Systemic Integrated Unmanned Aerial System

https://doi.org/10.3991/ijoe.v18i01.26435

Ahmed Refaat Ragab[1,2](✉), Pablo Flores Peña[3], Marco A. Luna[4],
Mohammad Sadeq Ale Isaac[4]
[1] October 6 University, Giza, Egypt
[2] University of Carlos III, Madrid, Spain
[3] Drone Hopper Company, Madrid, Spain
[4] Universidad Politécnica de Madrid, Madrid, Spain
a.refaat@drone-hopper.com

Abstract—Systemic integrated Unmanned Aerial System (UAS), is the process of gathering the subsystems into one fulfilled system. This integration is done in order to improve the system performance, reducing operational costs, and improving the time response of the system. Normally, such systems are integrated using different techniques such as communication processes, and computer networking. In this paper, a new integrated system is implemented by linking functionally computing systems and software applications together in one powerful system.

Keywords—systemic, UAS, computing systems, software applications

1 Introduction

Building a new innovative Unmanned Aerial System (UAS) depends upon many aspects such as; payload, Ground Control Station (GCS), sensors equipment, embedded systems, and electromechanical systems.

The purpose of building such a system is how to integrate all these subsystems into one powerful system such as UAS.

Through our work in Drone Hopper research center for building a new innovative UAS which is WILD HOPPER, the communication system and facilities gain our interest to design a powerful secured communication system for the WILD HOPPER, which is designed for critical missions (firefighting).

This paper gives a clear description on building and integrating the communication system, control system, and the software development of the flight pattern for WILD HOPPER. The paper is organized in five sections as follow; section two gives a brief description for the WILD HOPPER communication system, section three deals with the control architecture of the WILD HOPPER, section four cover the software of the flight patterns, and finally section five discuss the results and conclude the paper.

2 WILD HOPPER communication system

2.1 Proposed communication architecture

A communication architecture specifies how information flows between the ground Control station (GCS) and the WILD HOPPER, and during the mission itself if we are operating with more than UAVs, to be between the UAVs itself.

The proposed architecture will be performed into four communication architectures for networking WILD HOPPER.

1. Centralized communications

A centralized WILD HOPPER Communication architecture is to have the GCS to work as a central node, as shown in Figure 1, in which all the drones are connected to the GCS.

The centralized communication is the most common technique, in which each drone is directly connected to the GCS, where all the drones are transmitting and receiving information, control data, and commands, while the drones are itself not directly connected to each other.

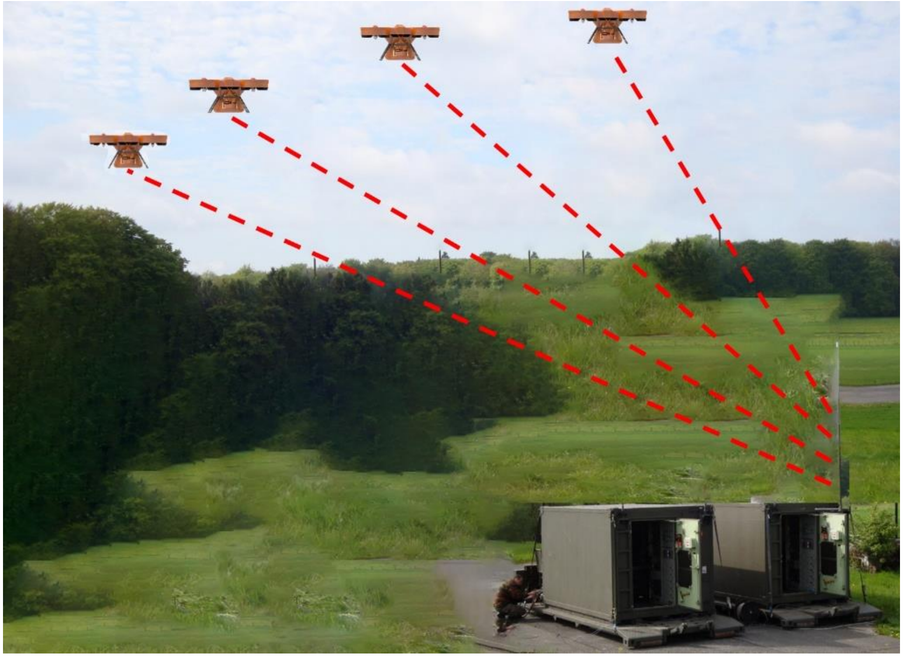


Fig. 1. Centralized communication

Through this communication technique, the overall network is centered by the GCS, and in order to have a communication between two or more drones this must be routed through the ground control station which is acting as a relay.

2. Decentralized communications

Through the decentralized architecture, two drones can communicate directly without the aid of the GCS, so an easy exchange of data and information between the drones itself exist instead of sending it to the GCS to be routed again.

The decentralized communication architecture is classified into three main categories as follow:

a) WILD HOPPER Ad-hoc network

Ad-hoc is a Latin word means "for this" or for this purpose, the most powerful issue into the ad-hoc network is that it doesn’t depend for the infrastructure of the network, where each node inside this network act as a router or access point by itself, and in our case here each drone will be a part of the data forwarding for the other drones.

Through Figure 2 it is cleared that a backbone WILD HOPPER serves as a gateway of the ad-hoc network formed from the other WILD HOPPERS Sending and receiving data between the ad-hoc network and the GCS.

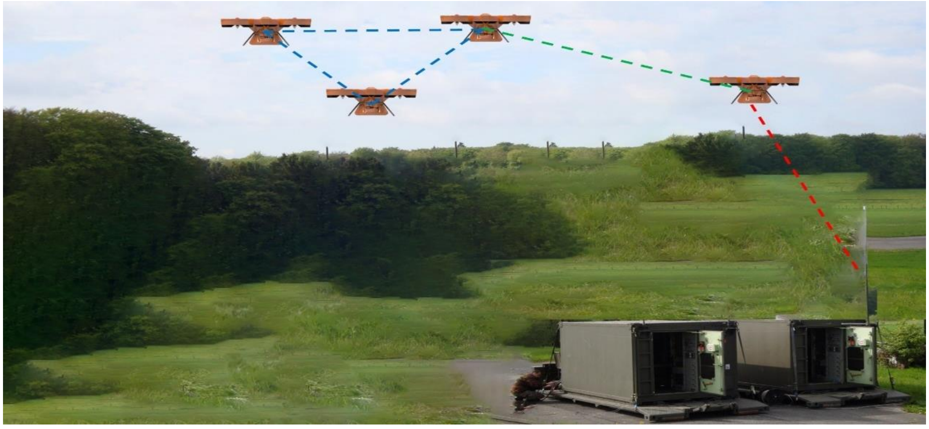


Fig. 2. WILD HOPPER Ad-hoc network

In this architecture the backbone WILD HOPPER is designed with two antennas, one to communicate with the other drones and the other to communicate with the GCS.

Through this mean of communication, the coverage area for WILD HOPPER mission could be extended due to the exist of only one drone to be directly connected to the GCS.

b) Multi-Group WILD HOPPER network

In this architecture, WILD HOPPERs are categorized into multi-group and each group has its WILD HOPPER backbone as shown in Figure 3, where each backbone have a connection to the GCS.

