



HAL
open science

UAVs that fly forever: Uninterrupted structural inspection through automatic UAV replacement

Milan Erdelj, Osamah Saif, Enrico Natalizio, Isabelle Fantoni

► To cite this version:

Milan Erdelj, Osamah Saif, Enrico Natalizio, Isabelle Fantoni. UAVs that fly forever: Uninterrupted structural inspection through automatic UAV replacement. *Ad Hoc Networks*, 2019, 94, pp.101612. 10.1016/j.adhoc.2017.11.012 . hal-02184513

HAL Id: hal-02184513

<https://hal.science/hal-02184513>

Submitted on 16 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UAVs that Fly Forever: Uninterrupted Structural Inspection through Automatic UAV Replacement

Milan Erdelj, Osamah Saif, Enrico Natalizio and Isabelle Fantoni

*Sorbonne Universités
Université de Technologie de Compiègne
CNRS UMR 7253 Heudiasyc
CS 60 319 - 60 203 Compiègne cedex, France*

Abstract

The field of structural inspection acquired a new boost with the development of UAVs (Unmanned Aerial Vehicles). However, the flight time of a UAV is still short compared to the time needed to perform a complete structural inspection, and therefore the reliability of a single UAV is under question. This paper presents an algorithm that allows an Unmanned Aerial System (UAS) to provide continuous uninterrupted structural inspection service. MAVLink protocol is extended with a set of messages and commands that allow the implementation of the proposed algorithm. The proof-of-concept simulation and implementation on UAVs show that the algorithm is suitable for the use in multi-UAV waypoint mission dedicated to structural inspection.

Keywords: Unmanned Aerial Vehicles; Autonomous Distributed Systems; Continuity of Service; Structural Inspection; MAVLink

1. Introduction

In the last decade, the potential applications of Unmanned Aerial Vehicles (UAVs) have attracted the attention of researchers, practitioners and enterprises. For example, electrical grid inspection is achieved by using human-piloted helicopters or line inspectors who walk or drive along the power line corridor. Often, no roads exist along the power line corridors, thus inspectors can only inspect less than 10km a day by walking. The field of structural inspection got a new boost with the development of UAVs. Ideally, a UAV can act as an "eye in the sky" for an inspection specialist, whose focus should be to inspect the power lines, and not to pilot the UAV.

Such operations should also be possible when the operator does not have direct line of sight with the UAV.

However, the flight time of an UAV is still short compared to the time needed to perform a complete structural inspection, and therefore the reliability of a single UAV is under question. The use of multiple cooperating UAVs is promising in order to accomplish complex tasks that are impossible, economically non viable, or inconvenient to be completed by a single UAV. In order to understand the broad topic of Ad-Hoc networks relying on the use of UAVs, the reader should consult [6], where the differences between FANETs (Flying Ad-Hoc Networks), MANETs (Mobile Ad-Hoc Networks) and VANETs (Vehicular Ad-Hoc Networks) are clarified and design challenges are presented.

This work is part of the project AIRMES[1], which aims at providing the solution to structural inspection (notably electrical power lines and railway infrastructure) with a fleet of cooperating heterogeneous UAVs. Done in the framework of the project AIRMES, this work proposed a lightweight algorithm for automated airborne operation based on MAVLink messaging protocol [3] that allows the continuous structural inspection service.

The contributions of this paper are the following:

1. We present the algorithm that automates the replacement of the UAVs thus providing the uninterrupted service to the user;
2. We extend the widely used communication standard for commercial UAVs (MAVLink [3]) with a set of messages and commands that allow the implementation of the proposed algorithm;
3. We provide a proof-of-concept implementation on UAVs to validate the algorithm and assess its performance.

The paper is structured as follows. First, we discuss the continuity-of-service issues with multiple UAVs in Section 2 and analyze the related work in Section 3. The overall system architecture is presented in Section 4, the proposed continuity-of-service algorithm is presented and analyzed in Section 5 and the proof-of-concept simulation results are provided in Section 6 and the algorithm performance is evaluated in Section 7. Section 8 discusses some potential future works on this topic. Finally, the conclusions are drawn in Section 9.

2. Continuity of service with multiple UAVs

With the recent development of a UAV technology, commercial and professional drones are more and more used for structural inspection [15]. The problem that all the electrically powered UAVs encounter is the inability to operate on a long time scale due to the limited battery capacity. In that sense, most of the commercially available UAVs for structural inspection achieve up to 25-30 minutes of airborne operation, after which a UAV should return to the launch point and replace or recharge its battery.

In order to cope with those limited operation time issues, different techniques are employed, including the use of different power sources or UAV types. The use of internal combustion engines (that consequently increases the size, cost and the operational risk of such a drone) can indeed prolong the airborne operation, however it introduces a completely new set of issues regarding its increased size, cost and risk during the operation. Different UAV types, such as fixed-wing UAVs, allow increased area coverage, however they are not suited for the applications of precise structural inspection.

Focusing on the structural inspection, the tool needed is the precisely controllable UAV that allows integration of different structural inspection sensors, such as lidar (light-based radar), multispectral cameras, robotic arms or manipulators. These extensions increase the battery drain and thus lower the time of airborne operation.

In the case of larger structural inspection operations, there is a need for longer airborne operation, that is traditionally achieved through iterative inspection by segments, followed by recurring battery charging cycles. Another approach relies on the use of multiple UAVs, that requires multiple operators present on the inspection locations, which increases the cost of the complete operation. Aforementioned issues require an automated solution that would allow an uninterrupted airborne operation.

For example, the task of cooperative surveillance of pre-selected areas of interest in outdoor environments by groups of closely cooperating micro-UAVs is tackled in [16]. Similarly, a platform for the creation of swarms of multiple drones is presented in [8]. It is based on commercially available quad-copters enhanced with on-board processing and communication units enabling full autonomy of individual drones. In order to achieve continuous long-range communication relay infrastructure, artificial potential field based path planning of UAVs is discussed in [9]. In [11], authors propose the use of the self-organizing UAVs for the application in disaster management.

In order to be operated, a suitable Unmanned Aerial System (UAS) comprises several UAVs, a ground control station (GCS) and the communication

links. Such a system should be designed in order to meet some specific requirements, such as having a sufficient flight time to perform a meaningful inspection, having robust low-level control algorithms for autonomous flight in turbulent weather conditions, being capable of detecting and avoiding obstacles, incorporating Flight Safety Modes to deal with GPS loss, communications losses and hardware failures. In the context of UAS, some of the works integrate Wireless Sensor Networks (WSN) together with UAVs in order to leverage the advantages of both sub-systems [12].

Our approach on implementing an uninterrupted airborne operation is based on the use of multiple UAVs running the continuity-of-service algorithm that create a *virtual* uninterrupted UAV service, offered to the user. In that manner, the system presented in this work provides a user with a transparent structural inspection experience all along the airborne operation. Therefore, the user can uninterruptedly control and retrieve the data from a *virtual* UAVs composed of several interchangeable UAVs that run the continuity-of-service algorithm. In general, whenever a UAV reaches its energy limits, it is automatically replaced by other available UAVs without user's involvement in the replacement process, thus creating a constantly available *virtual* UAV. The concept of *virtual* UAVs is essential for a general uninterrupted service provided to the user, since it solves the problem of an individual UAV autonomy.

3. Related Works

Several works propose different approaches to prolonging the UAV mission duration. For instance, in [4], the authors propose a distributed approach to solve long endurance area surveillance missions with a fleet of UAVs by considering communication constraints. Each UAV in the fleet has its own separate task, for example, surveying a dedicated area. In order to have a fault-tolerant system, UAVs tasks are exchanged periodically, using one-to-one coordination approach between neighbors, to ensure an appropriate reaction to changes in the size of the team. When a UAV has to drop its task, a neighboring UAV would ensure both its task and the leaving UAV's one. This solution adds more load on the replacing UAV, which leads it to consume more energy and time. Furthermore, a task negotiation strategy becomes essential to make the scheme work.

