirSim have been employed to construct UAV simulation platform [12], [13], [14], [15], [16], [17], [18]. AirSim has been employed to test autonomous flights of UAVs because it provides a flight simulation environment as well as the machine learning data needed for deep-learning technologies [14], [17], [19]. Especially, PX4 and Gazebo are so useful for building UAV simulation platforms because they provide various APIs and commands and have an active community. If these software packages are incorporated with an open-source GCS, it is easier to observe and evaluate missions and functional behaviors of UAV. However, conventional platforms based on PX4 and Gazebo can only test the functionalities and performances of an application without considering network environments.

The reliability and usability of UAV-based applications are dependent on wireless network environments because the wireless network conditions greatly influence the transmission of packets containing command and data. Hence, it is essential to simulate the packet transmission and related operations between UAV and GCS under various wireless network conditions. For this purpose, open-source network simulation tools such as NS-3 and OMNet+ have been introduced to UAV simulations [20], [21]. NS-3 is one of the commonly used simulation tools in the network field and it is useful for simulations of packet delivery in virtual environments for Wi-Fi wireless communications. For instance, this tool has been utilized to verify video transmission in UAV systems and build virtual network environments for supporting simulations of swarm UAVs [22], [23]. However, it cannot visualize the flight of UAV or evaluate the performance of the application. In order to successfully simulate an
UAV-based application, the functional behaviors of UAV and the packet transmissions between UAV and GCS should be simultaneously tested.

Recently, co-simulation platforms have been suggested to carry out UAV simulation and wireless network simulation simultaneously. A simple co-simulation platform called CUSCUS has been suggested to simulate multiple UAVs-based control algorithms [8]. However, it cannot be useful for evaluating the effect of the network environment on the operations of the UAVs because the UAV control loop with GCS module is discarded from the network simulation. Koutsoubelias et al. have constructed an UAV simulation environment capable of testing packet transmission using Ardupilot and NS-3 [9]. However, it has disadvantages in that the missions to be tested are limited and visualization function is not supported. In addition, it is impossible to perform UAV simulation and wireless network simulation concurrently. Based on Ardupilot, NS-3, and ZMQ, a co-simulation platform called FlyNetSim has been developed to support UAV simulations under various wireless communication conditions [10]. However, this platform only simulates simple UAV movements such as takeoff, landing, and moving forward without visualization. Furthermore, only packet delivery from UAV to GCS is supported and the simulation results cannot be accurate because of time delays occurred in the synchronization process where data packet is sequentially processed. Another co-simulation platform called CORNET has been built by introducing an open-source ROS to FlyNetSim [11]. In this platform, only the packets from UAV to GCS are available and the UAV simulation results also suffer from time delays during the synchronization process. These limitations clearly show that the conventional co-simulation platforms must be improved to perform the UAV simulation and the wireless network simulation at the same time.

III. PROPOSED CO-SIMULATION PLATFORM

In this section, we examine the shortcomings of the conventional co-simulation platforms and propose an advanced co-simulation platform that overcomes them. In addition, we describe the architecture and operational mechanism of the proposed platform.

A. ISSUE FOR IMPROVING THE CONVENTIONAL CO-SIMULATION PLATFORM

As mentioned in the previous section, co-simulation platforms have limitations when it comes to verifying and evaluating the functionalities and performances of the UAV-based applications because they have been constructed based on the network simulation tools and have focused on wireless network simulations. Through the analysis and comparison of conventional co-simulation platforms, their characteristics can be compiled into Table 1.

As shown in Table 1, simple flights of UAV are supported without visualization in AeroLoop and FlyNetSim. In addition, the packets are delivered only via down link, one-way transmission from UAV to GCS, in the AeroLoop, FlyNetSim, and CORNET. Even the packet transmission between UAV and GCS is not supported in the CUSCUS. In particular, a single data packet containing both the flight information and location data of the UAV is transmitted over a single channel. This packet is sequentially processed for UAV simulation and wireless network simulation on the conventional co-simulation platforms. Fig. 1 represents the synchronization scheme based on the single channel and sequential processing in the conventional platforms. Note that only the location data included in the packet is utilized for the wireless network simulation indicated by the dotted-box in Fig. 1, and the others are used for UAV simulation. As shown in Fig. 1, the packet to be transmitted to GCS for UAV simulation is determined in the packet management stage based on the time delays \( d_p \) and \( d_n \). In Fig. 1, \( d_p \) represents the processing time required to receive and interpret a packet and perform a wireless network simulation and \( d_n \) indicates the network propagation time obtained from the NS-3 tool. If \( d_p \) is smaller than \( d_n \), the packet transmission to the GCS is delayed to synchronize the UAV simulation with the wireless network simulation. Otherwise, the packet is discarded in the packet management stage. Packets discarded during synchronization can cause inaccurate simulation results. For these reasons, UAV simulation and wireless network simulation cannot be successfully performed on the conventional co-simulation platforms.