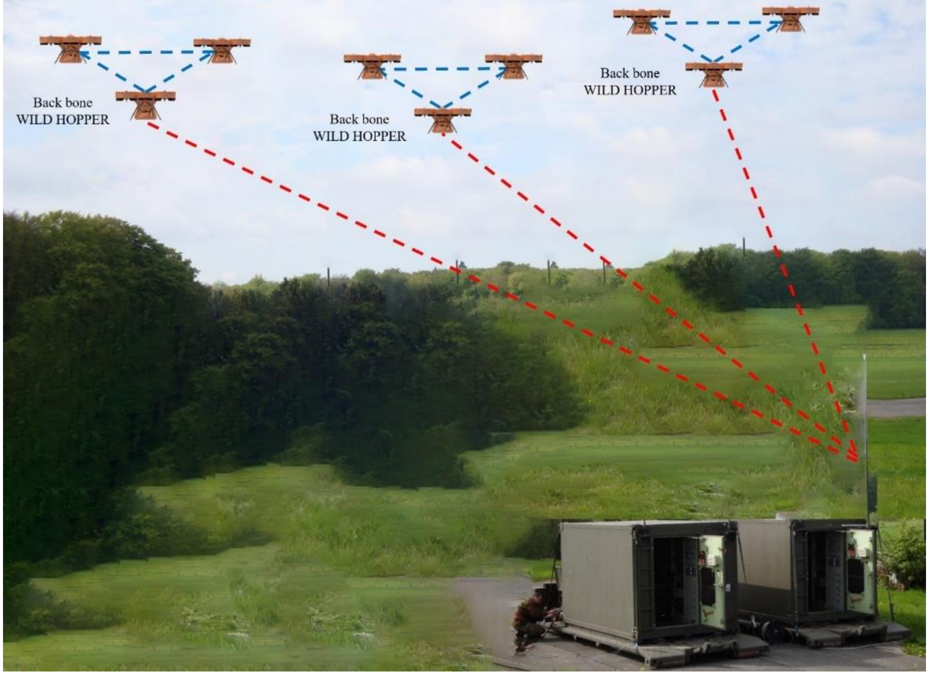


Fig. 3. Multi-Group WILD HOPPER network

Also, the mean of communication between the groups are by communicating through the backbone with the GCS, which is acting as a relay to route the data to its destination.

This architecture could be used in different missions for the WILD HOPPER at the same time (duration), where each group is dealing with a specific mission.

c) Multi-layer WILD HOPPER Ad-Hoc network

In this communication architecture the network is classified into groups, each group have a backbone as shown in Figure 4.

Through this architecture the communication with the GCS is done by one of the WILD HOPPERs backbone.

Also, there is a communication between the different groups by the mean of the backbones connected together. Thus, working with this architecture results in reducing the communication load and the computation in the GCS.

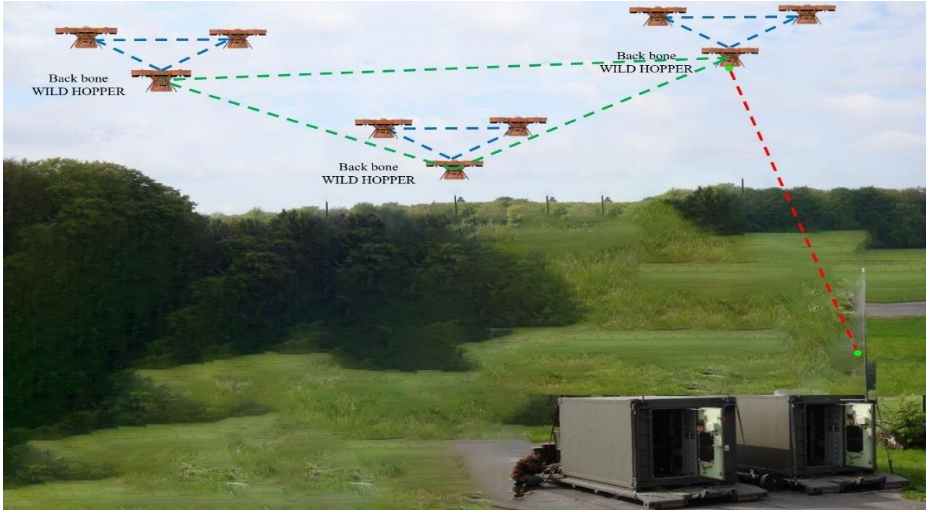


Fig. 4. Multi-layer WILD HOPPER Ad-Hoc network

Therefore, in the missions that's need a high response from the multi-group of WILD HOPPERS it is recommended to use this architecture. Table 1 shows a comparison between the different communication architectures, showing the advantages and disadvantages for each architecture.

Table 1. Communication architecture comparison

		Advantages	**Disadvantages**
1	Centralized communications	- The network is centered by the GCS.	- not robust - if any problem occurs to the GCS the overall network will be down. - Single point-of-failure - the drones are not directly connected to each other.
2	WILD HOPPER Ad-hoc network	- because multiple WILD HOPPERs are flying relatively close to each other, the transceiver device in the WILD HOPPER can be low-cost and lightweight. - this ad hoc network architecture is particularly appropriate for networking a group of similar WILD HOPPERs for operations such as persistent surveillance operations.	- the mobility patterns, such as speeds and heading directions, need to similar for all WILD HOPPERs.
3	Multi-Group WILD HOPPER network	- only one backbone is directly connected to the GCS	- Losing the backbone for any group leading on losing the all group.
4	Multi-Layer WILD HOPPER Ad-Hoc network	- Robust - Reduced computation and communication load in GCS. - information exchange between any two groups does not - need to be routed through GCS.	- Losing the backbone connected to the GCS impact in losing connection with the overall groups.

2.2 Communication system definition

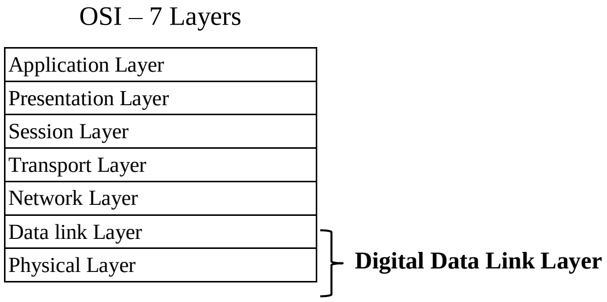
Communication systems represent the backbone for any unmanned aerial system, also it represents an important role for the service components and networking functionalities.

Through UAS the networking functionalities depends upon the communication between the UAVs and the GCS, and between the UAVs itself, this communication must be characterized by reliability, functionality and it must be secured.

Digging into telecommunications, the main responsible of connecting two entities to exchange data and information is the data link layer through the OSI - 7 layers with its devices, that are used to transmit and receive data and information, and this transmission and communication process are governed with the chosen protocol.

These devices must be defined with the operating frequency, data rate, transmission range, message format, security and channel protection, communication mode, and the power of the transceiver equipment.

Through our work we combine both data link layer and physical layer of the OSI – 7 layers to work as one layer called digital data link, in order to meet the communication requirements for the control and command into WILD HOPPER System, as shown in Figure 5.



OSI – 7 Layers

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data link Layer
Physical Layer

Digital Data Link Layer

Fig. 5. Digital Data Link representation through the OSI – 7 Layers

Thus, the communication process through the WILD HOPPER System will take place in the form shown in Figure 6.