A theoretical optimal scheduling replacement strategy between UAVs, that performs a perimeter surveillance, is proposed in [7]. The proposed solution is optimal since it uses a centralized algorithm that possesses all the information about active and standby UAVs, to solve the problem of

replacement. A similar centralized algorithm is proposed in [14], where a recharge scheduling problem of a fleet of UAVs is addressed. For this purpose, the authors propose to use mobile charging robots for a UAV to dock and recharge its battery. An optimal algorithm for scheduling the UAVs recharges in order to perform the given mission is proposed. Several results addressing the possibility of achieving a persistent service provision from multiple UAVs based on scheduled recharging, are presented in [5] and [13]. The solution we propose is not based on a predefined schedule, but takes into account the current battery level in order to reactively perform a replacement.

In [18], the authors designed a mechatronics system for battery change and recharge for small UAVs. This system is used to ensure a persistent presence of multiple UAVs in a mission by reducing efficiently their downtime due to automatic replacement of batteries. When a UAV reaches its critical voltage threshold, it returns to the charging station to quickly replace its battery and go back to the mission. The drawback of this method is that a UAV in a battery-replacement phase should leave the mission with no backup, which reduces the performance of the mission and interrupts the service provided by the leaving UAV. However, in our work the UAV will be replaced instantly during the mission, which ensures the continuity of service. In fact, the proposed system in [18] could optimize the global objective of our proposed framework by using it in the charging phase.

In [17], the authors propose an experimental framework to solve the problem of persistent UAV service. They proposed a Mixed Integer Linear Programming (MIPL) model to plan the operations and the replacement of several UAVs, in order to ensure persistent missions. They validate their algorithm on an experimental test bench composed of two UAVs and 3 battery-charging stations. The UAVs use image-base localisation method to find their targets, pursue the mission and to land on the charging stations.

In fact, the work in [17] is original especially with the experimental validation of the proposed method with no need to global localization sensors such as Vicon or GPS. However, they use a scheduling model, based on split jobs, for UAVs replacement, which means that the replacement process is predefined before the start of the mission. In other words, the replacing UAV knows where and when exactly it will replace the other UAV. Moreover, it is not clear if collision avoidance aspect are taken into consideration for the hand-off phase.

In our scheme, we use a reactive, not predefined, replacement of UAVs based on a threshold battery level. In fact, when a replacing UAV is elected, it will approach the UAV to be replaced, while the latter is still providing

the requested service. Then, a positioning and handover phase will occur taking in consideration the collision avoidance aspects.

The closest idea to what we propose is presented in [10], where authors propose a set of additional MAVLink commands for communicating between swarm UAVs and a ground station. A set of MAVLink commands and messages are added to the MAVLink open-source library and are offered for public users. The addition of messages and commands are based on an imaginative scenario of negotiations between multiple-UAVs, ground control station and intelligent recharging station, for the purpose of swarm maintenance. However, no simulation or experimental implementation of the proposed extension are given in this work. Moreover, in this work, the replacement process is not reactive and does not ensure the continuity of service, that is, a replaced UAV leaves its mission without immediate replacement by another UAV.

Table 1 summarizes a comparison between related works in literature and our work.

Table 1: Comparison between related works

Work	Year	Simulation	Implementation	Replacement	Continuity of service
[5]	2004	✓	✗	✗	✓
[13]	2006	✓	✗	✗	✗
[4]	2013	✓	✗	✗	✓
[17]	2014	✗	✓	scheduled	✓
[14]	2015	✓	✗	scheduled	✗
[18]	2015	✗	✓	reactive	✗
[10]	2016	✗	✗	not specified	not specified
[7]	2017	✗	✗	yes	✗
this paper	2017	✓	✓	reactive	✓

4. System Architecture

The basic infrastructure of the communication system is composed of a Ground Control Station (GCS) that is used by an operator to control the

group of UAVs. Figure 1 illustrates the GCS communication with two UAVs. Each of the parties in the communication contains the same communication blocks: UDP broadcast, UDP unicast and UDT¹ unicast block.

The UDP communication blocks implement the header-only message marshaling library MAVLink [3], a protocol dedicated for communicating with small unmanned vehicles that is widely used for commercial UAVs. MAVLink is intended to be used as a communication protocol between the ground control station or a controller, and the UAV. The full list of implemented messages and commands is available at the MAVLink website², which can be easily extended. One of the main contribution of this paper is that it proposes a set of messages and commands that are needed for multi-UAV applications and that allow the cooperation of the UAVs in a fleet.

Table 2: Available information about each UAV contained in the following MAVLink messages: HEARTBEAT, SYS_STATUS and LOCAL_POSITION_NED.

HEARTBEAT #0		SYS_STATUS #1		LOCAL_POSITION_NED #32	
type	field	type	field	type	field
uint8_t	type	uint32_t	sensors_present	uint32_t	time_boot_ms
uint8_t	autopilot	uint32_t	sensors_enabled	float	x
uint8_t	base_mode	uint32_t	sensors_health	float	y
uint32_t	custom_mode	uint16_t	load	float	z
uint8_t	system_status	uint16_t	voltage_battery	float	vx
uint8_t	mavlink_version	int16_t	current_battery	float	vy
uint64_t	last_update	int8_t	battery_remaining	float	vz
		uint16_t	drop_rate_comm	uint64_t	last_update
		uint16_t	errors_com		
		uint64_t	last_update		

UDP broadcast block is used for the neighborhood discovery and the broadcast of the fleet commands from the GCS. Table 2 shows the contents of the 3 MAVLink messages, HEARTBEAT, SYS_STATUS and LOCAL_POSITION_NED, which are periodically broadcast in the network by all the communicating parties, thus allowing the other receiving parties to construct and regularly update the information about all the their neighbors in the network. Additionally, the GCS broadcasts additional MAVLink fleet commands that are

¹UDP Transmission Protocol.

²<http://mavlink.org/messages/common>

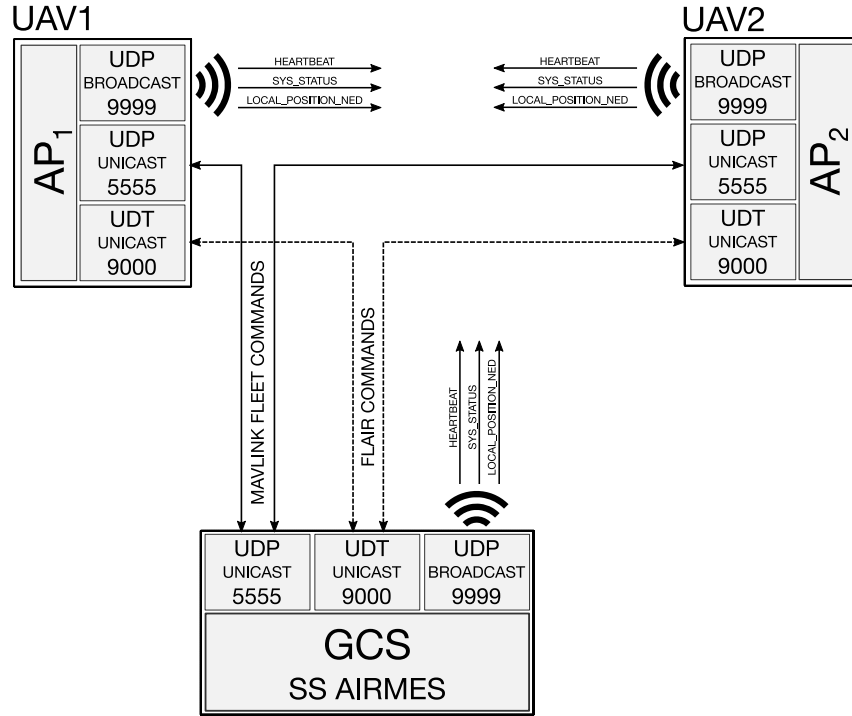


Figure 1: Communication system architecture consisting of a Ground Control Station (GCS) and multiple UAVs. Both GCS and UAVs contain the same communication infrastructure composed of UDP broadcast and unicast communication modules.

listed in Table 3.