Based on the analysis and comparison mentioned above, the following issues should be addressed to improve conventional co-simulation platform:

- Effective visualization of UAV flights.
- GCS supporting mission planning and UAV management.
- Up-down link for packet transmission between UAV and GCS.
- Efficient synchronization scheme supporting co-simulation.
- Real-time operation of the simulation platform.

B. DESIGN OF THE PROPOSED CO-SIMULATION PLATFORM

In order to perform the co-simulation successfully, we propose an advanced co-simulation platform composed of an UAVSIM part, a WNETSIM part, and an ITG part. Fig. 2 illustrates the internal configuration and operation flow of the proposed co-simulation platform. It also shows how these parts relate to and integrate with each other.

1) UAVSIM PART

The UAVSIM part in Fig. 2 aims to verify the functionalities of the UAV-based application being developed and to visualize the flights of UAV under a virtual environment. To achieve this purpose, the UAVSIM part is constructed with a virtual environment module, a virtual UAV module, and a GCS module. The virtual UAV module is developed with PX4 to support various flight scenarios based on a dynamic flight
TABLE 1. Characteristics of the conventional co-simulation platforms.

| Co-simulation platform | Open-source | Type of UAV flight | Visualization | Packet transmission mode | Channel | Synchronization scheme |
|------------------------|-------------|--------------------|---------------|--------------------------|---------|------------------------|
| CUSCUS[8]              | FL-AIR NS-3 | Waypoints          | Yes           | Up-down link (UAV - UAV) | Single  | Sequential processing  |
| AeroLoop[9]            | Ardupilot NS-3 | Waypoints          | No            | Down link only (UAV - GCS) | Single  | -                      |
| FlyNetSim[10]          | Ardupilot NS-3 | Basic movements   | No            | Down link only (UAV - GCS) | Single  | Sequential processing  |
| CORNET[11]             | PX4 SITL Gazebo, NS-3 | Waypoints          | Yes           | Down link only (UAV - GCS) | Single  | Sequential processing  |

FIGURE 1. Synchronization scheme in the conventional co-simulation platforms.

model, and the virtual environment module is designed with Gazebo to visualize the flights of UAV under a specific virtual environment. In addition, this part has two data channels for transferring data and commands to the ITG part. One channel is used to send LO-packets carrying only the location data of UAV, which are used for wireless network simulation. The other channel is employed to transmit FC-packets containing the flight data of UAV and commands, which are delivered to and from the virtual UAV module and the GCS module via the ITG part. Compared to the conventional platforms, the GCS module in the proposed platform has been upgraded to better support mission planning and UAV management. In this module, commands to control the UAV are sent to the virtual UAV module through the ITG part, while the mission and flight of UAV are simulated by using the FC-packet transmitted from the ITG part.
2) WNETSIM PART
The goal of the WNETSIM part is to perform a wireless network simulation and send simulation results such as delay and throughput to the ITG part. As shown in Fig. 2, the WNETSIM part consists of a WNS module and a monitoring module. In the WNS module with the NS-3 tool, the virtual UAV is regarded as a node and its location is periodically updated based on the information of distance between UAV and GCS received from the ITG part. Then network simulation is carried out using the location of the UAV under a Wi-Fi wireless network environment. Since the simulation results depend on the wireless network conditions, it is reasonable to consider them as network parameters. For this reason, the simulation results are sent to the ITG part where the transmission of FC-packet is managed. This ultimately allows UAV simulation based on the state of wireless network at the UAVSIM part. The monitoring module provides real-time observation of the wireless network simulation by graphically displaying the simulation results.