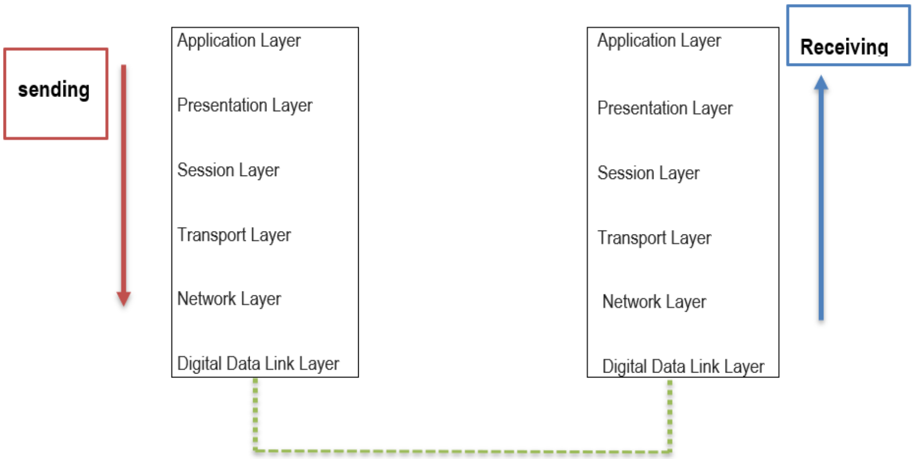


Fig. 6. Sending and receiving data using Digital Data Link Layer

The main challenge through designing any communication system is to design the transmitter and the receiver with respect of encoding and decoding, as the source encoder removes the redundancy from the source signal, resulting by a sequence of symbols (source codeword).

Through the channel encoder the data stream is processed, result in producing a channel codeword, and then a waveform is produced by the modulator (analog signal), which is transmitted to its destination over the wireless channel, and through the receiver a reverse form is applied to the signal, this is shown clearly through Figure 7, showing the pairing between the functional blocks of the transmitter and the receiver in the form of:

- Source encoder – decoder
- Channel encoder – decoder
- Modulator – Demodulator

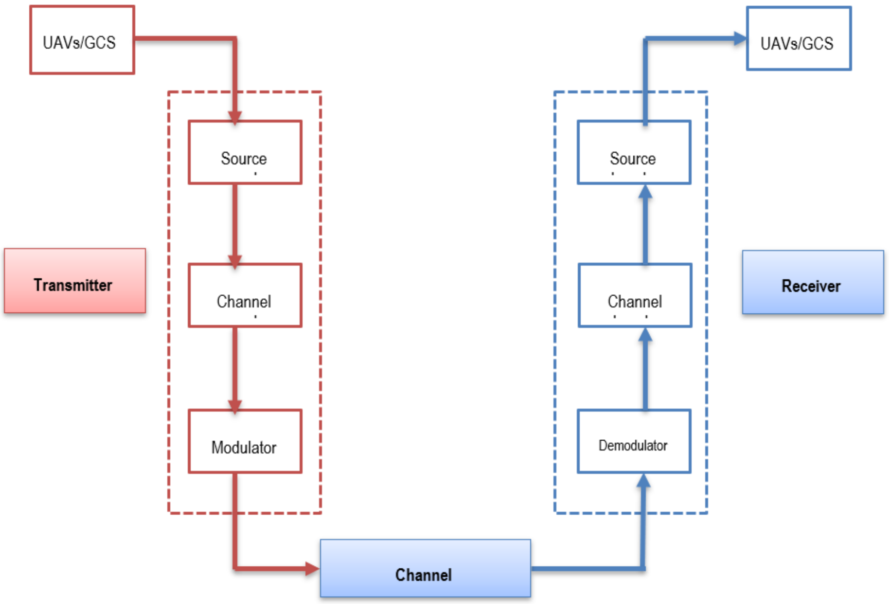


Fig. 7. Block diagram of the digital communication system

2.3 Communication system requirements

The main requirements through designing the WILD HOPPER Communication System are to have the following parameters:

1. Anti-spoofing capabilities
2. Anti-jamming capabilities
3. Defining all the standards with which the system is compliant.
4. A detailed diagram that shows the system architecture of the C2 link, including informational or data flows and the performance of the subsystem, and values for the data rates and latencies.
5. A description of the control link(s) connecting the WILD HOPPER to the GCS.
6. Defining the spectrum that will be used for the control link.
7. Defining the type of signal processing and/or link security (Encryption).
8. Defining the datalink margin in terms of the overall link bandwidth at the maximum anticipated distance from the GCS.
9. Defining the satellite links used.
10. Define the overall design system characteristics that prevent or mitigate the loss of the datalink due to the interference, flight beyond the communications range, antenna masking, loss of functionality for the GCS, loss of functionality of the WILD HOPPER itself, and the atmospheric attenuation.
11. Avoiding communication link degradation.
12. Avoiding communication link loss.
13. A detailed description of all the support equipment that is used on the GCS.

14. Powerful secured operating system.
15. A detailed description of the standard equipment available, and the backup or emergency equipment (UPS, etc).

2.4 Communication system functionalities

WILD HOPPER is an innovative system, with a communication network based scalable and distributed hardware architecture, and a service-based software architecture, this system can be defined as an abstraction communication layer, allowing the separation of concerns to facilitate interoperability and platform independence, as it was discussed before into the communication system definition.

Decision making system. WILD HOPPER is a system designed to have its own decision making as each drone is equipped by Jetson Xavier as shown in Figure 8, giving WILD HOPPER the facilities to act as a computer with computing and decision-making capabilities.

Fig. 8. Jetson Xavier

Dealing with jetson Xavier will give WILD HOPPER powerful specification represented in core i6 NVIDIA Carmel ARM®v8.2 64-bit CPU 6 MB L2 + 4 MB L3, vision accelerator with 7-Way VLIW Vision Processor, memory of 8 GB 128-bit LPDDR4x @ 51.2GB/s, video encoding capability of 2x 4K @ 30 | 6x 1080p @ 60 | 14x 1080p @ 30 (H.265/H.264), video decoding capability of 2x 4K @ 60 | 4x 4K @ 30 | 12x 1080p @ 60 | 32x 1080p @ 30 (H.265) or 2x 4K @ 30 | 6x 1080p @ 60 | 16x 1080p @ 30 (H.264), and Gigabit Ethernet, M.2 Key E (WiFi/BT included), M.2 Key M (NVMe) as a connectivity.

And to have an example upon this, we suppose a scenario for a huge forest fire occurred at X region, at first the fire starts through this region in an area of 100 km, as shown in Figure 9.

At the beginning of the fire, detailed images are captured through the satellite system and sent to the GCS.

Through this scenario we got nine WILD HOPPERs, classified into three swarms, dealing with multi-layer WILD HOPPER architecture as we discussed before, each swarm got its WILD HOPPER backbone.