UDP unicast block is reserved for the direct communication between the GCS and each individual UAV in the fleet, in the sense of sending commands to, and receiving command acknowledgements from each individual UAV for the sake of communication reliability.

UDT unicast block is reserved for the internal communication among the processes of the FL-AIR framework, presented in Section 6.1, and will not be discussed in detail. Interested reader can consult the FL-AIR website [2].

Figure 2 shows the different modules of the UAV and GCS. The UAV system is composed of four modules:

- *Autopilot*: All the low level algorithms of controlling the UAV, such as, rotational and translational dynamics control, state estimation from sensors and security check, trajectory following and the control state

Table 3: Additional MAVLink messages dedicated to continuity-of-service algorithm (blue) and the chosen MAVLink messages from the common set (green).

ID	Type	MAVLink Msg	BCAST	Parameters	Comment
0	MSG	HEARTBEAT	✓	<MAVLink Common Set>	These three messages are periodically broadcast and used for neighborhood discovery in the UAV fleet.
1	MSG	SYS_STATUS	✓	<MAVLink Common Set>	
32/33	MSG	LOCAL_POSITION_NED	✓	<MAVLink Common Set>	
39	MSG	MISSION_ITEM		<MAVLink Common Set>	Messages and commands from the common MAVLink message set that are used in the algorithm implementation.
42	MSG	MISSION_CURRENT		<MAVLink Common Set>	
47	MSG	MISSION_ACK		<MAVLink Common Set>	
32	CMD	MAV_CMD_DO_FOLLOW		<MAVLink Common Set>	
200	CMD	MAV_CMD_DO_CONTROL_VIDEO		<MAVLink Common Set>	
153	MSG	FLEET_UAV_REQUEST	✓	uint8_t request_type uint8_t request_ID uint8_t ID_to_replace uint32_t latitude uint32_t longitude uint16_t altitude uint8_t request_urgency	Additional MAVLink messages for the UAV fleet management that are used in algorithm. These messages represent a subset of custom MAVLink messages intended to be used for a fleet management in the project AIRMES.
155	MSG	FLEET_UAV_REQUEST_RESPONSE		uint8_t request_ID uint64_t arrival_time	
160	MSG	FLEET_UAV_ELECTED_ID	✓	uint8_t request_ID uint8_t elected_ID	
161	MSG	FLEET_UAV_ELECTED_ACCEPT		uint8_t request_ID	
162	MSG	FLEET_REPLACEMENT_IN_POSITION		uint8_t request_ID	
180	MSG	FLEET_TARGET_STREAM_READY		uint8_t target_ID	
40001	CMD	FLEET_UAV_CLEAR_TO_LEAVE		uint8_t request_ID	

machine are done in this module.

- *Guidance and formation module*: Formation control algorithms and trajectory generation to achieve a task or a mission as well as a part of the UAV replacement state machine in coordination with the *MAVLink module* are implemented in this module.
- *MAVLink module*: The role of this module is the interpretation and the construction of MAVLink messages from or to the *Guidance and formation module*.

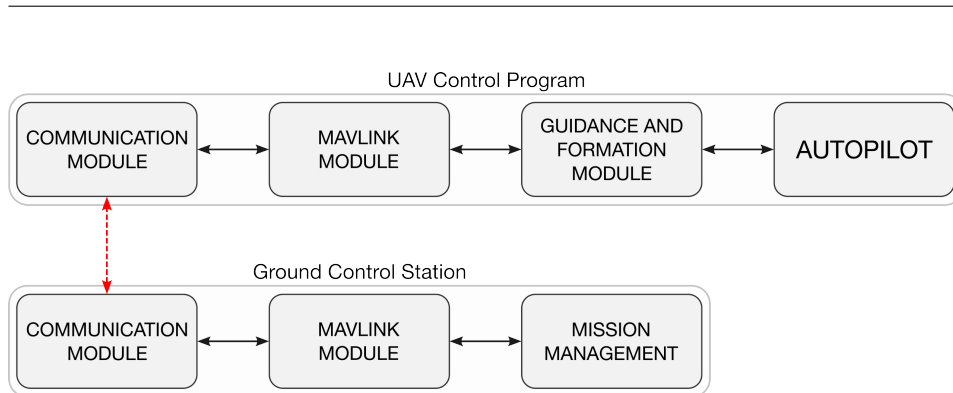


Figure 2: Modular architecture of a UAV and GCS for fleet management.

- *Communication module*: Communication protocol management is done in this module, more details are in Section 5.

The *GCS* system is composed of three modules:

- *Mission management*: missions planning, high level commands such as *start mission*, *pause mission*, *continue mission*, *return to launch*, etc. and a part of the UAV replacement state machine are implemented in this module.
- *MAVLink module*: The interpretation of MAVLink messages from UAVs and the construction of high level commands from the *Mission management module* in order to be sent to UAVs, are performed in this module.
- *Communication module*: Its role is similar to the *Communication module* in the UAV system, and more details could be found in Section 5.

5. Continuity-of-Service Algorithm

The continuity-of-service state machine is presented in Figure 3. Every UAV among the available UAVs can take one of the states presented in the state machine. The main states of an active UAV are the following:

- ① **Standby**: The current UAV in this state is turned on and waiting on the launch position for a mission request from the controller. If the energy threshold is reached in this state, the UAV passes to **Charging** state. In case a replacement request or mission request are received, the UAV switches to **Replacement offer** or **Mission offer** states, respectively.

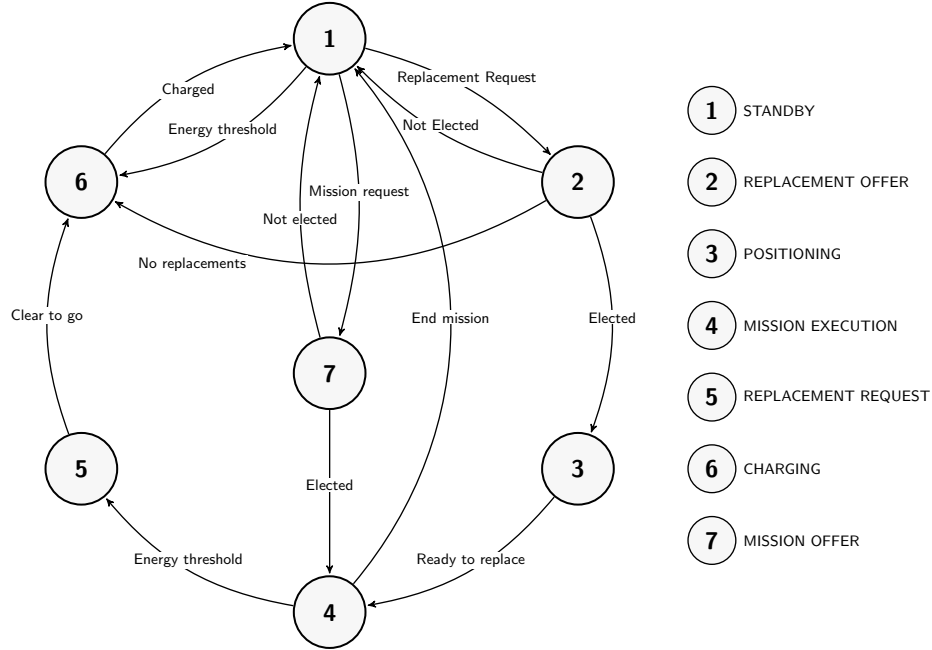


Figure 3: Active UAV state machine.