3) ITG PART
The ITG part plays a key role in the proposed co-simulation platform. It is because this part links the UAVSIM part and the WNETSIM part and makes it possible to perform UAV simulation and wireless network simulation at the same time. The ITG part is composed of an Info_Ext module, a P_Mgt module, and a P_Sync module, as shown in Fig. 2. In the Info_Ext module, the LO-packets are periodically received from the virtual UAV module of the UAVSIM part, and the distance between UAV and GCS is obtained and sent to the WNETSIM part. In the P_Mgt module, which is constructed with UDP sockets and queues, the FC-packets received from the UAVSIM part are forwarded to the P_Sync module, and then the FC-packets are re-transmitted from the P_Sync module and sent to the GCS module or the virtual UAV module in the UAVSIM part. Since wireless network environment is not taken into account in the FC-packet transmission between UAV and GCS in the UAV simulation, it is desirable to reflect the wireless network state while transmitting the FC-packets to perform the UAV simulation under a virtual wireless network environment. For this purpose, the re-transmission of FC-packets in the P_Sync module is determined by networks parameters, which are periodically updated with the simulation results obtained from the WNETSIM part.

4) SYNCHRONIZATION SCHEME BASED ON PARALLEL PROCESSING
In general, synchronization scheme has significant impacts on the performance of the co-simulation platform. However, conventional co-simulation platforms have inefficient synchronization schemes. Because data packets are transmitted through a single channel and are sequentially processed in the conventional synchronization schemes, the NS-3-based network simulation stage is always executed before the packet management stage as shown in Fig. 1. This makes it impossible to precisely synchronize the UAV simulation with the
wireless network simulation and restricts the performance of co-simulation platforms.

In order to overcome the limitations of the conventional scheme, the NS-3-based network simulation stage and packet management stage need to be concurrently driven. A simple way to satisfy this requirement is to introduce two data channels so that the LO-packets and FC-packets are separately transmitted to the NS-3-based network simulation stage and the packet management stage, respectively. Based on this idea, an efficient synchronization scheme based on parallel processing is proposed in this paper. Fig. 3 illustrates the packet flow of the proposed synchronization scheme. From Fig. 3, we can confirm that FC-packets and LO-packets are concurrently transmitted to the WNS module and the P_Sync module and these modules are simultaneously operated. As a result, the processing time delay \( d_p \) is reduced and the parallel processing of the packets prevents the FC-packets from being discarded. Therefore, the proposed scheme enables accurate synchronization of UAV simulation and wireless network simulation. In addition, the task of interpreting packets can be eliminated due to the independent packet transmission. This means that various structures of FC-packet can be used on the proposed platform.

### C. IMPLEMENTATION OF THE PROPOSED CO-SIMULATION PLATFORM

In this paper, the functions and modules described above are configured into a hierarchical structure so that the proposed co-simulation platform can be easily utilized and extended. Fig. 4 illustrates the software architecture of the proposed co-simulation platform based on the functions and internal configuration of the three parts. This software architecture was implemented with open-source software packages and the Ubuntu environment. As shown in Fig. 4, Gazebo 9.0, the 3DR Solo model of PX4 SITL, and QGroundControl were introduced to develop the virtual environment module, the virtual UAV module, and the GCS module, respectively. The WNS module was constructed using NS3.34 to support the Wi-Fi 802.11a/b/g/n/ac wireless network environment. In addition, MAVSDK [24] was adopted to transmit the LO-packet in the Info-Ext module. The other modules in this paper were implemented using C++ with QT5. In particular, the P_Sync module was implemented based on the multi-threads to support concurrent queuing of FC-packets in the P-Mgt modules. The graphical user interface (GUI) module in Fig. 4 consists of six panels. These panels visualize the UAV flight and the network parameters generated by the WNS module, monitor the status of NS-3 and PX4 SITL, interface with the GCS, and set the simulation parameters.

### IV. EXPERIMENTAL RESULTS

In order to verify the proposed co-simulation platform, we carried out some experiments based on an Intel Core i7-3770 3.40 GHz CPU, DDR3 12 GB RAM, NVIDIA GTX 1060, and Ubuntu 20.04 LTS. In these experiments, three test scenarios were applied to evaluate the performance, effectiveness, and extendibility of the proposed co-simulation platform.

