

Analysis, Development, and Simulation of a Mission System for MUM-T Operational Scenarios

M. Cicalini^{*, $\diamond$, $\dagger$} , M. S. Lukasiewicz $\diamond$, D. Frisini $\dagger$, P. Masarati $\diamond$

$\diamond$ Politecnico di Milano - Dipartimento di Scienze e Tecnologie Aerospaziali, via La Masa 34, 20156 Milano, Italy

$\dagger$ TXT E-TECH, via Milano 150, 20093 Cologno Monzese (MI), Italy

* e-mail: marco.cicalini@txtgroup.com

Abstract

This paper describes the modelling and the simulation of a mission system for manned-unmanned teaming operational scenarios, with a specific focus on rotorcraft applications. To accomplish the task, a high-level architecture to enable teaming is defined and prototypes of a Pilot Onboard Station and a Ground Control Station have been realized. Both prototypes are coded with open-source software like QGroundControl, and are meant to cover all the flight phases of planning, monitoring, and decision support. They were integrated in a helicopter flight simulator specifically designed for teaming simulation, in which the aeromechanical and feedback implementation of the main vehicle are based on an MBDyn/Simulink architecture and the uncrewed vehicles are simulated using AirSim with PX4-Autopilot as flight controller. The involved graphics engine is Unreal Engine 5, chosen due to its high graphical realism. With this configuration, a final test campaign was performed, from hardware and software setup to the description of results. These are encouraging in terms of required workload and intuitiveness, opening the way to more mature implementations.

1 INTRODUCTION

Manned-unmanned teaming (MUM-T), which could perhaps be better worded as Crewed-Uncrewed Teaming, is a rapidly growing theme in the aerospace industry. It consists of the cooperation of manned and unmanned flying assets sharing a common mission objective, resulting in a force multiplier and increased capability for the crewed aircraft. Developing systems for this technology requires novel solutions in different fields, including system architecture, human-machine interface (HMI), and new criteria for drone management and simulation. This work proposes the analysis, development, and simulation of a mission system prototype for MUM-T operational scenarios.

2 STATE OF THE ART

Manned-unmanned teaming involves disciplines like aeronautics, telecommunications, electronics, air navigation, human psychology, and user experience. The current literature proposes different Levels of Interoperability (LOI) [1] between manned and unmanned vehicles, with an increasing level of complexity and with the final goal identified as the complete interoperability with a swarm of Unmanned Air Vehicles (UAVs). The achievement of higher and higher LOI will bring to the development of better systems for MUM-T.

Also, along with the realization of secure, efficient and intuitive mission systems, the development of reliable simulation platforms is candidate to become a driving factor for the development of this technology.

2.1 Mission Systems

The current literature focuses on the development of both onboard platforms and ground facilities.

According to [2] the onboard system should grant the flight phases of planning, monitoring, and decision support, while the tasks of the ground station can be summarised by some of the available literature [2, 5] in flight data backup, tactical support, and direct control of the UAVs when needed.

Ref. 3 shows a possible onboard HMI based on touch screen devices, which enables the human operator to insert, modify, and remove tasks to be performed by each UAV. The visual representation should be on the map, and allow mission management. Meanwhile, the experimental campaign described in [4] candidates the use of the touch screen as a viable means to enable aerial vehicle teaming. The same paper advises assisted decision-making in the system and rational workload distribution to be sufficient means to reduce the down effects of this specific input system. The implementation of task-based logic to control the UAVs seems to be a common ground of different papers [3, 5].

Substantial research interest is also put in the development of release and capture systems [1, 2] and some successful tests of this type have already been performed¹.

¹ Airbus, "Remote carrier demonstrator released and operated from flying A400M for the first time," Press release, Madrid, 12/12/2022, <https://www.airbus.com/en/newsroom/press-releases/2022-12-remote-carrier-demonstrator-released-and-operated-from-flying-a400m>

2.2 Simulation Platforms

Regarding simulation platforms, different prototypes to replicate MUM-T have been described and are now available. Ref. 3 proposes a solution that enables the reproduction of a complete military operation, including those elements referred to Crewed-Uncrewed Teaming. Meanwhile, [6] presents a full UH-60 MUMT-T simulator. This simulator implements the control onboard using a touch screen device and proposes the use of novel tools like the graphics engine Unreal Engine and the flight controller PX4-Autopilot.

3 SYSTEM FUNCTIONALITIES

What is wanted is the realization of a high-level prototype of a mission system for aerial cooperation between a helicopter and one or more UAVs, integrated in a reliable simulation environment. The development of such a platform may serve to replicate use cases and simulate concepts of operations, thus collecting feedback to improve the current state of the art.

3.1 Mission System

The mission system shall consist of a Pilot Onboard Station (POS) and a Ground Station (GS), each with specific functionalities and contributing redundancy to the final architecture. Both instruments shall maintain communication with each other and the controlled drone swarm, ensuring situational awareness and the possibility of intervention to human operators.

3.1.1 Onboard Station

In summary, the Pilot Onboard Station should grant the following flight phases, as proposed in [2]:

- **PLANNING:** it shall ensure the mission definition that each UAV has to perform in terms of tasks, altitude, cruise speed, and waypoints to be followed.

The current literature advises the application of task-based functionalities to reduce the pilots' workload [3] [5]. This is necessary since the flight crew already has numerous tasks to perform and the direct control of an external UAV (or even a swarm of flying objects) would be beyond human capabilities;

- **MONITORING:** it shall ensure the possibility to monitor the state of the mission. This means visualizing the drone position and heading on map [3], reading the data coming from sensors, being aware of flight altitude and speed, seeing the video streaming of UAVs' cameras, and advising the pilot in case of mission advancements or issues. Each of these pieces of information shall be presented to the crew most effectively and improve the pilot's situational awareness;

- **DECISION SUPPORT:** it shall ensure the possibility for the flying crew to make decisions along the mission development.