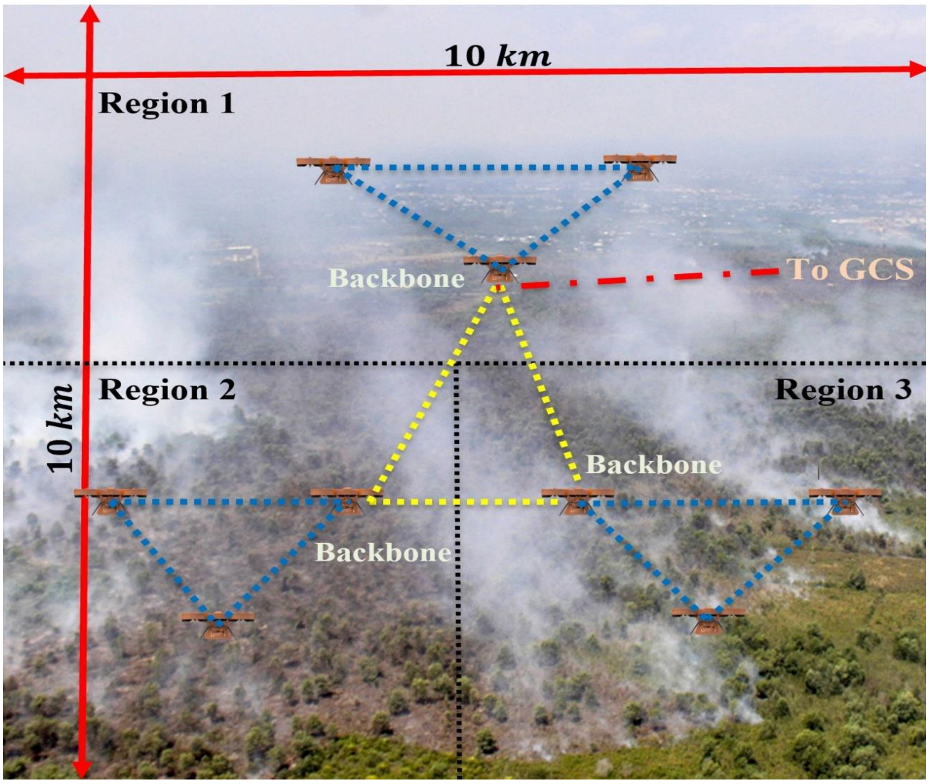


Fig. 9. WILD HOPPER forest fight scenario

The only decision that a human being will take is to classify these three swarms into the fire area to work in three regions, as shown in Figure 9.

Through the swarm mission the three swarms will begin to communicate with each other sending and receiving data for the action taking, and the total percentage of the fire upon the region they are working in.

Supposing that into the first region the fire extends, so the first swarm will send to the other two swarms asking for help, and through computing and decision-making capabilities the decision will be taken from one of the other two swarms to start helping the first swarm, all this is done automatically without the aid of the GCS or any human intervention.

3 WILD HOPPER control architecture definition

In this section we are giving a preliminary Definition of the control architecture that will be implemented in the WILDHOPPER, using the prototype of WILD HOPPER through our examination platform.

During the process of defining the control architecture, the following aspects have been considered:

3.1 Fixed pitch propellers

The prototype of WILDHOPPER does not have variable pitch propellers, the main rotors are used exclusively for the lift of the UAV and are not part of the Attitude control system. Note: The use of variable pitch propellers would significantly increase the efficiency and controllability of the aircraft.

3.2 Micropilot firmware

By using the Xtender it is possible to integrate the Micropilot MP2128HELI2 autopilot in any type of UAV, regardless of the type of control or actuators present.

The test was carried out by an integration specifically designed for a gasoline multirotor with Attitude control through Flaps and EDFs.

3.3 Redundancy of control

The WILDHOPPER is a special UAV that has redundancy in its attitude control systems through control by EDFs and by Flaps. In the event of the loss of one of the systems, the other has sufficient capacity to maintain stability in flight.

3.4 Model simulation

All the models proposed in this paper have been tested in Matlab Simulink + Simscape and all cases, it is shown that the control of this prototype using the proposed systems (Flaps and EDFs) can be enhanced to fly at high speeds or in strong gusts of wind, the experimental test was based upon three models respectively attitude control using EDFs, attitude control using flaps, and mixed attitude control EDFs + Flaps.

Model 1: Attitude control using EDFs. The attitude control in this model is carried out exclusively using the EDFs installed in the UAV (R1, R2, R3, and R4), while the lift of the UAV is produced by the main rotors (R5) as shown in Figure 10.

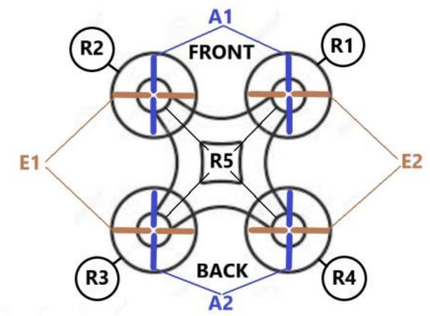


Fig. 10.WILD HOPPER prototype EDF control definition schematic

The implementation of the control by EDFs in Horizon is carried out with the pre-established model 34-Custom Multirotor Mixing, which consists of a multirotor with 5 rotors in which the contribution of each of the rotors to the flight phases is fully configurable. Following this approach, in this model, each of the EDFs constitutes one of the channels assigned to the multirotor while the gas control is assigned as the fifth rotor with its channel (R5), as shown in Table 2, and through Figure 11 which implement the EDFs control architecture.

Table 2. Contribution of model 1 motors (EDFs)

	ROTOR 1 (R1)	**ROTOR 2 (R2)**	**ROTOR 3 (R3)**	**ROTOR 4 (R4)**	**ROTOR 5 (R5)**
AILERON	1	-1	-1	1	0
ELEVATOR	-1	-1	1	1	0
YAW	1	-1	1	-1	0
THROTTLE	0	0	0	0	1

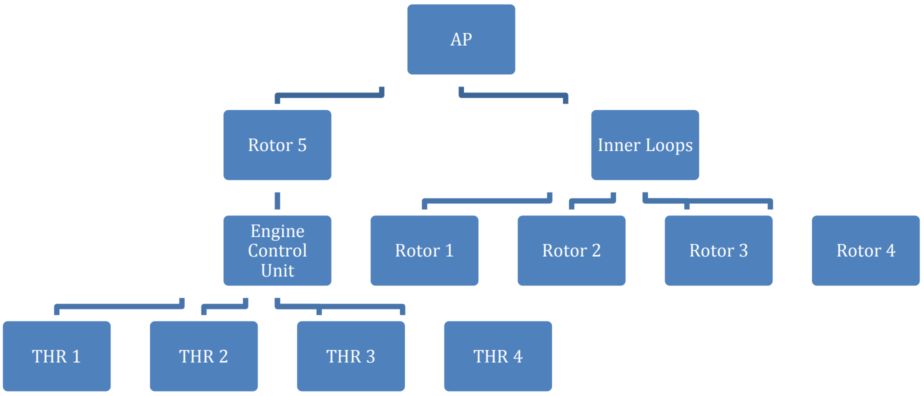
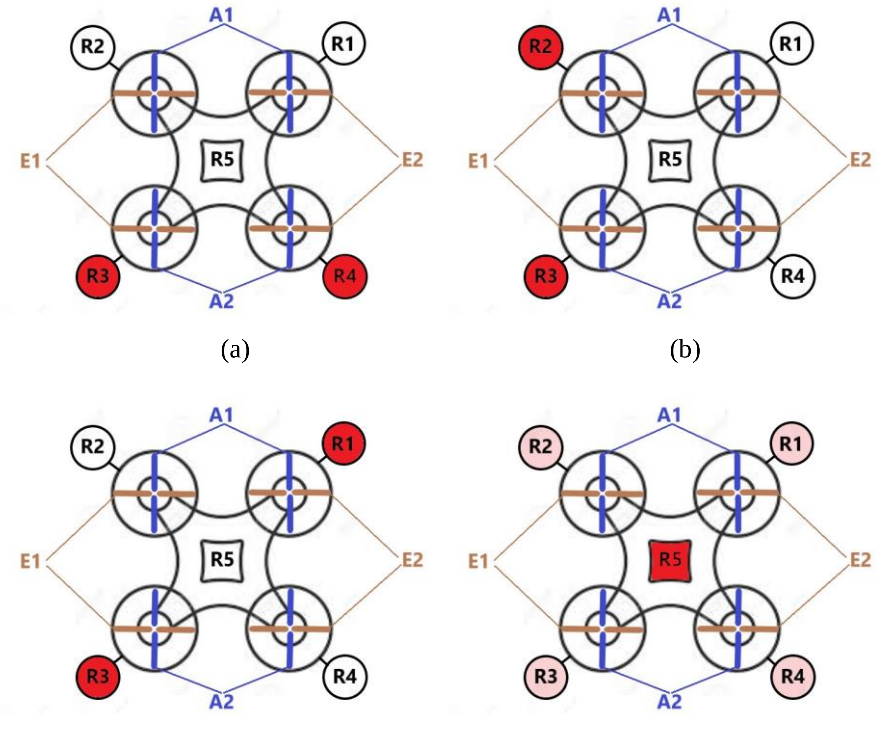