#### A. CO-SIMULATION PERFORMANCE OF THE PROPOSED PLATFORM

In order to validate the functionalities of the proposed co-simulation platform, we set up the 1st test scenario in which a single UAV flies over a square-shaped flight path and the proposed platform simultaneously performs UAV simulation and wireless network simulation. In this experiment, adjacent waypoints were spaced 200 m apart and TxPower, default delay, and data rate were set to 40dbm, 0\( \mu \)s, and 10Mbps, respectively. In addition, two propagation loss models, Logdistance and Frils, were applied to the Wi-Fi
FIGURE 5. GUI displaying the intermediate results and status of the co-simulation based on Wi-Fi 802.11g model. (a) Friis model. (b) Logdistance model.

802.11g model provided by the NS-3 tool. Fig. 5 illustrates the GUI panels displaying the intermediate results and status of the co-simulation being performed on the proposed platform. The upper left and lower left panels show the UAV flying in the virtual environment and the flight path and flight progress monitored in the GCS, respectively. The upper right panel contains four graphs with network parameters such as distance, delay, throughput, and RxPower displayed in real-time according to the flight of UAV. The lower right panels in Fig. 5 show the operating status of the PX4 SITL and NS-3 tools, respectively. Through Fig. 5, it is apparent that the visualization of UAV flight as well as the wireless network simulation are executed at the same time. In addition, Fig. 5 demonstrates that the GCS panel of the GUI supports mission planning and UAV management, and that packet transmission between GCS and UAV is effectively performed on the proposed platform.

It is meaningful to investigate the wireless network simulation of the proposed platform in detail according to the Wi-Fi model. Fig. 6 shows the network parameters generated by co-simulation, in which the 1st test scenario of Fig. 5 is applied to the Wi-Fi 802.11g model and the Wi-Fi 802.11ac model. In Fig. 6, the Wi-Fi 802.11g model and the Wi-Fi 802.11ac model are represented by blue and red lines, respectively. Fig. 6 (a) shows the distance change between the GCS and the UAV where the two lines completely overlap as the flight path are the same. As the distance between the GCS and the UAV increases, Fig. 6 (b) shows how the packet transmission time increases for each Wi-Fi model. In addition, Fig. 6 (c) and (d) illustrate how the amount of data transmission per second and
Figure 6. Network parameters obtained from the co-simulations based on Logdistance model. (a) Distance. (b) Delay. (c) Throughput. (d) RxPower.

the signal strength reaching the UAV decrease for each Wi-Fi model, respectively. Note that these graphs are actually drawn in the network parameter panel of the GUI. According to Fig. 5 and Fig. 6, it is obvious that the proposed co-simulation platform can support various wireless network environments and perform co-simulation successfully.

B. EFFECTIVENESS OF SYNCHRONIZATION SCHEME

In order to ensure the accuracy of simulation results shown in the previous section, it is necessary to verify whether the wireless network simulation on the proposed platform is correctly operated or not. This was checked by comparing the results of the wireless network simulation performed on the proposed platform with those of the standalone NS-3 tool. Fig. 7 shows the delay and throughput obtained from the proposed platform and the standalone NS-3 tool for the 1st test scenario and simulation conditions described in the previous section. In Fig. 7, the red and blue lines indicate the delay and the throughput, respectively. The wireless network simulation on the standalone NS-3 tool was performed based on the log data recorded by the virtual UAV during the co-simulation. Fig. 7 illustrates that the delay and throughput of the proposed platform are the same as those of the standalone N3-tool. This clearly shows that the wireless network simulation works correctly on the proposed platform.

Since co-simulation performance typically depends on the synchronization scheme, it is desirable to observe the effect of parallel processing applied on the synchronization scheme of the proposed co-simulation platform. Based on the 2nd test scenario where the UAV maintains a constant position at a fixed distance from the GCS, the processing time delay $d_p$ of the proposed platform was compared to that of FlyNetSim. Fig. 8 shows the processing time delays measured in the proposed platform and FlyNetSim for a sequence of packets transmitted from the UAV to GCS. The average processing time delays of the proposed platform and FlyNetSim are 0.186ms and 2.617ms, respectively. It means that the processing time was saved by approximately 92% due to the synchronization scheme of the proposed platform. This clearly shows that the two-channel approach with parallel processing of the proposed platform is more effective for UAV simulation than a single-channel approach.