Some of the features to grant are the modification of altitude and cruise speed, the go-home functionality, the possibility to change the mission development in a general sense, and the possibility to activate and deactivate some of the UAVs' payload.

3.1.2 Ground Station

A Ground Station is required to permit efficient and safe use of a teaming technology, as highlighted in many available publications [2, 5]. It shall grant the following functionalities:

- **FLIGHT DATA BACKUP:** it shall perform the flight data backup of the UAVs, to have a record of the main sensors and performance parameters of the flight;
- **TACTICAL SUPPORT:** it shall collect and share information with both the manned and the unmanned agents. The exchanged data may regard the weather information of the drones' position, obstacles, other threats to the safety of flight and any other relevant information with respect to the operations;
- **SCENARIO SUPPORT:** the presence of a ground station provides a much higher computational capacity than that available onboard the aircraft. The ground station shall perform complex computations for flight support like path planning and trajectory assessment [7] or perform risk assessments for both the main helicopter and the uncrewed aircraft;
- **CONTROL OF THE UAVs:** the Ground Station should also intervene to directly control the unmanned aerial vehicles if needed. This may be necessary for specific mission profiles [3] or for safety and security reasons.

3.2 Simulation Platform

The simulation of MUM-T demands requirements that are common to any flight simulation and others specific to this particular application, raising the level of complexity.

For a start, it is necessary to accurately simulate the Flight Dynamic Model (FDM) of the main vehicle in real-time and with minimum latency. Also, MUM-T simulation has to take into account the realization of a reliable real-time FDM for each of the simulated unmanned vehicles. These models have to interact both with the environment and with the main vehicle of the simulation, as well as with any other moving object that is taken into account. Both unmanned vehicles and the aircraft need to receive inputs from the outside, i.e. the simulation platform shall grant the non-interference in command chains of the various involved actors while simultaneously receiving inputs.

Another crucial aspect is the simulation of the natural environment: Earth and the Atmosphere should be defined consistently and as close to reality as possible.

4 METHODOLOGY AND ARCHITECTURE

The described problem was treated with Systems Engineering methodology [18]. In particular, the developed design was driven by the functionalities described in Section 3, and the test campaign of Section 6 served as Verification & Validation (V&V) step.

4.1 Simulation Platform

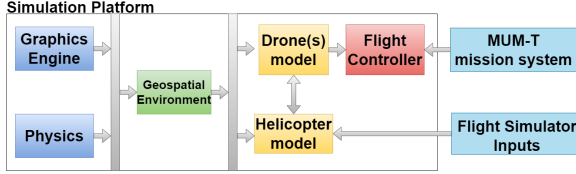


Figure 1: Architectural scheme of the simulation platform

4.1.1 Graphics Engine

The Graphics Engine used in the simulation is Unreal Engine 5. It was chosen among other available tools given its graphical realism, adaptability, and possibility to visualize complex environments. While it is a commercial product, it is licensed under a source-available license. The tool is written in C++ with multiple interfaces that need to be extended by user code. C++ is a low-level language defined in an open standard that allows almost full control of the machine's resources and memory with support for both object-oriented and procedural programming. Notably, Unreal Engine is also gaining interest in the aerospace community for simulation platforms based on this technology².

4.1.2 Geospatial Environment

Different tools were evaluated and eventually Cesium³ For Unreal was chosen. It was adopted because of its open-source Apache license while offering the possibility to handle accurate Earth models and high graphical realism. A high-resolution dataset was used, based on the World Geodetic System 1984 (WGS84), an Earth-centered and Earth-fixed coordinate system that also offers Earth Gravitational Model (EGM) and World Magnetic Model (WMM).

4.1.3 Drone Model

The drone is modelled using the Unreal Engine plugin AirSim [8]. It is developed in C++ and Python and allows high-fidelity simulations for multirotor drones, like quadcopters

and hexacopters. The simulation of multiple instances is allowed, thus permitting drone swarm simulations.

Specifically, the drone is modelled as a rigid body with a specified number of actuators generating forces and torques. The state of the drone is expressed in terms of position, orientation, linear velocity, linear acceleration, angular velocity, and angular acceleration. The next state is computed by knowing the acting forces and torques. The state is updated with a frequency of 1000 Hz, enabling real-time simulation.

Table 1: Coefficients defined in the vehicle model

UAV parameters	Symbol	Unit
Mass	M	[kg]
Inertia matrix	I	[kg m ²]
Vehicle cross section	A	[m ²]
Maximum angular velocity	ω_M	[rad/s]
Propeller Diameter	D	[m]
Number of Propellers	n	[n.d.]
Linear Drag coefficient	D_l	[n.d.]
Angular Drag coefficient	D_a	[n.d.]
Coefficients of friction	$f d_i$	[n.d.]
Thrust Coefficient	C_T	[n.d.]
Power Coefficient	C_P	[n.d.]

AirSim also offers the possibility to simulate a broad variety of sensors, including a barometer, IMU, GPS, magnetometer, distance sensor, LIDAR, and multiple cameras.

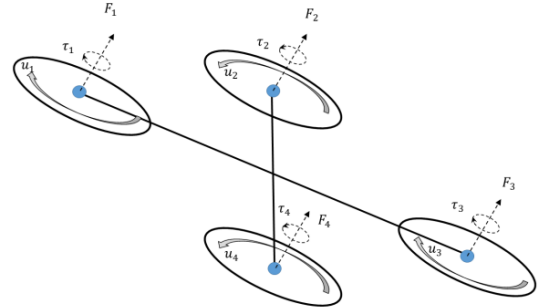


Figure 2: Quadcopter scheme [8]

4.1.4 Flight Controller

The flight controller is crucial to ensure the controllability of the UAVs. Its main task is to take the desired state and the sensors' data as inputs to compute the current state of the UAV and consequently output the actuator control signals to achieve the desired state.