- ② **Replacement offer:** Once a replacement request is received, the current UAV passes to Offering replacement state, where it responds with its estimated replacement arrival time. After the active UAV responds with an elected UAV identifier, the current UAV passes to **Standby** state if not elected, or to **Positioning** state if elected for replacement.
- ③ **Positioning:** In case the active UAV elects the current UAV for replacement, the current UAV passes to **Positioning** state where its goal is to position itself close by the active UAV based on the real-time information about the network neighborhood (details are presented in Section 5.2). The position state comprises two stages, absolute positioning and precise relative positioning based on the visual input³.
- ④ **Mission execution:** The UAV in this state is referred to as an active UAV. In this state, the UAV executes the mission required by the oper-

³Details about the visual positioning are omitted due to the article length constraints.

ator. The example used in this work is the waypoint following algorithm (more details in Section 6).

- ⑤ **Replacement request:** The active/current UAV passes into this state if the energy threshold is reached during the mission execution. In this state a replacement UAV is requested and chosen based on the set of received replacement offers. The UAV remains in this state until it receives the clear-to-go message from a replacing UAV.
- ⑥ **Charging:** In this state the current UAV is connected to an automatic recharging station, or its battery is being manually replaced (the actual charging and battery replacement implementation can vary depending on the use case and available equipment). After the battery has been recharged, the UAV passes to **Standby** state.
- ⑦ **Mission offer:** The UAV enters this state upon receiving a mission request from the user/controller. It bids for the mission and awaits for the information about the elected UAV, passing to **Mission execution** state if elected, or passing to **Standby** state if not.

The most important states regarding the communication aspects of the continuity-of-service algorithm are the initial election of a UAV that will start the service, and the replacement negotiation that occurs when the UAV needs to be replaced due to the energy issues for example. Detailed sequence diagrams of these two states are presented in the following section.

5.1. Message Exchange

In order to achieve a real-time network neighborhood discovery with a system architecture presented in Figure 1, the following MAVLink messages are used to transmit an essential set of information: **HEARTBEAT**, **SYS_STATUS** and **LOCAL_POSITION_NED**. These messages are broadcasted periodically with a fixed frequency of 3 Hz, and convey the information about each UAV's position, system state, energy levels. The full list of parameters sent with these messages are presented in Table 2. In this work, the use of local coordinates is assumed. Therefore, the **LOCAL_POSITION_NED** message is used, whereas in case the global positioning system is available, the use of **GLOBAL_POSITION_INT** message is more appropriate.

In the following, we present in detail the sequence diagrams for the cases of UAV initial election for the mission and UAV replacement negotiation.

The initial election, presented in Figure 4, is initiated by the user via its Ground Control Station (GCS) where the UAV request is being broadcasted in the network of UAVs. Upon receiving the UAV request (MAVLink

message `FLEET_UAV_REQUEST`, listed in Table 3), each available UAV responds with its identifier and estimated arrival time. It is worth noting that these messages are sent as unicast messages towards the GCS. After choosing the UAV with shortest announced arrival time, GCS broadcasts the `FLEET_UAV_ELECTED_ID` message, thus informing all the awaiting UAVs about the elected one. The selected UAV then responds with `FLEET_UAV_ELECTED_ACCEPT` unicast message, while all the other UAVs return into `Standby` state. The mission parameters are then exchanged between the GCS and the elected UAV, following the MAVLink Waypoint protocol⁴.

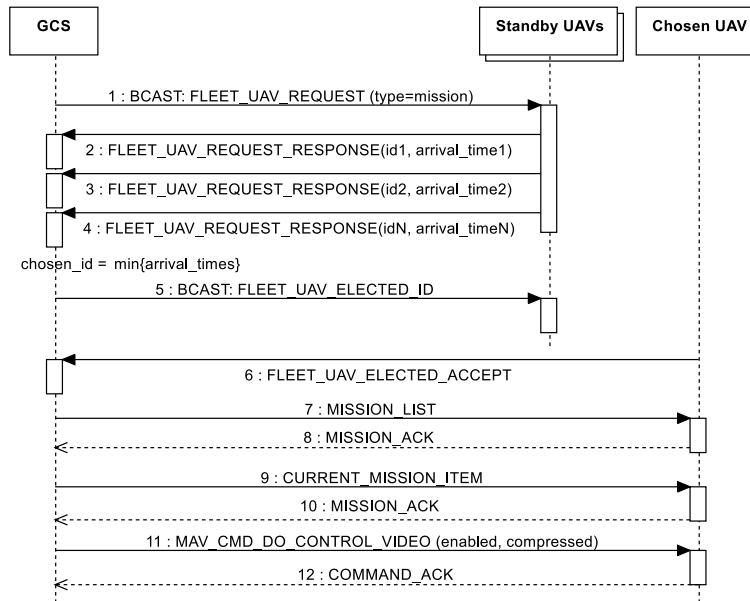


Figure 4: Initial election.

Replacement negotiation, presented in the Figure 5, is initiated by the active UAV that reached its energy threshold. The negotiation starts with the replacement request sent towards the GCS (`FLEET_UAV_REQUEST` with *replacement* as the type parameter). GCS then broadcasts the request in the network and all the UAVs respond with a `FLEET_UAV_REQUEST_RESPONSE` message containing their identifier and estimated arrival time.

⁴http://qgroundcontrol.org/mavlink/waypoint_protocol

Regarding the replacement request sent from the active UAV, two possibilities exist: 1) the active UAV sends the replacement request towards the GCS that then broadcasts the request, and 2) the active UAV directly broadcasts the replacement request in the network and awaits the responses. In this particular implementation, the latter is preferred since it allows the user to control the continuity-of-service algorithm and track the message exchange between the UAVs and GCS.

Based on the neighborhood table with the information regarding all the UAVs in the network, the GCS decides which UAV represents the best replacement, and broadcasts its choice in the `FLEET_UAV_ELECTED_ID` message. The elected UAV responds with the `FLEET_UAV_ELECTED_ACCEPT` unicast message, while all the other bidding UAVs switch back to `Standby` state. The GCS instructs the elected UAV to follow and replace the currently active UAV with `MAV_CMD_DO_FOLLOW` message and the replacement UAV passes into `Positioning` state.

Without going into further details, it is considered that in this state, a UAV relies on absolute positioning system (such as GPS, GLONASS, etc) to approach the active UAV, and then follows a visual navigation to precisely position itself in order to minimize the shift between the video streaming image after the link handover.

When both absolute and precise visual positioning took place, the replacement UAV sends the `FLEET_REPLACEMENT_IN_POSITION` message and the GCS transfers the mission information to it, including the current mission item that has been serviced by the previously active UAV. Finally, the GCS instructs the previously active UAV to proceed to charging station and pass into `Charging` state.

5.2. Addition to MAVLink Protocol

In order to manage the group of UAVs in this approach, it was necessary to add a set of customized messages for fleet management. Table 3 summarizes the MAVLink messages used in this algorithm, including necessary parameters for each message, and the indication whether the message should be broadcast or not.