Fig. 11.WILD Hopper prototype EDFs control architecture

Also, through this model an examination was taken place for the surface control using EDFs, to control pitch down, roll right, yaw left, and throttle up as shown in Figure 12.



(a)

(b)

(c)

(d)

Fig. 12.EDFs control. a) Pitch Down. b) Roll right. c) Yaw left. d) Throttle up

Through our experimental results and conclusion from Model 1, six main important aspects were considered for their main importance as shown in Table 3.

Table 3. Experimental suggestions related to control by EDFs

Description	Recommendation	Importance
Variable pitch rotors	The use of variable pitch rotors is recommended. Its use would increase the attitude control and the efficiency of the system.	HIGH
EDFs size	The use of EDFs with higher thrust capacity is recommended. Although the EDFs only contribute to the control of Attitude and not to the lift of the UAV, the current EDFs have a thrust of 100N, and the simulations show that the control of Attitude with EDFs of these characteristics is very slow of reactions and could reach to be insufficient in certain flight circumstances.	HIGH
Onboard generator	A generator is required on the UAV board to power the system batteries.	HIGH
Angle of	It is essential to install theE DFs with an angle of inclination of at	HIGH

inclination of EDFs	least 5 degrees towards the outside.	
Relosution of EDFs	Although the control of the EDFs is only used in the control of attitude, the effective thrust they provide begins in a range greater than 50% of its travel, which makes it necessary for the EDFs to always remain active in an operating range.	Medium
EDFs Position	If necessary, a greater distance of the EDFs with respect to the center of the UAV would give a greater entity to the Attitude control of the EDFs.	Medium

Model 2: Attitude control using FLAPS. The attitude control in this model is carried out exclusively through the FLAPS installed in the UAV (A1, A2, E1, and E2), while the lift of the UAV is produced by the main rotors (R5), as shown in Figure 13, and the Flaps are grouped into four main groups as follow:

1. A1: Roll Flaps located on the front rotors of the UAV.
2. A2: Roll Flaps located on the rear rotors of the UAV.
3. E1: Pitch flaps located on the right-side rotors of the UAV.
4. E2: Pitch flaps located on the left-side rotors of the UAV.

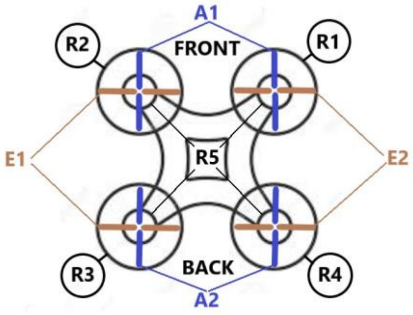


Fig. 13.WILD Hopper prototype FLAPS control definition schematic

The implementation of the FLAPS control in Horizon is carried out with the pre-established model 34-Custom Multirotor Mixing, which consists of a 5-rotor multirotor in which the contribution of each of the rotors to the flight phases is fully configurable. Following this approach, in this model, each of the FLAPS groups (A1, A2, E1, and E2) constitutes one of the channels assigned to the multirotor while the gas control is assigned as the fifth rotor with its channel (R5), as shown in Table 4, and Figure 14.

Table 4. Contribution of model 2 motors (FLAPS)

	ROTOR 1 (A1)	**ROTOR 2 (A2)**	**ROTOR 3 (E1)**	**ROTOR 4 (E2)**	**ROTOR 5 (R5)**
AILERON	1	1	0	0	0
ELEVATOR	0	0	1	1	0
YAW	-1	1	1	-1	0
THROTTLE	0	0	0	0	1

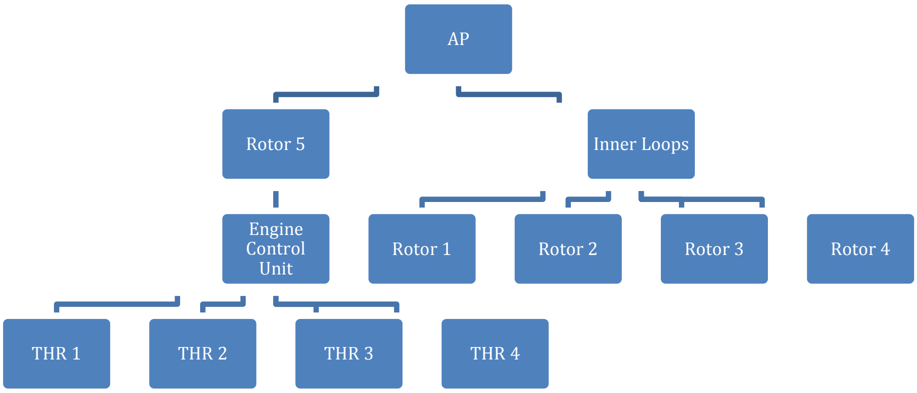
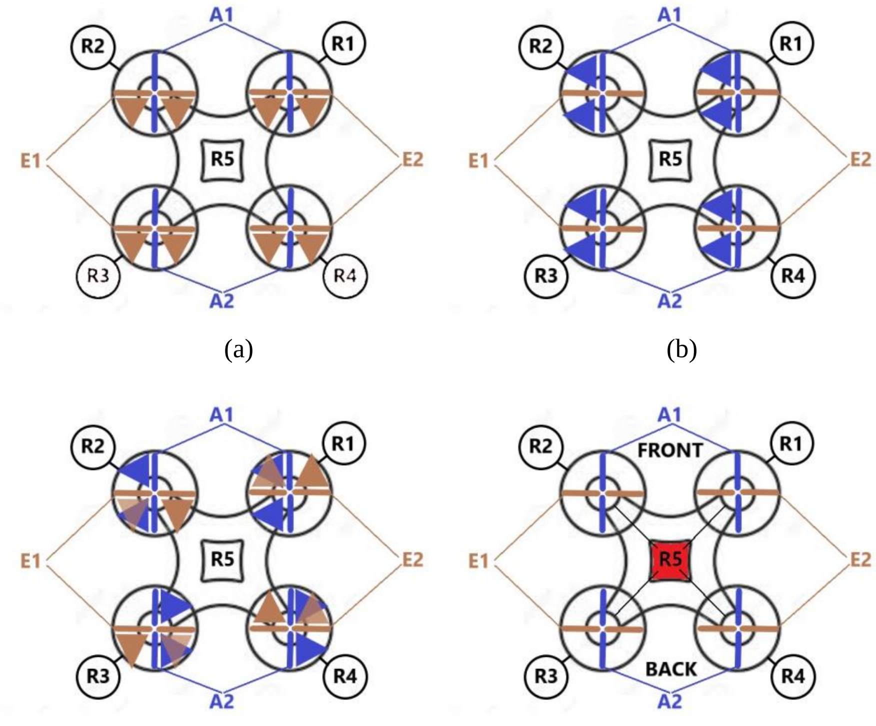


Fig. 14.WILD Hopper FLAPS control architecture

Besides, through this model an examination was taken place for the surface control using FLAPS, to control pitch down, roll right, yaw right, and throttle up as shown in Figure 15.