In practical terms, the flight controller is usually implemented as firmware on the dedicated hardware of the Unmanned Vehicle. In this case, it was necessary to work Software in The Loop (SITL). Consequently, it was decided to use PX4-Autopilot⁴, an adaptable and modular drone flight controller compatible with SITL applications.

²Epic Games - Antoinette Project: tools to create the next generation of flight simulators, <https://www.unrealengine.com/en-US/blog/antoinette-project-tools-to-create-the-next-generation-of-flight-simulators>

³Cesium - The platform for 3D geospatial, <https://cesium.com/>

⁴PX4 - Open Source Autopilot For Drone Developers, <https://px4.io/>

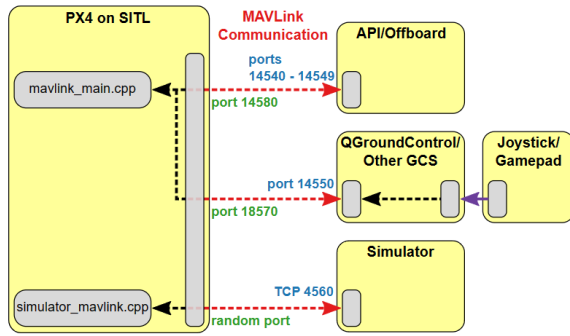


Figure 3: PX4-Autopilot in SITL

The communication with the simulation environment is allowed using Transmission Control Protocol (TCP) on port 4560 (4560+i in case of multi-vehicle simulations) and the communication with the POS by means of MAVLink packets on User Datagram Protocol (UDP) (port 14550+i by default).⁵



Figure 4: Outcome of drones simulation

4.1.5 Physical Environment Modelling

The implemented tools allow the simulation of the most relevant physical phenomena that characterize the simulation environment, like the gravity, the magnetic fields, and the Air Model. The latter, in particular, is modelled according to the International Standard Atmosphere (ISA).

4.1.6 Helicopter Model

The aeromechanical model and the simulation of the control chain of the main vehicle is realized using an MBDyn/Simulink co-simulated platform with data representative of a lightweight rotorcraft. MBDyn⁶ [9] is a free general-purpose multibody solver, which is at the core of the flight dynamics module of the FRAME Sim architecture at Politecnico di Milano [10–12].

The main vehicle can be modelled as three interacting layers:

- The aeromechanical characterization of the helicopter is defined using an MBDyn rigid model, that defines

the most salient parameters of the rotorcraft and that models the complete rotor assembly in terms of control chain and allowed degrees of freedom. The helicopter state is updated 110 times for each rotor's complete revolution (700 Hz) and the differential model is solved using the Newton-Rapson method updating the Hessian Matrix every 2 iterations. The simulation is performed in real time, enabling human operators to pilot the simulated aircraft;

- The outputs of the MBDyn model (computation of forces, torques, and update of the kinetic state of the aircraft) are then used as inputs for the Simulink component. This model collects the attitude and position setpoints imposed by the pilot and allows to get the appropriate aeromechanical response in strong interaction with the underlying MBDyn outcomes; also, an autopilot system that allows to reach desired trimmed conditions of flight is implemented using dedicated feedback loops. The MBDyn and Simulink models can communicate in real-time using a dedicated port of communication;
- The graphical representation is granted using Unreal Engine 5. Specifically, the resulting attitude and position of the helicopter coming from the MBDyn/Simulink architecture is passed to the graphics framework using the Python library Lidia, using the Unreal Engine plugin UDP-Unreal and sending the states as SMOL messages. Two slightly different versions of the project were realized. The former is meant for desktop applications, the latter shows the external environment as filtered for representations on curved surfaces. This second solution was specifically developed for the AT-TILA Frame-SIM simulator of Politecnico di Milano and the discussed test campaign of Section 6.

The outcome is a real-time flyable helicopter model, that can be used for MUM-T simulation if paired with the Mission System described in Section 4.2.



Figure 5: Simulation between the helicopter and a swarm of 4 quadcopters

⁵PX4 Guide - Simulation, <https://docs.px4.io/main/en/simulation/>

⁶MBDyn - Multi-Body Dynamics, <https://mbdyn.org/>

4.2 Mission System

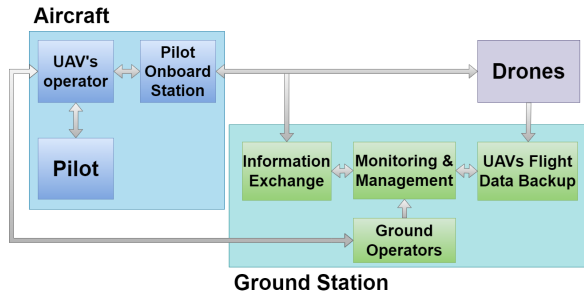


Figure 6: High-level interaction scheme of MUM-T operations

4.2.1 Pilot Onboard Station

Its prototype is based on a customized version of QGroundControl (QGC). This software is cross-platform, open-source, and written in C++ with Graphical User Interface (GUI) based on the Qt libraries. It allows to handle up to 8 drones and it is compatible with SITL implementations. Also, the source code of QGC natively supports the use of touchscreen devices to interact with the application, making it an appropriate prototype base for the designed purpose [4]. QGC was customized to suit MUM-T's needs and integrated with additional pieces of code when needed.

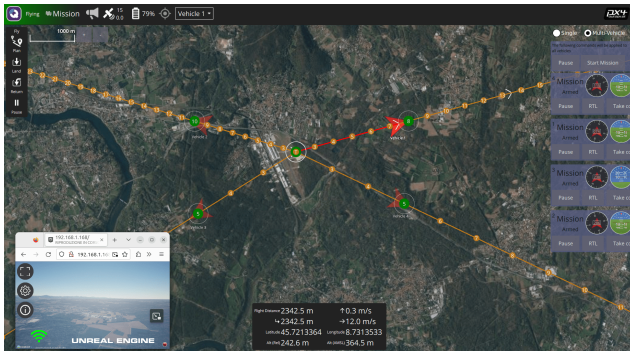


Figure 7: Onboard Station software

The POS covers the flight phases as follows:

- **Planning** QGC already in the not-customized version allows a complete mission definition. It is possible to define take-off, waypoints of the mission, altitude, cruise speed, fences of forbidden areas, and alternative landing points when performing a Return to Launch (RTL).