The additional MAVLink messages used in this work are the following:

- `FLEET_UAV_REQUEST`: The message is broadcast from the GCS in order to acquire all the request responses from the currently available UAVs in the `Standby` state. The message contains the information regarding the request type, and in the case of a replacement request, the location and the identifier of the UAV that needs to be replaced. Although the

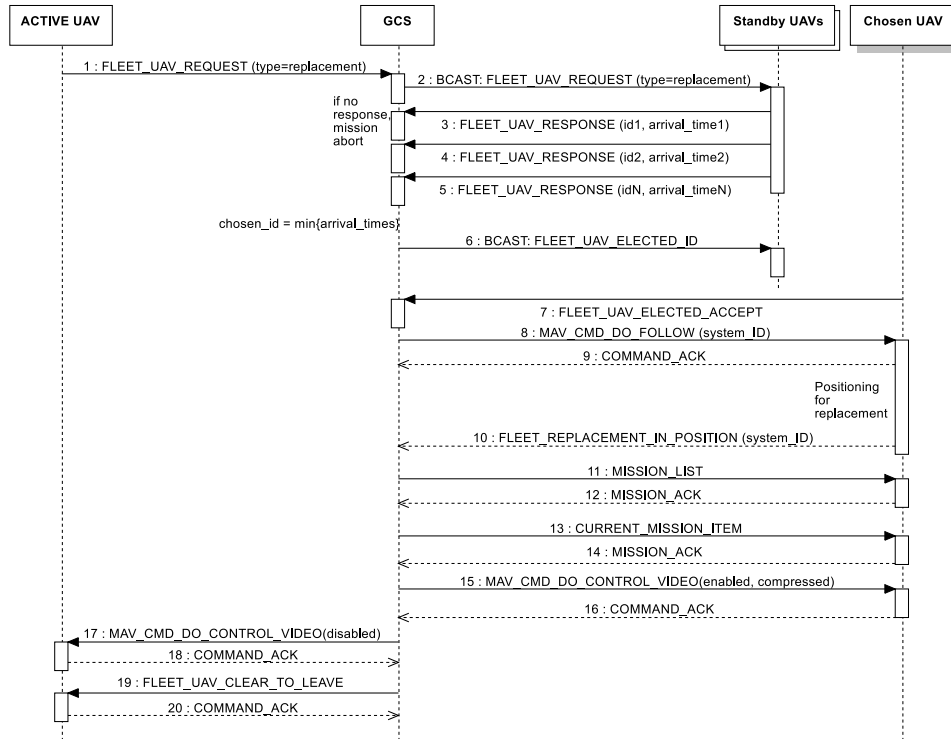


Figure 5: Replacement negotiation.

location is provided in the message, the updates to the location of the UAV could be consulted from the neighborhood table.

- **FLEET_UAV_REQUEST_RESPONSE:** Represents the answer to a received request message, with an estimated arrival time as a parameter.
- **FLEET_UAV_ELECTED_ID:** After the arrival times are sent individually by each available UAV, the GCS broadcasts this message in order to announce the UAV chosen for the task.
- **FLEET_UAV_ELECTED_ACCEPT:** This message represents a confirmation after the UAV has been selected by the GCS.
- **FLEET_REPLACEMENT_IN_POSITION:** The UAV has finished the positioning phase and is ready to start configuring the video stream. The message is sent from the UAV towards the GCS.

-
- **FLEET_TARGET_STREAM_READY**: The precise visual positioning has been completed, the video stream has been established. The message is sent from the UAV towards the GCS.
 - **FLEET_UAV_CLEAR_TO_LEAVE**: After the positioning process has been completed, and the video stream established, the GCS uses this command to inform the replaced UAV that it has the permission to leave the mission and connect to the charging station.

Table 4 presents the messaging cost in terms of the overall number of bytes that need to be exchanged. The table presents the data valid for the use case that assumes the use of 3 UAVs and 12 mission waypoints. *Basic info broadcast* represents the three messages that are broadcast periodically in order to announce UAVs' current status and position. The table presents the size of the messages supposing that the broadcast is done with the frequency of 4 Hz. Depending on the number of UAVs, the overall size of the broadcast messages should be multiplied accordingly.

Waypoint mission segment represents overall size of the messages needed to transmit the description of a waypoint mission using MAVLink commands listed in the table. Depending on the number of waypoints in the mission, the size of the waypoint mission segment varies. The waypoint mission is sent towards the active UAV in the beginning of the mission, as well as to the replacement UAV after the replacement took place.

6. Waypoint Mission Simulation in FL-AIR

This section provides details on the proof-of-concept simulation on the use case of a waypoint mission for structural inspection.

6.1. Framework Libre AIR

As a part of ROBOTEX⁵ project, Heudiasyc laboratory is equipped by a fleet of UAVs in order to carry out scientific research on autonomous flight and formation control. However, an autonomous flight in formation could be risky, that is why the laboratory developed a simulator of fleet of UAVs - Framework Libre AIR (FL-AIR). FL-AIR is a simulation framework written in C++ that aims at helping the development of applications for robots, and more specially for UAVs [2]. The goal of this simulator is to run on a computer, a code identical to that used in the real UAVs, to perform all the algorithms development steps safely.

⁵<http://equipex-robotex.fr>

Table 4: Overall size (in bytes) of the messaging segments of the algorithm.

Communication segment	MAVLink message	Bytes	Hz	Bytes/s	Total
Basic info broadcast	HEARTBEAT	17	4	68	368
	SYS_STATUS	39	4	156	
	LOCAL_POSITION_NED	36	4	144	
Communication segment	MAVLink message	Bytes	No.	Sum	Total
Waypoint mission	MISSION_ITEM	46	14	644	923
	MISSION_SET_CURRENT	12	1	12	
	MISSION_CLEAR_ALL	11	1	11	
	MISSION_ACK	12	17	204	
	MAV_CMD_DO_CONTROL_VIDEO	41	1	41	
	COMMAND_ACK	11	1	11	
Initial election	FLEET_UAV_REQUEST	22	1	22	92
	FLEET_UAV_REQUEST_RESPONSE	17	3	51	
	FLEET_UAV_ELECTED_ID	10	1	10	
	FLEET_UAV_ELECTED_ACCEPT	9	1	9	
Replacement negotiation	FLEET_UAV_REQUEST	22	1	22	225
	FLEET_UAV_REQUEST_RESPONSE	17	3	51	
	FLEET_UAV_ELECTED_ID	10	1	10	
	FLEET_UAV_ELECTED_ACCEPT	9	1	9	
	MAV_CMD_DO_FOLLOW	41	1	41	
	FLEET_REPLACEMENT_IN_POSITION	9	1	9	
	MAV_CMD_DO_CONTROL_VIDEO	41	1	41	
	FLEET_UAV_CLEAR_TO_LEAVE	9	1	9	
COMMAND_ACK	11	3	33		

For this purpose, a Linux system is installed on a PC. In the simulator, virtual sensors and actuators are connected to a discrete nonlinear model of a UAV. As a result, all UAVs' states are calculated at each instant of time. Each UAV in the fleet simulator is an independent computer process. Moreover, UAVs evolve in a 3D virtual environment, thanks to Irrlicht engine⁶. The program in the simulator is connected to a Ground Control Station

⁶<http://irrlicht.sourceforge.net>

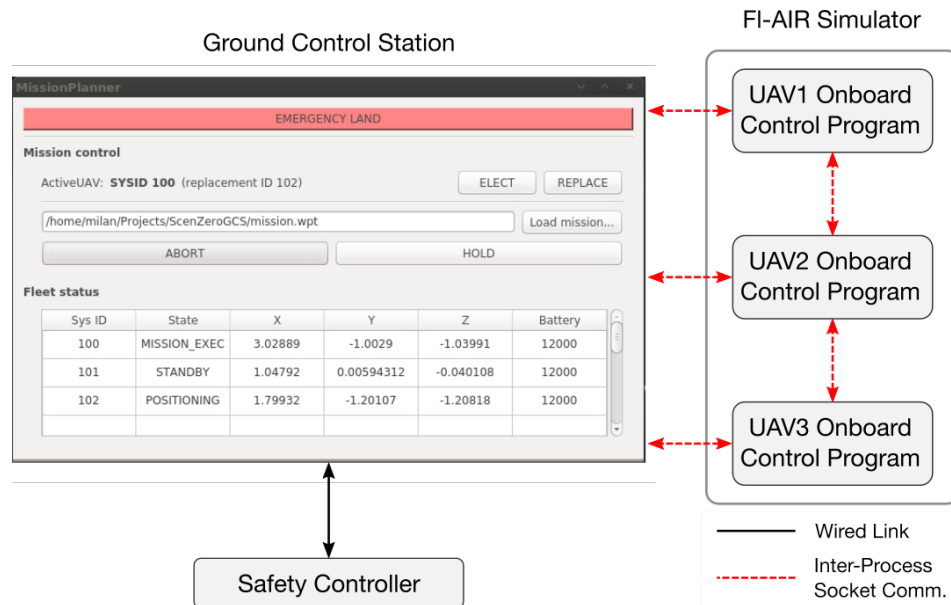


Figure 6: Architecture of the simulator of fleet of UAVs.