(a)

(b)

(c)

(d)

Fig. 15.FLAPS control. a) Pitch Down. b) Roll right. c) Yaw left. d) Throttle up

Also, through our experimental results and conclusion from Model 2, four main important aspects were taken into account for their main importance as shown in Table 5.

Table 5. Experimental suggestions related to control by FLAPS

Description	Recommendation	Importance
Variable pitch rotors	The use of variable pitch rotors is recommended. Its use would increase attitude control and the efficiency of the system.	High
Slow response in simulation	The results of the tests carried out in simulation with a model with characteristics similar to the WILDHOPPER result in a slow response in the control of the UAV.	High
Stall in tail turns	In tail turns, a drop in height can occur caused by the loss of thrust caused by the action of the flaps simultaneously. The Height Control should be able to compensate for this effect.	Medium
Reduction of the number of flaps	The number of flaps has been reduced to 16 (initially there were 32).	High

Model 3: Mixed attitude control EDFs + FLAPS. The attitude control in this model is carried out in a mixed way using the EDFs installed in the UAV (R1, R2, R3, and R4) and the FLAPS (A1, A2, E1, and E2) while the lift of the UAV is produced by the main rotors (R5), as shown in Figure 16. In this model, it is possible to deactivate the control of one of the two systems (EDFs or FLAPS) at any time while the other maintains control of Attitude.

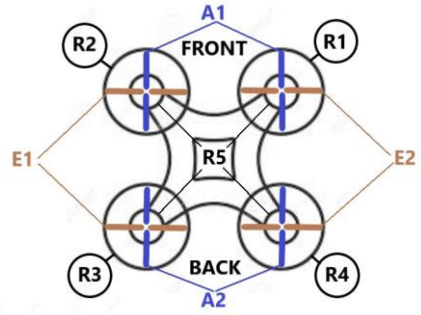


Fig. 16.WILD Hopper prototype EDFs + Flaps control definition schematic

The implementation of control by EDFs in Horizon is carried out with the pre-established model 34-Custom Multirotor Mixing, which consists of a multirotor with 9 motors/rotors in which the contribution of each of the motors to the flight phases is fully configurable. Following this approach, in this model, each of the EDFs is identified with engines 1, 2, 3, and 4 of the model selected in the autopilot, while the FLAPS are assigned to engines 5, 6, 7, and 8. The throttle control is done by assigning the Throttle channel (R5) to Engine 9, as shown in Table 6, and Figure 17.

Table 6. Contribution motors model 3 (EDFs + FLAPS)

	R1	R2	R3	R4	A1	A2	E1	E2	R5
AILERON	1	0	0	1	1	1	0	0	0
ELEVATOR	-1	-1	1	1	0	0	-1	-1	0
YAW	-1	0	-1	0	-1	1	1	-1	0
THROTTLE	0	0	0	0	0	0	0	0	1

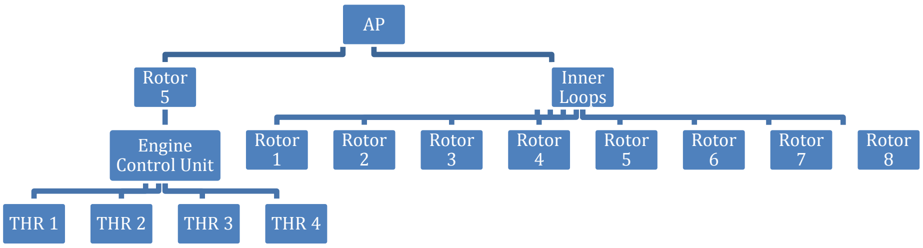
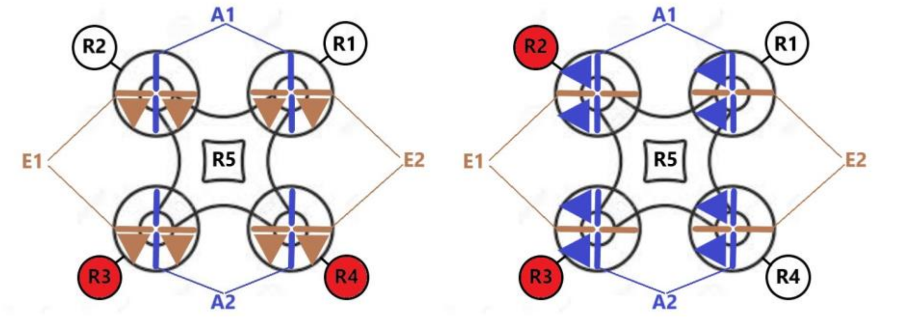
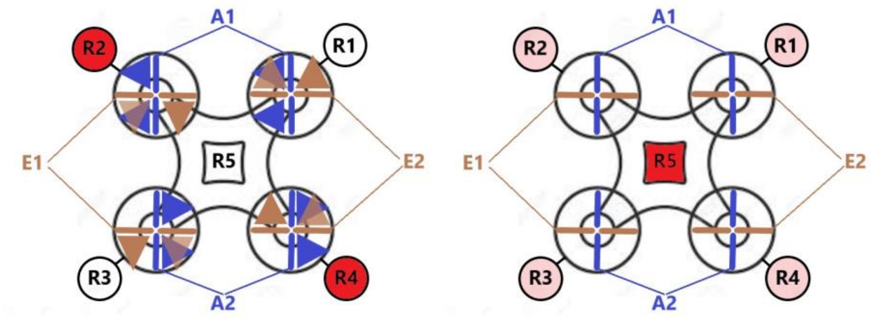


Fig. 17.WILD Hopper EDFs + FLAPS control architecture

Also, through this model an examination was taken place for the surface control using EDFs + FLAPS, to control pitch down, roll right, yaw right, and throttle up as shown in Figure 18.



(a) (b)



(c) (d)

Fig. 18.EDFs + FLAPS control. a) Pitch Down. b) Roll right. c) Yaw left. d) Throttle up

Finally, from all the above-mentioned results and analysis, which is based on the simulations carried out, it is recommended to use model 1 or model 3 as the WILD Hopper control architecture.

Also, the recommendations made in Tables 3 and 5 must be followed to complete the integration successfully.

4 WILD-HOPPER software for fast response flight patterns

This section describes the software for the fast response flight pattern performance. It consists of an application developed in C++ complementary to the main GCS program that uses Micropilot MP2 x 28g2 flight commands such as: "fly to"; "hover at"; "circle left"; "circle right"; and others; however; the same methods could be implemented in different platforms.