A rudimentary storage system was added to this available base and it was introduced the assisted planning modes described in Section 4.3.

- **Monitoring** QGroundControl allows us to visualize the drone position and heading on the map. Each drone has a specific sub-menu where it is possible to visualize its mission development and the data coming from

its sensors. The drone's map position is centred by default. A swarm display is also available.

UAVs' camera streaming to both POS and GCS is granted in the simulation environment by implementing the Unreal Engine plugin Pixel Streaming, which allows streaming projects to other devices in a Local Area Network (LAN).

- **Decision Support** QGC allows complete mission control. It is permitted the modification of altitude and cruise speed, the RTL functionality and to pause the drone or make it land when required. No form of customization was needed to implement all the requirements for the Decision Support phase.

4.2.2 Ground Control Station

The prototype of the GCS consists of different interacting software modules, mostly written in C++.

Flight data backup is granted by collecting the UAV's sensor information, with a GUI entitled to show the most relevant results with a defined frequency. The dedicated GUI is realized through Qt libraries. The simulated UAVs' control from the ground is ensured using the PX4 terminal, which allows full management of the swarm. Lastly, the possibility to visualize UAV cameras streaming is granted using the Unreal Engine plugin Pixel Streaming.

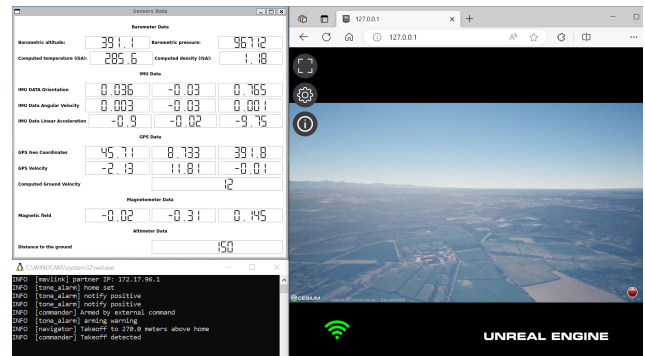


Figure 8: Frame of use of the GCS

4.3 Mission Planning

Task-based logic in drone management is seen as a key milestone in the development of MUM-T technologies. This is particularly true in the context of mission planning. Given the need to speed up the operations and to automatize as much as possible the process, two different planning modes were implemented in the developed mission system. The proposed algorithms try to find a compromise in terms of computational rapidity and performance and are seen as a starting point for future improvements.

4.3.1 Sequential Path Planning

The first available task-based planning mode is an assisted sequential generator. The algorithm outputs a single drone complete mission profile in the form of a .plan file. It acquires the home position and the pilot's inputs in terms of wanted cruise speed, altitude and sequence of forced nodes to be visited. The mission sequence of sub-waypoints is defined by generating straight lines in the absence of obstacles or points requiring an excessive vertical speed to be reached. The algorithm can interrogate an elevation dataset and when a terrain obstacle is detected, a sequence of random points inside a maximum radius is generated and a new node of the path is added in the position of minimum terrain elevation. The complete path plan, obtained by connecting the generated sub-waypoints, drives the UAVs to their destination.

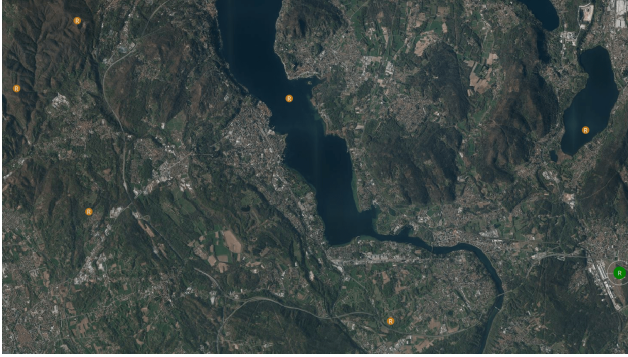


Figure 9: Sequential example - Pilot node inputs

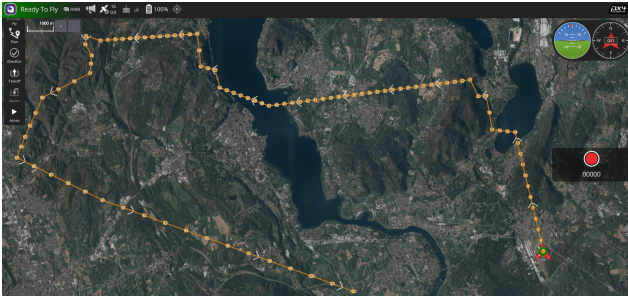


Figure 10: Sequential example - Generated algorithm outputs

The used dataset in Section 6 is the EU-DEM25⁷ but other sources can be used accordingly to the geographical region of interest and required resolution. The result is a mission path that can also evolve in height when specified by the UAV operator, adapting to the terrain altitude, as shown in Figure 11.

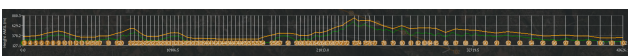


Figure 11: Sequential example - Generated Height profile

⁷EU-DEM, <https://www.opentopodata.org/datasets/eudem/>

4.3.2 Swarm Mission Path Planning

The second planning mode is meant for swarm coordinated mission profiles in which the same task has to be repeated in m waypoints by the n drones of the unmanned team.