C. EXTENDIBILITY OF THE PROPOSED PLATFORM

Since extendibility is one of the key features to be considered in developing a software platform, it is needed to
examine whether the proposed co-simulation platform can be extendible or not. For this purpose, Gazebo adopted to supply a virtual environment in the proposed platform was replaced with AirSim. AirSim provides a flight simulation environment based on Unreal engine and supports various missions of UAV by connecting with PX4. Hence, it is useful for UAV-related simulations and researches [14], [17], [19]. In this experiment, co-simulation was performed according to the 1st test scenario and simulation conditions of Fig. 5 (a). Comparing Fig. 9 with Fig. 5 (a), it can be easily seen that the simulation results with AirSim are almost identical to those with Gazebo. This clearly shows that various virtual environments can be easily introduced and supported on the proposed platform.

In order to further examine the extendibility of the proposed platform, the proposed platform was applied to the development of UAV-related control system. In this experiment, QgroundControl, which was the GCS in the proposed platform, was replaced with a customized GCS implemented with MATLAB, and a co-simulation was carried out to track a simple guidance command of an UAV. For this simulation, we set up the 3rd test scenario in which the UAV makes a turning flight with a radius of 50m by receiving a velocity command from the customized GCS. Let \( \vec{p} \) be a two-dimensional location vector of UAV transmitted to the GCS. Then, the velocity command generated by the GCS is formulated as follows:

\[
\vec{v}(x, y) = \vec{p} \times \vec{\omega} + (r - \|\vec{p}\|) \frac{\vec{p}}{\|\vec{p}\|},
\]

where \( r \) and \( \vec{\omega} \) denote the desired turning radius and the angular velocity vector, respectively. Note that \( \vec{p} \) and \( \vec{v}(x, y) \) are transmitted between UAV and GCS during the turning flight. Fig. 10 represents co-simulation results on the proposed platform for the 3rd test scenario and the data rate of 4Mbps. From Fig. 10, we can identify that the flight path of the
UAV is distorted compared to the target reference path. This means that the wireless network environment affected the packet transmission between GCS and UAV causing packet delay during the simulation. Thus, the proposed platform is helpful for predicting how much drift of UAV can occur in a wireless network environment and for designing an efficient flight controller. According to these experimental results, it is obvious that the proposed platform can work with other software modules to perform various types of functions and missions.

V. CONCLUSION

It is evident that UAV simulation platform is essential for developing UAV-based applications. However, since the wireless network environment has not been considered in conventional UAV simulation platforms, they have limits when it comes to developing reliable and practical UAV-based application. For this reason, co-simulation platforms based on a synchronization scheme have been suggested as a solution to this problem. However, the conventional co-simulation platforms still are unsuitable for performing simultaneous UAV simulation and wireless network simulation due to critical shortcomings.

In this paper, we propose an advanced co-simulation platform capable of verifying and evaluating UAV-based applications under a virtual wireless network environment. Based on the analysis of the conventional co-simulation platforms, we have derived a list of improvements for the conventional co-simulation platform and developed the proposed platform consisting of UAVSIM, WNETSIM, and ITG parts. In the proposed platform, a two-channel approach was applied to support various types of missions and associated UAV flights and to parallelize synchronization. From the experimental results for three test scenarios, it has been shown that the
proposed platform efficiently provides the functionalities required for co-simulation based on efficient interoperation between the three parts. In addition, the processing time delay in the wireless network simulation was reduced by approximately 92% due to the synchronization scheme based on the two-channel approach. This allows the co-simulation to be performed more precisely on the proposed platform. Moreover, the proposed platform can support various simulations because it is extendible due to its modularized software architecture.

REFERENCES

[1] G. Cai, J. Dias, and L. Seneviratne, “A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends,” Unmanned Syst., vol. 2, no. 2, pp. 175–199, Apr. 2014.

[2] C.-J. Chen, Y.-Y. Huang, Y.-S. Li, Y.-C. Chen, C.-Y. Chang, and Y.-M. Huang, “Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying,” IEEE Access, vol. 9, pp. 21986–21997, 2021.

[3] J. Li, Y. He, X. Zhang, and Q. Wu, “Simultaneous localization of multiple unknown emitters based on UAV monitoring big data,” IEEE Trans. Ind. Informat., vol. 17, no. 9, pp. 6303–6313, Sep. 2021.

[4] J. Xu, K. Ota, and M. Dong, “Big data on the fly: UAV-mounted mobile edge computing for disaster management,” IEEE Trans. Netw. Sci. Eng., vol. 7, no. 4, pp. 2620–2630, Oct. 2020.