(GCS) program. The base station records and draws measurements, and is used to start and end simulations, and to set the parameters of UAVs and control laws. Figure 6 shows the architecture of the simulator.

Figure 7(b) shows the active UAV that passes through the set of provided waypoints, shown in Figure 7(a). Since there was no need for replacements during the mission, the other two available UAVs stayed in `StandBy` mode all along the mission duration.

If the replacement is needed during the mission execution, the active UAV requests a replacement following the aforementioned algorithm. In the example scenario (Figure 8(a)), the active UAV requests the replacement after servicing the third waypoint (WP3). The elected replacing UAV arrives after the active UAV has serviced the WP4, and continues towards the WP5, while the previously active UAV returns to the recharging station (launching site in this example).

In case the UAVs do not possess enough resources to complete the mission, it can be necessary to interrupt the mission multiple times in order to request a replacement. Figure 8(b) shows a scenario where all the 3 UAVs are needed to complete the mission. After the first UAV has started the mission and serviced 4 waypoints, the replacement arrives and continues from the WP5. Prior to arriving to the WP9, it requests another replacement,

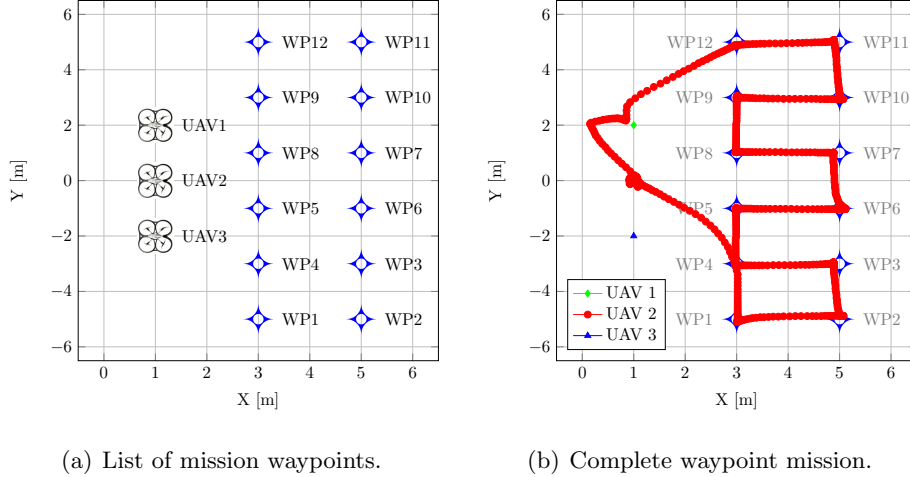


Figure 7: A mission is described by a list of mission waypoints that a UAV needs to visit.

and the third UAV arrives in time to service waypoints WP9 until WP12.

7. Algorithm evaluation

In this section we evaluate the performance of the proposed algorithm. First, we theoretically evaluate the mission continuity coefficient C (described thereafter), and then use the simulated environment to run our algorithm and to measure the obtained mission continuity coefficient.

Taking the inspiration from the commercial UAV industry and measuring the performance of a Parrot's AR.Drone 2.0 in particular, we took the following approximate values into consideration in order to evaluate the mission continuity coefficient:

- Mission execution time, $T_{mission} = 10$ min,
- Time of flight towards the first mission waypoint, or back towards the recharging/replacement station, $T_{flight} = 1$ min,
- Battery replacement time, $T_{replace} = 3$ min,
- Battery recharging time, $T_{charge} = 45$ min.

The choice of these particular values is based on the measured UAV performance in the laboratory experimental setting, where drones achieve 10-15 minutes of airborne operation. We assume that the mission waypoint

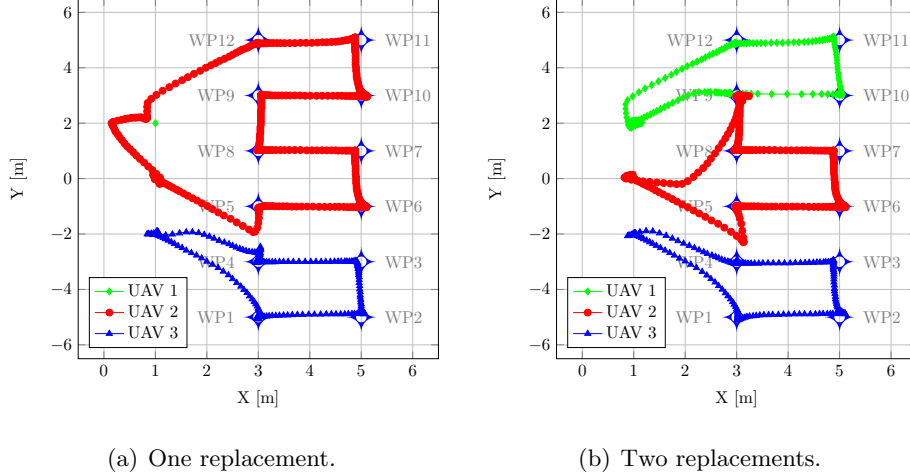


Figure 8: UAV trajectories for one and two replacements during the mission execution.

destination is reachable by a UAV in less than one minute, which means up to 50 m of distance from the launch point. Batteries that are provided with AR.Drones are fully charged after the period of 45 minutes (their capacity is 1500 mAh), and we assume that an operator can replace a depleted battery and reboot the UAV's onboard computer in less than 3 minutes.

The mission continuity coefficient C represents the measure of a continuity of service provided by a UAV or a set of cooperating UAVs. In the case when the UAV battery is being recharged during the mission execution, the continuity coefficient is calculated as follows:

$$C_{charge} = \frac{T_{mission}}{2T_{flight} + T_{mission} + T_{charge}}$$

Similarly, in the case where the batteries are being replaced during the mission execution, the continuity coefficient becomes:

$$C_{replace} = \frac{T_{mission}}{2T_{flight} + T_{mission} + T_{replace}}$$

Assuming the existence of landing pads with automated charging capabilities, or a presence of an operator that would put the depleted battery to a charger, the waypoint mission with one UAV is presented in Figure 9. It is evident that the mission is interrupted so that the battery can be charged, and taking into account the provided operation durations, the mission continuity coefficient value is $C_{charge} = 17.5\%$.

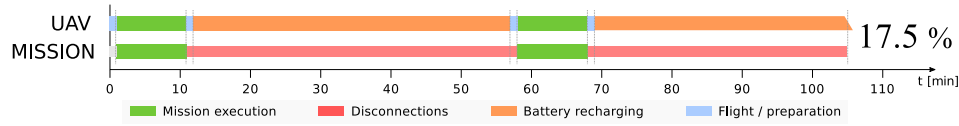


Figure 9: Mission continuity with a UAV battery recharging.

In a scenario where a UAV operator possesses a number of pre-charged batteries, a mission continuity coefficient with one operational UAV can be increased due to low battery replacement time (Figure 10). In this case, the mission continuity coefficient rises up to the value of $C_{replace} = 66.7\%$.