A problem like this can be treated as a Vehicle Routing Problem with Drones (VRPD), a topic that can already count on a large literature [15–17]. It is NP-Hard and usually solved using non-optimal heuristic solutions. The currently implemented algorithm is a generalization of the Nearest Neighbour Algorithm (NNH) applied to the resolution of the Travel Salesman Problem (TSP). It enumerates the available drones and computes the distance matrix of the given waypoints, thus distributing iteration by iteration the nearest non-allocated waypoint to the turning drone. In this way, a close-to-parity distribution is obtained in terms of total distance and time to complete the assigned mission for each UAV. Once a drone is assigned a specific sequence to visit, its mission path is generated with the sequential path planning described in Section 4.3.1.

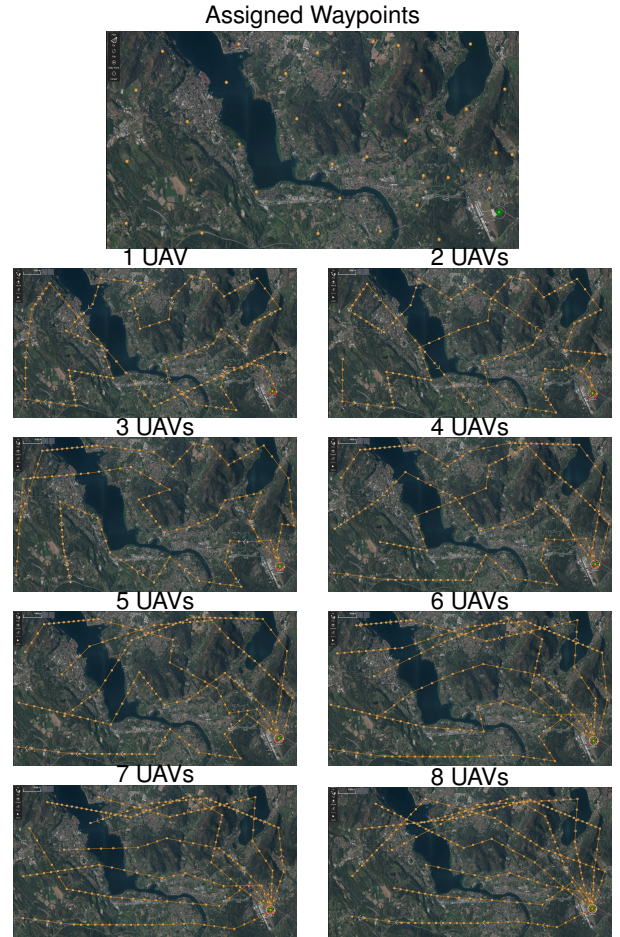


Table 2: Developed output profiles varying the number of UAVs

An optimization in the number of involved drones can be found for each analyzed parameter, like total mission time, distance and range/endurance autonomy of the single aircraft. Given the computed results, the operator can choose the mission profile that better suits the mission goals.

4.4 Global Interaction

The helicopter's and the UAVs' simulations are managed independently and put in communication so to allow reciprocal visual and physical representation. The choice to manage the projects separately permits wider scalability for the platform: the changes on the one side cannot affect the other component of the work.

The two simulations exchange data via UDP protocol, sending each other UDP packets at a defined frequency. The state of each aircraft is defined in terms of X , Y , and Z position and roll, pitch, and yaw angles.

We can interact with this simulation platform by means of the helicopter command chain and by means of the MUM-T mission system, respectively to control the main vehicle and the drone swarm. With the defined mission system, we can plan, monitor, and modify the UAVs' mission development.

An overall scheme of the proposed architecture is represented in Figure 12.

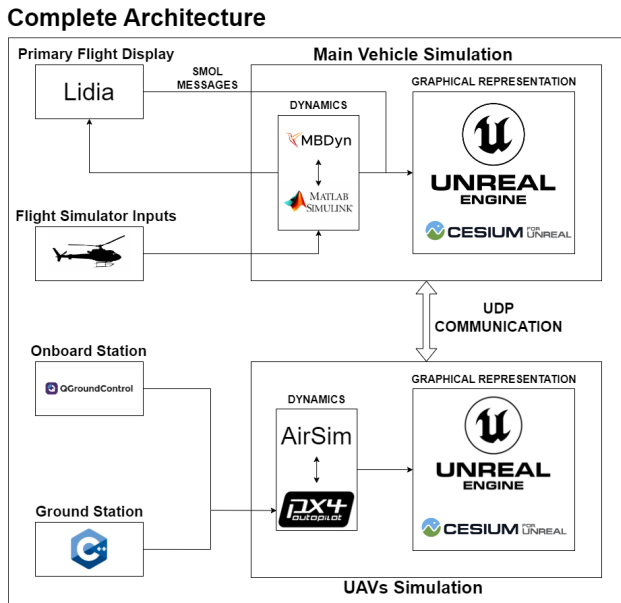


Figure 12: Scheme of the overall architecture

5 IMPLEMENTATION

The hardware & software integration was performed on the ATTILA Frame-SIM simulator of Politecnico di Milano. The chosen hardware & software setup is shown in Figure 13.

ATTILA Frame-SIM simulator provides a replication of the main control inceptors of a lightweight helicopter, providing a significant field of view (220° azimuth, $-70^\circ/+20^\circ$ vertical) on a curved screen. The simulator offers mechanical controls and the cockpit environment is completed by the presence of two touch-screen displays, on which it is possible to visualize the onboard panels. The computational effort is shared on the various available workstations: Attila vr, Attila 01, and Attila 03, to distribute the various operations on an integrated architecture.