In this application, the operator chooses a point in the GCS map, sends to the autopilot the command "Go Here"; and sets the pattern features as radius, dimensions, the width of streets, and the direction of rotation. Finally, the UAV executes the mission taking the selected point as the central point.

4.1 Flight patterns description

Next, each flight pattern is described. The results of the performance are presented in the Results section.

Orbit and Funnel flight patterns. In both patterns, the aircraft performs a circular trajectory flight and, the operator can customize de radio and the direction of rotation. While in the “Orbit” pattern, the UAV flies with a trajectory tangential heading in the “Funnel” pattern it points to the center of the circle. The program reads the required parameters (radius, direction) and passes them directly on the circle flight command. Only the heading change depending on the pattern.

Racetrack flight pattern. Flight in rectangle (four waypoints). The racetrack dimensions, rotation angle, and direction of flight are set by the operator. The program reads the parameters and places a set of waypoints depending on the direction of navigation (left or right), as presented in the Figure 19.

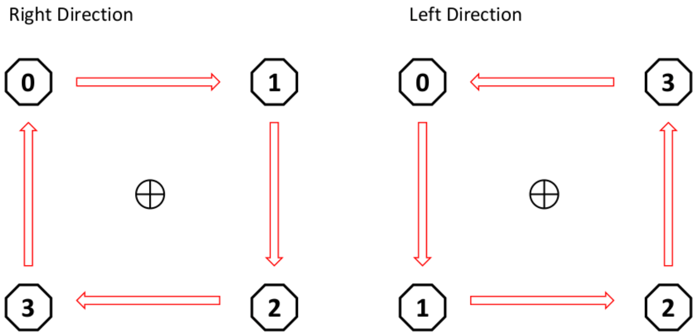


Fig. 19. Location of waypoints for racetrack pattern

All the operations use relative coordinates. The programming logic is described in Algorithm 1.

```
  Algorithm 1: Racetrack flight pattern
  Data: PrevtargetWpt, X_length, Y_length,Angle, PrevDirection,CurrentDirection
  if CurrentDirection=Right then
      x_coords [4]= {-X_length/2,-X_length/2,X_length/2,X_length/2}
    y_coords [4]= {Y_length/2,-Y_length/2,-Y_length/2,Y_length/2}
else
      x_coords [4]= {X_length/2,-X_length/2,-X_length/2,X_length/2}
    y_coords [4]= {Y_length/2,Y_length/2,-Y_length/2,-Y_length/2}
end
  if CurrentDirection=PrevDirection then
      current_target_waypoint= prev_target_waypoint
  else
      if current_target_waypoint= 0 then
          next_target_waypoint= 1
      end
      if current_target_waypoint= 1 then
          next_target_waypoint= 0
      end
      if current_target_waypoint= 2 then
          next_target_waypoint= 3
      end
      if current_target_waypoint= 3 then
```

```
          next_target_waypoint= 2
      end
  end
```

In Algorithm 1, each time that operator runs the pattern or changes the parameters, the "*previous direction of navigation*" and the "*previous target waypoint*" are evaluated to avoid jumps between waypoints. For instance, if the aircraft is going to a waypoint with the left direction and receives the order to change its flight to the right, it should not restart the entire racetrack; but continue flying with the new parameters from the nearest waypoint. In this context, the "target waypoints" do not change if the direction does not change; in the other cases, Figure 20 shows the equivalent waypoints. Trigonometric operations solve the angle parameter.

Search flight pattern. In “Search” mode, the UAV performs a spiral flight; the operator configures the radio of search and the width of streets. For these, the software commands multiple circular flights with different radius. Algorithm 2 describes the calculation process.

```
  Algorithm 2: Search flight pattern
  Data:   SearchRadius, StreetWidth
  CircleNumber="ROUND" (2×SearchRadius/StreetWidth)
  for i=0 to CircleNumber do
      i^th  CircleRadius=(i+2)/2* StreetWidth
      if  i mod 2=0
          i^th  CircleCenter(x,y)= (0,0) (Relative coords)
          FinishCircleParam=180 deg
      end
      else
          i^th  CircleCenter(x,y)= (-StreetWidth/2,0)
(Relative coords)
          FinishCircleParam=0 deg
      end
  end
```

Algorithm 2 proposes a two-center-circles-based spiral; the total number of circular trajectories is calculated by dividing the “Search Radius” for the “Street Width” and multiplying by 2. The radii are calculated with the formula: $\frac{i+2}{2} * StreetWidth$. The algorithm places the centers for even circles in the origin of the flight pattern. On the other hand, for odd, the center is displaced by $\frac{StreetWidth}{2}$ in the x edge. Finally, the finish circle param refers to the point (in degrees) of the circumference in which each trajectory ends after complete at least a half lap.

Grid flight pattern. Flight through streets covering a rectangular area. The operator configures the dimensions and rotation angle in the rectangle; and the street width in the grid. It is presented in Algorithm 3.

```
  Algorithm 3: Grid flight pattern
  Data:   Xlength, Ylength, Angle, StreetWidth
  StreetNumber="ROUND" (Ylenght/StreetWidth)
  for i=0 to StreetWidht   do
      if  i mod 2=0
          Wpt_A (x,y)=((-Xlength)/2,((i*StreetWidth)-
Ylenght/2  ))
          Wpt_B (x,y)=(Xlength/2,((i*StreetWidth)-
Ylenght/2  ))
      end
      else
          Wpt_A (x,y)=(Xlength/2,((i*StreetWidth)-
Ylenght/2  ))
          Wpt_B (x,y)=((-Xlength)/2,((i*StreetWidth)-
Ylenght/2  ))
      end
  end
```

In Algorithm 3, two waypoints are placed in each street, centered on (0,0) in relative coordinates. Depending on the street number, the algorithm locates the points in the left-right direction or vice versa to perform a continuous flight.

5 Results and conclusion

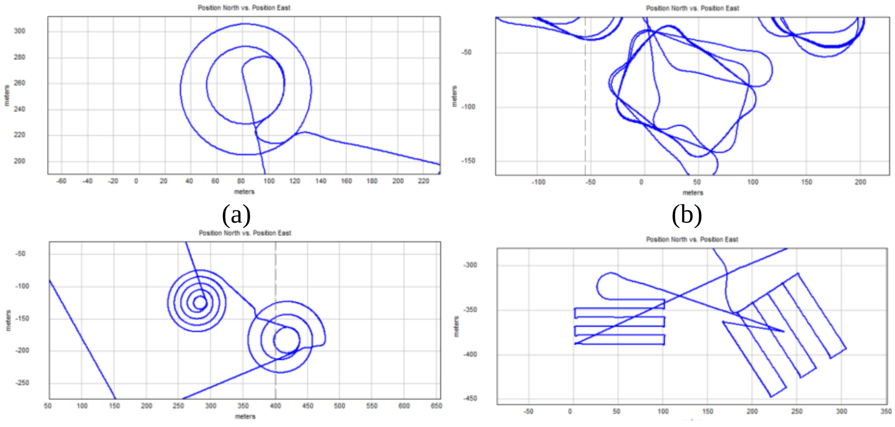
Through designing the WILD HOPPER communication system, the communication architecture was described clearly, showing the advantages and disadvantages of each proposed architecture for the system as shown in Table.1.

Also, the communication system was defined shown the combination done for both the data link layer and the physical layer of the OSI-7 Layers to become one digital layer called digital data link layer, as it was shown in the Figure.5, this new layer covers the communication requirements for the control and command of the system.