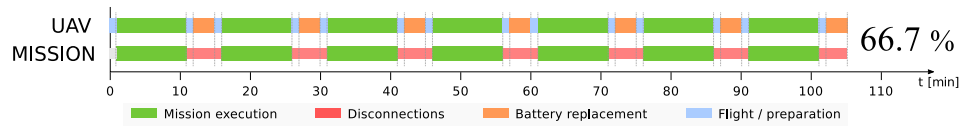


Figure 10: Mission continuity with a UAV battery replacement.

Although the approach with battery replacements increases the mission continuity coefficient, it still fails to provide a continuous service to a user. This is the justification for the use of a UAV fleet, where multiple UAVs can autonomously cooperate in order to provide a continuous service. The scenario of a multi-UAV usage with battery recharging is shown in Figure 11. In order to achieve a continuous mission, an operator would need to use a UAV fleet composed of 6 UAVs, taking into account the chosen timing values, especially regarding the battery recharge time. The proposed mission continuity algorithm running on all the UAVs, and a UAV station equipped with individual landing/charging pads can indeed provide the continuity of service, where the operator's task would be to assure the seamless operation of the fleet and act in the case of unpredicted UAV failures (that are out of the scope of this article). A disadvantage in this scenario is the need for individual automatic charging pads for each of the UAVs in the fleet, that increases the cost of the overall system. As a side note, a scheduling algorithm for a fleet of UAVs with a limited number of charging pads is an interesting research direction for future works.

In the scenario that assumes the availability of a number of pre-charged batteries, and an operator that could replace UAV batteries in a predefined maximal amount of time, a reduced number of UAVs is needed for a continu-

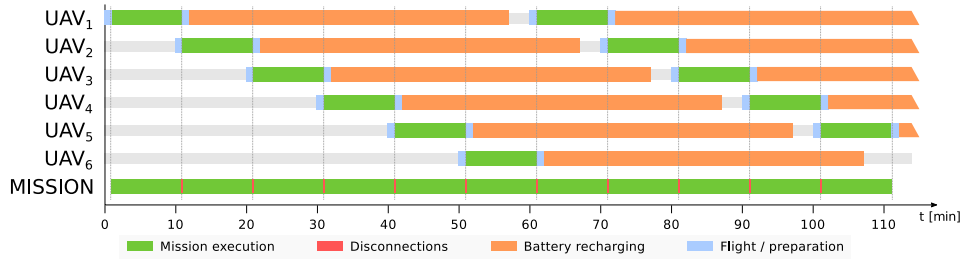


Figure 11: Multi-UAV mission with battery recharging.

ous mission. Figure 12 presents the time chart of that scenario. The obvious advantage is the use of fewer UAVs, however the role of the station operator is critical in replacing the batteries. The automated battery replacement mechanism presented in [18] could be indeed very useful in this context. A practical implementation requires the availability of enough pre-charged batteries, which means that the problem of charging multiple batteries still remains. However, this problem can be solved with additional dedicated battery chargers present at the UAV station.

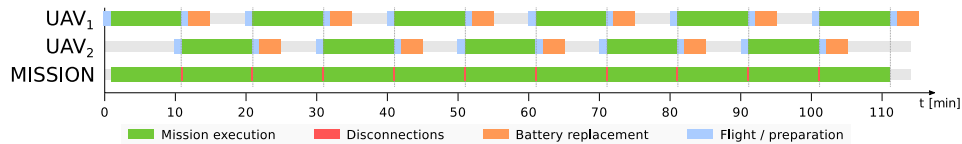


Figure 12: Multi-UAV mission with battery replacements.

In order to evaluate the mission continuity coefficient for the proposed continuity of service algorithm, we run a set of mission simulations with multiple UAVs. The idea is to send a sinusoidal signal from the active UAV towards a ground control station that records the received signal, so that it will be possible to evaluate the disconnection time during the signal handover among the UAVs as well as the overall signal delay after multiple handovers.

In more details, the sinusoidal signal generation is going to be implemented in the following manner: after receiving the command to start the mission, the active UAV starts incrementing an integer value at the frequency of 50 Hz, and starts sending it towards the ground station at the same frequency (therefore, each integer increment is sent towards the ground

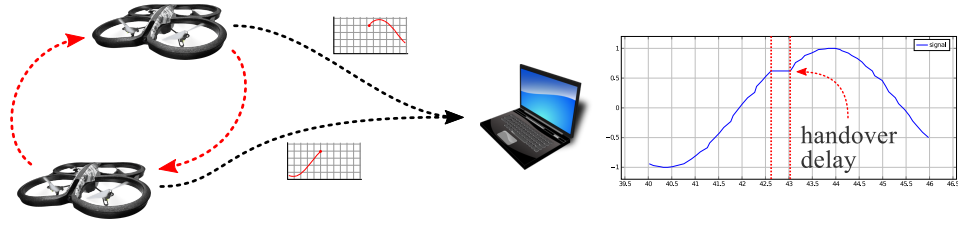


Figure 13: Multi-UAV signal handover illustration.

station). The ground station receives the integer, creates a sinusoidal value out of it and saves these values to a mission output file. The use of this simple technique could provide us the visual feedback on the signal quality as well as on the handover delay d_h that will be visible during the UAV handover (Figure 13). After multiple UAV handovers, we plot the received signal in comparison to the initial ideal sinusoidal signal, and then compute the mission continuity coefficient for the whole mission resembling the provided theoretical analysis with multiple recharging UAVs.

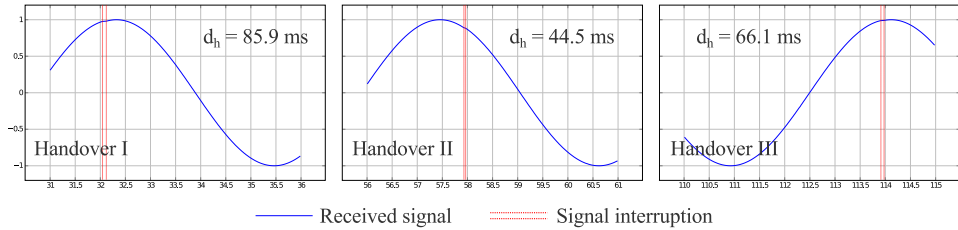


Figure 14: Measured handover delays d_h during the mission execution.

Since the only disconnections in this case are introduced by the mission handover, the mission continuity coefficient will in this case be calculated as:

$$C_h = \frac{T_{mission}}{T_{mission} + d_h^{max}}$$

Figure 14 shows the examples of measured values of delay handovers d_h during the handover algorithm implementation on UAVs. Measured handover delay values vary due to the network hardware and network conditions, and the implementation on UAVs in the laboratory environment showed the handover delays between $d_h^{min} = 38$ ms and $d_h^{max} = 112$ ms. In order to calculate the mission continuity coefficient, we add an arbitrary

confidence interval onto these measurements, and assume that in a worst case the handover delay will last for $d_h^{max} = 150$ ms. For a multi-UAV mission that lasts for $T_{mission} = 10$ min, the value of the mission continuity coefficient for handover algorithm is therefore $C_h = 99.975$ %, which we consider to be a satisfactory measure of the continuity of service and a relevant contribution of automatic replacement⁷.

8. Future Perspectives

As a part of future works on the topic of UAS for continuity-of-service in structural inspection missions, we foresee the use of unified UAV system integrating multiple UAVs in a fixed or mobile UAV station. In an example scenario of UAV usage for structural inspection, we propose the use of such UAV stations equipped with a fixed-wing and rotary-wing UAVs.

A conceptual design of a fixed (or static) UAV station is presented in Figure 15(a). The concept proposes the use of multiple heterogeneous UAVs, where fixed-wing UAVs are used for long distance and wide area surveillance (in the case of railways or power lines inspection), and the rotary-wing UAVs are used for precise inspection tasks. Bearing in mind the autonomy, duration of airborne operation and recharging cycles, we estimate that we would need 4-5 UAVs per station, to which we add an extra UAV in order to offer sufficient redundancy for the continuity-of-service algorithm. The fixed UAV station could be connected to a wired Internet link, as well as to a constant power supply for UAV recharging.