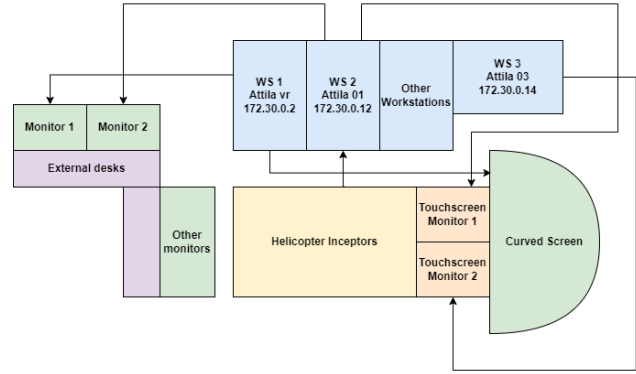


Figure 13: Integration scheme for MUM-T simulation setup in ATTILA Frame-SIM simulator

Given the hardware setup, the computational effort of the simulation was organized as follows:

- Attila vr:
 - run the H/C Unreal Engine simulation on the curved screen;
 - run the UAVs-only simulation in the background and when needed display it on Monitor 1;
 - run PX4-Autopilot and enable the flight controller software;
 - run and display on Monitor 1 the GCS software.
- Attila 01:
 - run the aeromechanical simulation of the main vehicle;
 - receive and interface with Simulink the flight set-points commanded by the pilot or the machine's auto-pilot;
 - visualize and project on Touchscreen monitor 2 Lidia's Primary Flight Display (PFD);
 - send the helicopter's state to the Unreal Engine simulation running on Attila vr. This is done using Lidia, which can exchange SMOL messages with other IP addresses inside the LAN;
 - display on Touchscreen Monitor 1 the streaming of the UAVs cameras, that are broadcasted in LAN;
- Attila 03:
 - run the POS software, which is represented and made available to the pilot on Touchscreen Monitor 2.

The various components of ATTILA Frame-SIM are connected in a Local Area Network (LAN), thus forming a distributed infrastructure for aerospace simulation.



Figure 14: Test campaign, 27/10/2023

5.1 Pilot Tracking

To gather a more objective measurement of additional load placed on the flight crew due to cooperation with unmanned vehicles, wearable eye tracking device [13] was integrated with the system, and fiducial markers [14] were displayed on touch screens. With that addition, the pilot's gaze can be located in the simulator cockpit. This in turn allows automatic and real-time classification of the pilot's attention focus between outside visual cues displayed on the spherical screen, flight instruments, and Pilot Onboard Station.

A frame from software used to replay the eye-tracking data is shown in Figure 15. Each screen surface was defined with four AprilTag markers in the corners. Initially, the markers were positioned outside the monitors printed on paper; however, this resulted in a poor detection rate because of the low light conditions required for good visibility of the image projected onto the spherical screen. Subsequently, the markers were overlaid on the displays used for the instruments and the POS.



Figure 15: View from eye tracking device camera with estimated gaze focus point and recognised surfaces overlaid on top

To ease experiments data analysis, a small program was written that translates in real-time the data from the eye-tracking device into custom messages of the MAVLink protocol used by PX4-Autopilot and other UAV autopilots. This

way, the pilot's gaze patterns can be saved together with other events in the same log file. This software is released under the open-source MIT License, together with detailed setup instructions for the experiment⁸.

6 TEST CAMPAIGN

The main test campaign was performed on 27/10/2023 and 15/11/2023, with a second phase dedicated to workload estimation that took place on 28/06/2024. Three mission scenarios were simulated: single drone SAR, multi-vehicle surveillance mission, and swarm corridor scan. These scenarios were chosen to test the widest possible range of situations and functionalities with a limited range of missions.

6.1 Single Drone SAR

The proposed mission wants to simulate a possible Search & Rescue application, in which a quadcopter drone teaming with a helicopter is in charge to reach an isolated mountainous area and scan an identified perimeter looking for the target. Once identified, the drone is dismissed and sent to a landing point which is different to the initial launch area. Projects like HEMS+ [2] propose interactions with ground mobile units and it is reasonable some kind of cooperation in the flight phases of landing and take-off. The presence of an unmanned vehicle in such a mission can significantly increase the searching capability of the flying crew, being able to explore the most dangerous areas meanwhile the helicopter looks for different zones of the environment. Also, the unmanned vehicle can serve as an external sensing platform or as a tool to assess the feasibility of trajectories and maneuvers of the main aircraft, as proposed in [7].

The mission proved the capacity of the architecture to handle a single drone all along a complex mission. The pilot could interact with the UAV and the touchscreen interface proved to be responsive and intuitive enough to permit easy and efficient use during the development of operations.



Figure 16: Touch Screen Monitors during SAR Mission - test campaign, 27/10/2023

The flight data backup capability was widely tested with this simulated scenario. Pieces of information like flight altitude (AMSL), ground speed, angular rates, and NED components of motion were recorded. Post-processing and

⁸Eye tracking - MARSH Documentation,
<https://marsh-sim.github.io/nodes/eye-tracking/>

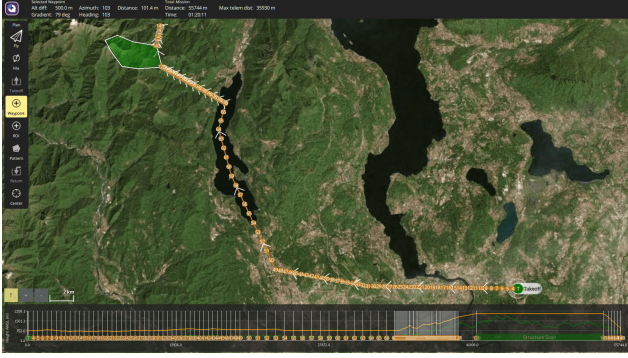


Figure 17: Complete UAV mission profile for the SAR mission

post-flight analysis is consequently permitted, as visible in the plots shown in Figure 18 and Figure 19.