On other hand, the communication system requirements were introduced, and the importance of the WILD HOPPER communication system appears at its role as an innovative decision-making system by adding Jetson Xavier, which allows the WILD HOPPER to act as a powerful decision-making system.

In addition, a clear definition of the control architecture was introduced, showing the power of adding both Flaps and EDFs to the system, to maintain the stability of the system during the flight.

While, the Graphic results for the flight patterns are presented in Figure 20, representing the trajectories generated in a simulated flight. As it can be seen, the parameters of each flight pattern were customized in-flight; the results show that desired missions (flight patterns) can be planned and performed efficiently. In the "Orbit" flight pattern (Figure 20 a), the operator performs a change of parameters in-flight (radius and direction of rotation); the figure shows a soft trajectory change. Also, Figure 20 b, presents the "Racetrack" flight pattern; in this case, the algorithm places the waypoints correctly, but the trajectory control depends on airspeed (the distance that autopilot considers a waypoint reached increases when airspeed increases); for this reason, routes are not rectangular. Figure 20 c, shows spiral flights corresponding to the "Search" pattern; circle paths are correctly stitched, and changes between them are soft. Finally, Figure 20 d, presents the "Grid" pattern; the algorithm calculates and places the waypoints accurately, even with rotation angles.



(a) (b)

(c) (d)

Fig. 20. Flight patterns performance. a) Orbit/Funnel flight pattern. b) Racetrack flight pattern. c) Search flight pattern. d) Grid flight pattern

In conclusion, as results show, this application is functional for a GCS operator in emergencies because of its ease and speed of execution of in-flight patterns; multiple GCS software requires 5 minutes or more to plan these types of missions and cannot upload them in flight. Also, generated trajectories are soft and safe for the aircraft.

6 Data and software availability

All the data is available through Drone Hopper Data Center 1, Data Center 2, and the software tool used is MATLAB available at WILD-HOPPER DATA.

Also, we would like to inform that WILD HOPPER is patented through the OFICINA ESPAÑOLA DE PATENTES Y MARCAS, ESPAÑA.

7 Acknowledgment

We would like to appreciate the European Commission for their supporting and funding of the WILD-HOPPER project, and also with great thanks to the University of Carlos III, campus Leganes, Madrid for performing laboratories and hangers through our research; and last but not least, to our colleagues at Drone Hopper for their supports during the project. This work has been Funded by “The European Commission, H2020” REF: Grant agreement ID: 880642.

8 References

[1] Georg Winkel, Filip Aggestam, Metodi Sotirov and Gerhard Weiss "Forest Policy in the European Union" chapter 3, European Forest Governance:Issues at Stake and the Way Forward, 2013. pg 52-63.

[2] D. Alexandrian et. al. "Forest fires in the Mediterranean area", Unasylva No. 197, http://www.fao.org/3/x1880e/x1880e07.htm

[3] Jesús San-Miguel-Ayanzet al. "Comprehensive Monitoring of Wildfires in Europe: The European Forest Fire Information System (EFFIS)", European Commission, Joint Research Centre Italy, https://ec.europa.eu/environment/forests/pdf/InTech.pdf

[4] Joaquim Vasconcelos Reinolds de Sousa and Pedro Vieira Gamboa " Aerial Forest Fire Detection and Monitoring Using a Small UAV " International Congress on Engineering — Engineering for Evolution,KnE Engineering, pages 242–256, Volume 2020. https://doi.org/10.18502/keg.v5i6.7038

[5] Amit Sharma, Pradeep Kumar Singh, Ashutosh Sharma and Rajiv Kumar "An efficient architecture for the accurate detection and monitoring of an event through the sky" Computer Communications September 2019. https://doi.org/10.1016/j.comcom.2019.09.009

[6] Omar A. Saraereh, Amer Alsaraira, Imran Khan and Peerapong Uthansakul "Performance Evaluation of UAV-Enabled LoRa Networks for Disaster Management Applications" Article in Sensors April 2020. https://doi.org/10.3390/s20082396

[7] Sezgin Kilic and Omer Ozkan " A self-adaptive UAV routing for forest fire risk mitigation: a conceptual model" Modeling & Simulation International (SCS), SummerSim-SCSC, 2019 July 22-24, Berlin, Germany.

[8] Akhloufi, M.A.; Couturier, A.; Castro, N.A. "Unmanned Aerial Vehicles for Wildland Fires: Sensing, Perception, Cooperation and Assistance". Drones. 2021; 5(1):15; pp 1-25. https://doi.org/10.3390/drones5010015

[9] Vladimir Sherstjuk, Maryna Zharikova, and Igor Sokol "Forest Fire-Fighting Monitoring System Based on UAV team and Remote Sensing" 2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO), 20 September 2018. https://doi.org/10.1109/elnano.2018.8477527

[10] Elena Ausonio, Patrizia Bagnerini, and Marco Ghio " Drone Swarms in Fire Suppression Activities: A Conceptual Framework" Drones 2021, 5, 17. https://doi.org/10.3390/drones5010017

[11] https://www.easa.europa.eu/document-library/regulations/regulation-eu-20181139

[12] Ahmed Refaat Ragab, et.al, "WILD HOPPER Prototype for Forest Firefighting," International Journal of Online and Biomedical Engineering, September 2021.

[13] Ahmed Refaat Ragab, "A New Classification for Ad-Hoc Network," International Journal of Interactive Mobile Technologies, Vol. 14, No. 14, 2020.

9 Authors

Ahmed Refaat Ragab is a postdoctoral researcher at the University of Carlos III, and Drone Hopper Company, Leganes, Madrid, Spain. He was graduated from Air Defense College, then got a diploma in computer engineering from the National telecommunication institute in Egypt, master degree in computer engineering from the college of engineering in the Arab Academy for Science and Technology and Maritime Transport, with an excellent degree, and the thesis title was "Secure Routing In

UAV". He obtained the Degree of Doctor of Philosophy in Electrical Engineering, from the faculty of engineering, Benha University, with an excellent degree, and the thesis title was" UAV Cloud Computing System".

Pablo Flores Peña Ph.D. in Aeronautical engineer, he is the CEO & CTO for Drone Hopper Company, he creates the research center at Drone Hopper Company. Also, he is an Aeronautical engineer & IE MBA. He got more than 15 years of experience as chief engineer at Airbus Company.

Marco A. Luna is a Ph.D. student at the Polytechnic University of Madrid, and a software engineer at Drone Hopper Company, Leganes, Madrid, Spain. He obtained his BSc in automation and control engineering from Universidad de las Fuerzas Armadas-ESPE (Ecuador) and his master's in automatic and robotics from the Polytechnic University of Madrid (Spain), and the thesis title was "3D location system for aircraft inspection by UAVs".

Mohammad Sadeq Ale Isaac is a Ph.D. student at the Polytechnic University of Madrid I, and a control engineer at Drone Hopper Company, Leganes, Madrid, Spain. He was graduated from the Khaje Nasir University of Technology, then obtained his master's in aerospace engineering, specializing in flight dynamics and control from the Polytechnic University of Tehran (Amir Kabir), and the thesis title was "Simulation and Implementation a Controlled Landing of a Quad-copter on a Moving Platform.

Article submitted 2021-08-23. Resubmitted 2021-10-03. Final acceptance 2021-10-08. Final version published as submitted by the authors.