[5] M. Bacco, P. Barsocchi, P. Cassara, D. Germanese, A. Gotta, G. R. Leone, D. Moroni, M. A. Pascali, and M. Tampucci, “Monitoring ancient buildings: Real deployment of an IoT system enhanced by UAVs and virtual reality,” IEEE Access, vol. 8, pp. 50131–50148, 2020.

[6] A. S. Vempati, H. Khurana, S. Flueckiger, R. Siegwart, and P. Beardsley, “A virtual reality interface for an autonomous spray painting UAV,” IEEE Robot. Autom. Lett., vol. 4, no. 3, pp. 2870–2877, Jul. 2019.

[7] Y. Wang, M. Koutsoubelias, N. Grigoropoulos, and S. Lalis, “A modular simulation environment for multiple UAVs with virtual WiFi and sensing capability,” Proc. 14th IEEE Annu. Consumer Commun. Comput. Netw. Conf. (CCNC), Las Vegas, NV, USA, Jan. 2017, pp. 287–292.

[8] M. Koutsoubelias, N. Grigoropoulos, and S. Lalis, “A modular simulation environment for multiple UAVs with virtual WiFi and sensing capability,” Proc. IEEE Sensors Appl. Symp. (SAS), Seoul, South Korea, Mar. 2018, pp. 1–6.

[9] S. Baidya, Z. Shaikh, and M. Levorato, “FlyNetSim: An open source synchronized UAV network simulator based on ns-3 and ardublock,” in Proc. 21st ACM Int. Conf. Modeling, Anal. Simul. Wireless Mobile Syst., Montreal, QC, Canada, Oct. 2018, pp. 37–45.

[10] K. D. Nguyen and T.-T. Nguyen, “Vision-based software-in-the-loop simulation for unmanned aerial vehicles using gazebo and PX4 open source,” in Proc. Int. Conf. Syst. Sci. Eng. (ICCSSE), Dong Hoi, Vietnam, Jul. 2019, pp. 429–432.

[11] R. Garcia and L. Barnes, “Multi-UAV simulator utilizing X-plane,” J. Intell. Robotic Syst., vol. 57, nos. 1–4, pp. 393–406, Oct. 2010.

[12] C. Ma, Y. Zhou, and Z. Li, “A new simulation environment based on airsim, ROS, and PX4 for quadcopter aircrafts,” in Proc. 6th Int. Conf. Control, Autom. Robot. (ICCAR), Singapore, Apr. 2020, pp. 486–490.

[13] V. Sadhu, S. Zonouz, and D. Pompili, “On-board Deep-learning-based unmanned aerial vehicle fault cause detection and identification,” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), Paris, France, May 2020, pp. 5255–5261.

[14] W. J. Cho received the B.S. and M.S. degrees in computer science and aerospace engineering from Gyeongsang National University, Republic of Korea, in 2017 and 2019, respectively. From 2019 to 2021, he was at the Engineering Research Institute, Gyeongsang National University. His research interests include image processing, simulation software, and augmented reality.

[15] Seongheon Kim received the B.Eng. degree in aerospace engineering from Gyeongsang National University, Republic of Korea, in 2021. His current research interest includes building a test platform for manned-unmanned teaming (MUM-T) of air vehicles.

[16] Yoonsoo Kim (Member, IEEE) received the B.Eng. degree in aerospace engineering from Inha University, Republic of Korea, in 1999, the M.Sc. degree in aerospace engineering from the University of Minnesota, USA, in 2001, and the Ph.D. degree in aerospace engineering from the University of Washington, USA, in 2004. He held a postdoctoral and a senior lecture positions at the University of Leicester, U.K., from 2004 to 2007, and the University of Stellenbosch, Stellenbosch, South Africa, from 2007 to 2011, respectively. Since 2011, he has been a Faculty Member with the Department of Aerospace and Software Engineering, Gyeongsang National University, Republic of Korea, where he is currently a Professor. His main research interest includes distributed control of networked dynamical systems.

[17] Yong Ho Moon received the B.Eng., M.S., and Ph.D. degrees in electronics engineering from Pusan National University, Republic of Korea, in 1992, 1994, and 1998, respectively. He was at the Corporate Research and Development Center, Samsung Electronics, South Korea, from 1998 to 2001. He has been on the faculty with Gyeongsang National University, since 2007, where he is currently a Professor with the Department of Aerospace and Software Engineering. His research interests include image processing, A/V, SoC, embedded software, and avionics.