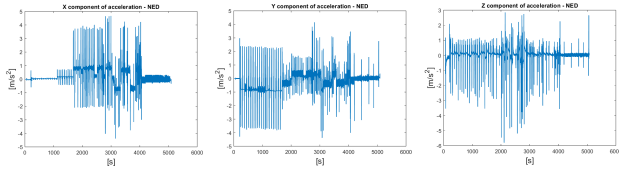


Figure 18: NED X, Y, and Z components of acceleration - test campaign 27/10/2023

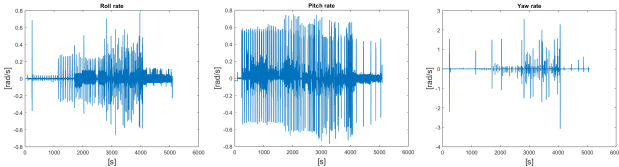


Figure 19: Roll, pitch, and yaw rates - test campaign 27/10/2023

6.2 Surveillance Swarm

The mission profile analyzed in this section is a possible application of MUM-T in surveillance missions. In the described profile, the drone operator commands an array of 20 geographical points to visit; these points are allocated to the various drones of the fleet and consequently surveil, acquiring data.

This simulation profile proved the mission system to manage successfully a drone swarm. Both one-to-one and one-to-many implemented features and controls resulted adequate for MUM-T purposes. Also, it was successfully tested the possibility to acquire information and data on areas of investigation by means of the uncrewed aircraft.

The commanded mission path plans were driven by the algorithms described in Section 4.3, with specific reference to that for Swarm Mission Path Planning in Section 4.3.2. The developed mission system remains intuitive and responsive also when managing multiple unmanned vehicles, both regarding the POS and the GCS.

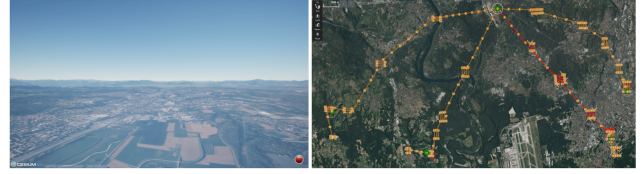


Figure 20: Surveillance Mission as seen from the POS - test campaign 27/10/2023

6.3 Swarm Corridor Scan

In this scenario, a helicopter and a drone swarm are committed in a ground troops transport. The UAVs shall acquire reconnaissance information about the helicopter routes and landing sites in real-time. The broader is their sensor footprint, the higher the safety for the manned rotor. The UAVs can be used to find threats nearby and eventually plan alternate drop zones; after unloading the troops, the helicopter comes back together with the unmanned fleet.

The corridor scan mission proved the mission system to be precise and responsive enough to cover a specific corridor of space surveyed by the drone swarm. The drones advance together maintaining an adequate relative distance and the implemented one-to-many monitoring and control tools resulted in always granting a full picture of operations. Based on the scanned corridor, a path for the helicopter was defined and the pilot achieved and followed it successfully.

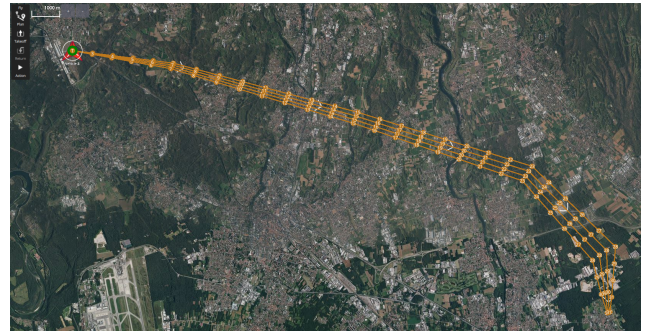


Figure 21: Complete mission profile for the swarm corridor scan - test campaign, 15/11/2023

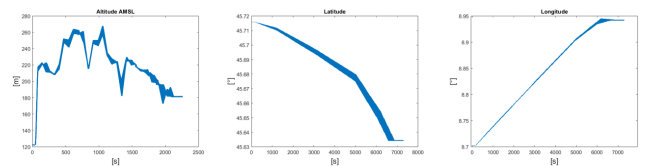


Figure 22: Cumulative Altitude AMSL, Latitude, and Longitude of the scanned corridor - test campaign 15/11/2023



Figure 23: Swarm advancing together (left) and hovering on the destination area (right) as seen from the POS - test campaign 15/11/2023

6.4 Additional Workload Estimation

The corridor scan mission was also repeated in another test on 28/06/2024, with the only difference from the main test campaign being the inclusion of eye-tracking software. This was done to estimate the additional workload caused by the cooperation with the unmanned system. In this mission profile, the POS can be treated as an additional navigational aid that provides real-time data gathered by the UAVs.

The data for pilot gaze was gathered throughout the flight at around 300 Hz frequency, and classified between the focus regions outlined in Section 5.1. A classification of a 5 minute long fragment representative of cruise flight following the UAV swarm is presented in Figure 24. The exact classification is as follows:

- Outside view: 47 times, 144 seconds
- Instruments: 44 times, 139 seconds
- Pilot Onboard Station: 13 times, 17 seconds

The POS, the element introduced for unmanned cooperation, was focused on for 5.5% of the total time, confirming the subjective assessment that it is intuitive to use and does not require much attention. From the pattern in Figure 24 one can recognize that the POS was only viewed occasionally to verify flight conditions, unlike instruments and outside view that form the main ‘scan loop’ for the pilot.

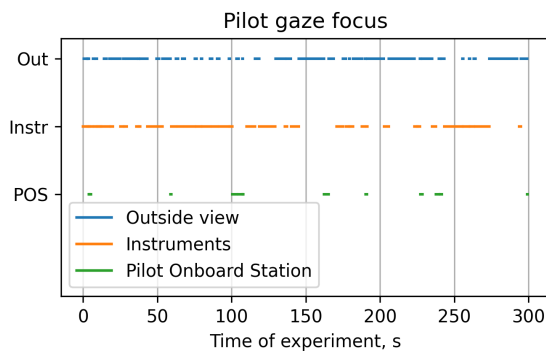


Figure 24: Pilot's eye focus point in flight - test 28/06/2024

7 CONCLUSIONS

A simulated mission system for MUM-T operational scenarios was prototyped and tested, in the context of a he-

licopter flight simulator and a simulation platform for up to 8 UAVs. In light of the results and the positive outcome of the performed test campaign, this architecture allows to interact with the drones realistically and in a satisfying manner, with minimum additional workload caused by the cooperation with the unmanned system.