A mobile UAV station can be envisioned as a vehicle with a storage space for multiple rotary-wing and fixed-wing UAVs, equipped with a long-distance communication antenna, electricity generator and a system for automatic UAV battery recharging (Figure 15(b)). The mobile UAV station could be operated by a single human operator, mostly to maintain the station and to act as a safety supervisor, if something goes wrong during the UAV network operation. The main advantage of the mobile UAV station is its flexibility regarding the location of the structural inspection mission. The disadvantage is the necessity of local power generation and well as the cellular or satellite internet antenna that needs to be installed on the vehicle, which significantly increases its cost.

The proposed UAV stations could also implement an approach for automatic battery replacement, together with an approach for vision-based

⁷The video of the UAVs behavior during the handover algorithm, and further implementation details are available at <https://www.hds.utc.fr/~erdeljmi/>

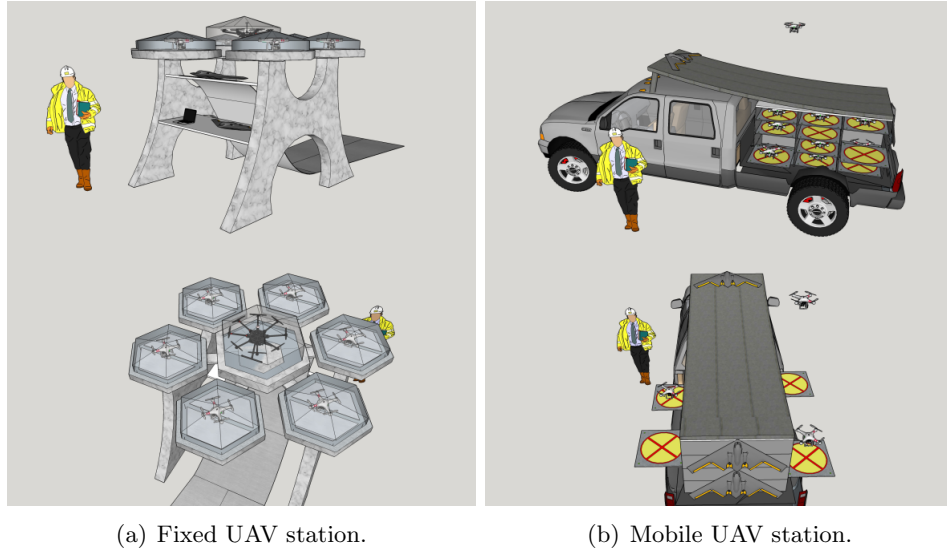


Figure 15: Conceptual UAV stations allowing a continuous operation of UAVs.

formation control in order to allow a simplified yet effective control of a group of UAVs. We assume that the system can rely on the GPS positioning, while the operator can manually correct the hovering position of a UAV based on the multimedia input. A good UAV management software can allow the UAV network to be operated to a minimal number of personnel. According to the extension of the area to monitor, the resources needed would grow, but still the system will be feasible and rapidly deployable.

9. Conclusion

This article intended to present a continuity-of-service algorithm based on its practical implementation linked to the extension of MAVLink protocol. The implementation details and the proof-of-concept simulation show that the algorithm is suitable for the use in multi-UAV waypoint mission dedicated to structural inspection. The contributions pointed out in the article are the following:

1. Description of the algorithm that automates the replacement of the UAVs thus providing the uninterrupted service to the user;
2. Extension to the MAVLink protocol with a set of messages and commands that are needed for multi-UAV applications and that allow the cooperation of the UAVs in a fleet;

-
3. In order to validate the algorithm and assess its performance, a proof-of-concept implementation on UAVs is provided.

We are currently working on the mathematical model of the continuity-of-service approach as well as in depth evaluation study including the UAV swap duration, extended flight time, network overhead, minimal number of UAVs needed for an unlimited network lifetime, and maximal network lifetime that the algorithm can provide with a given set of UAVs.

Acknowledgements

This work has been carried out in the framework of the FUI project "AIRMES" (*Heterogeneous UAVs cooperating within a fleet*) and "IMATISSE" (*Inundation Monitoring and Alarm Technology in a System of Systems*) project, which is funded by the Region Picardie, France, through the European Regional Development Fund (ERDF).

References

- [1] AIRMES – Drones hétérogènes coopérant en flottille, <https://www.hds.utc.fr/heudiasyc/a-la-une/article/airmes-drones-heterogenes>.
- [2] Framework Libre AIR, <https://devel.hds.utc.fr/software/flair>.
- [3] MAVLink – Micro Air Vehicle communication protocol, <http://qgroundcontrol.org/mavlink>.
- [4] J.J. Acevedo, B.C. Arrue, I. Maza, and A. Ollero. Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions. *Journal of Intelligent & Robotic Systems*, 70(1):329–345, 2013.
- [5] J.W. Barnes, V.D. Wiley, J.T. Moore, and D.M. Ryer. Solving the aerial fleet refueling problem using group theoretic tabu search. *Mathematical and Computer Modelling*, 39(6):617 – 640, 2004.
- [6] I. Bekmezci, O.K. Sahingoz, and Ş. Temel. Flying ad-hoc networks (FANETs): A survey. *Ad Hoc Networks*, 11(3):1254–1270, 2013.
- [7] O. Burdakov, J. Kvarnström, and P. Doherty. Optimal scheduling for replacing perimeter guarding unmanned aerial vehicles. *Annals of Operations Research*, 249(1):163–174, 2017.

-
- [8] A. Bürkle, F. Segor, and M. Kollmann. Towards autonomous micro UAV swarms. *Journal of Intelligent & Robotic Systems*, 61(1):339–353, 2011.
- [9] O. Cetin and I. Zagli. Continuous airborne communication relay approach using unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 65(1):549–562, 2012.
- [10] T. Dietrich, O. Andryeyev, A. Zimmermann, and A. Mitschele-Thiel. Towards a unified decentralized swarm management and maintenance coordination based on MAVLink. In *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 124–129, May 2016.
- [11] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz. Help from the Sky: Leveraging UAVs for Disaster Management. *IEEE Pervasive Computing*, 16:24–32, Jan–Mar 2017.
- [12] M. Erdelj, M. Król, and E. Natalizio. Wireless Sensor Networks and Multi-UAV systems for natural disaster management. *Computer Networks*, 124:72–86, 2017.
- [13] Z. Jin, T. Shima, and C. J. Schumacher. Optimal scheduling for refueling multiple autonomous aerial vehicles. *IEEE Transactions on Robotics*, 22(4):682–693, Aug 2006.
- [14] N. Mathew, S. L. Smith, and S. L. Waslander. Multirobot rendezvous planning for recharging in persistent tasks. *IEEE Transactions on Robotics*, 31(1):128–142, Feb 2015.
- [15] J.R.T. Neto, A. Boukerche, R.S. Yokoyama, D.L. Guidoni, R.I. Meneguette, J. Ueyama, and L.A. Villas. Performance evaluation of unmanned aerial vehicles in automatic power meter readings. *Ad Hoc Networks*, 60:11 – 25, 2017.
- [16] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *Journal of Intelligent & Robotic Systems*, 84(1):469–492, 2016.
- [17] B. D. Song, J. Kim, J. Kim, H. Park, J. R. Morrison, and D. H. Shim. Persistent UAV service: An improved scheduling formulation and prototypes of system components. *Journal of Intelligent & Robotic Systems*, 74(1):221–232, Apr 2014.

-
- [18] N. K. Ure, G. Chowdhary, T. Toksoz, J. P. How, M. A. Vavrina, and J. Vian. An automated battery management system to enable persistent missions with multiple aerial vehicles. *IEEE/ASME Transactions on Mechatronics*, 20(1):275–286, Feb 2015.