Platforms like this may serve to gain further confidence in the management of a drone swarm for test pilots and other flying personnel, to understand specific dynamics in Crewed-Uncrewed applications, and to spread the knowledge of this concept as a useful and feasible technology to speed up a large variety of mission profiles and simplify complex tasks.

8 FUTURE DEVELOPMENTS

Future developments will address the possibility to simulate other classes of UAVs, like fixed-wing aircraft and unmanned helicopters. Implementing more sophisticated path planning algorithms is another milestone, so to align the implemented logics to the current state of the art.

The realization of certifiable HMIs, implemented using certifiable software like VAPS XT⁹, is another necessary step to obtain a more mature and closer to real operations system architecture.

REFERENCES

- [1] L. Thomassey, L. Arlen: “*TEAMX or Manned and Unmanned cooperation*”, Vertical Flight Society's 77th Annual Forum & Technology Display, 2021, Vertical Flight Society. DOI:10.4050/F-0077-2021-16819.
- [2] A. Avi, N. Frisco, M. Giurato, M. Lovera, P. Masarati, S. Panza, G. Parnisari, F. Roncolini, M. Sesana, G. Quaranta: “*Scout Drone: a Drone-Helicopter Collaboration to Support HEMS Missions*”, 2022, 48th European Rotorcraft Forum, Winterthur, Switzerland, 6–8 September, paper 112, ISBN:9781713870296.
- [3] R. Strenzke, J. Uhrmann, A. Benzler, F. Maiwald, A. Rauschert, A. Schulte: “*Managing cockpit crew excess task load in military manned-unmanned teaming missions by dual-mode cognitive automation approaches*”, AIAA Guidance, Navigation, and Control Conference, 2011. DOI:10.2514/6.2011-6237.
- [4] H. Xue, Q. Zhang, X. Zhang: “*Research on the Applicability of Touchscreens in Manned/Unmanned Aerial Vehicle Cooperative Missions*”, Sensors, 2022, MDPI. Basel, Switzerland. DOI:10.3390/s22218435.
- [5] T. Mehling, F. Viertler, T. Paul: “*Potential of the Unmanned Wingman in Manned-Unmanned Teaming tested in flight*”, Deutscher Luft- und Raumfahrtkongress 2019, Darmstadt, 2019. DOI:10.25967/490010.

⁹VAPS XT, <https://pace.txtgroup.com/products/vaps-xt>

- [6] S. Kim, Y. Kim: "Development of MUM-T Integrated Simulation Platform", IEEE Access 2023, Gyeongsang National University, Jinju 52828, Republic of Korea. DOI:10.1109/ACCESS.2023.3248096.
- [7] F. Roncolini, G. Galante, G. Quaranta, P. Masarati: "Constrained path planning for manned-unmanned rotorcraft teaming in emergency medical service missions", CEAS Aeronautical Journal, July 2024. DOI:10.1007/s13272-024-00753-7.
- [8] S. Shah, D. De, C. Lovett, A. Kapoor: "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles", In book Field and Service Robotics (pp.621–635), Microsoft Research, 2018, DOI:10.1007/978-3-319-67361-5_40.
- [9] P. Masarati, M. Morandini, P. Mantegazza: "An efficient formulation for general-purpose multibody/multiphysics analysis", J. Comp. Nonlin. Dyn., 2014, 9(4):041001, DOI:10.1115/1.4025628.
- [10] P. Masarati, A. Zanoni, G. Quaranta: "Physics-Based Piloted Flight Simulation Using Multibody Dynamics", DETC2016-60385, ASME IDETC/CIE 2016, Charlotte, NC, USA, August 21–24, 2016.
- [11] A. Zanoni, L. Conti, P. Masarati: "FRAME-Sim: a Free-Software, Multibody-Based, Pilot in the Loop Rotorcraft Flight Simulator", DETC2018-85289, ASME IDETC/CIE 2018, Quebec City, Canada, August 26–29, 2018.
- [12] M. Daniele, G. Quaranta, P. Masarati, A. Zanoni: "Pilot in the Loop Simulation of Helicopter-Ship Operations Using Virtual Reality", l'Aerotecnica, Missili & Spazio, 2020, 99(1):53–62, DOI:10.1007/s42496-020-00037-3.
- [13] M. Kassner, W. Patera, and A. Bulling: "Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction", 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Seattle Washington, September 2014, pp. 1151–1160. DOI:10.1145/2638728.2641695.
- [14] J. Wang, E. Olson: "AprilTag 2: Efficient and robust fiducial detection", 2016 IEEE/RSJ International Conference On Intelligent Robots And Systems (IROS), Daejeon, Korea (South), 2016, pp. 4193–4198. DOI:10.1109/IROS.2016.7759617.
- [15] K. Dorling, J. Heinrichs, G. G. Messier, S. Magierowski: "Vehicle Routing Problems for Drone Delivery", IEEE Transactions on Systems, Man, and Cybernetics: Systems (Volume: 47, Issue: 1, January 2017). DOI:10.1109/TSMC.2016.2582745.
- [16] Z. Wang, J. Sheu: "Vehicle routing problem with drones", Transportation Research Part B: Methodological Volume 122, April 2019, Pages 350-364. DOI:10.1016/j.trb.2019.03.005.
- [17] N. Ahn, S. Kim: "Optimal and heuristic algorithms for the multi-objective vehicle routing problem with drones for military surveillance operations.", Jimo, Journal of Industrial and Management Optimization, 2022, DOI:10.3934/jimo.2021037.
- [18] INCOSE "Systems Engineering Handbook - a Guide for system life cycle processes and activities," Wiley, 2023, Fifth Edition, ISBN: 978-1119814290.