Toward Fault-Tolerant Coordination in UAV Swarms for Offshore Wind Farm Inspection

Jiayi Chen*, Maha Riad†, Saeedeh Ghanadbashi‡,
Guthemberg Silvestre*, Abdollah Malekjafarian‡, Fatemeh Golpayegani†
*Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, France
Email: jiayi.chen@recherche.enac.fr, silvestre@enac.fr
†School of Computer Science, University College Dublin, Ireland
Email: maha.riad@ucdconnect.ie, fatemeh.golpayegani@ucd.ie
‡School of Civil Engineering, University College Dublin, Ireland
Email: saeedeh.ghanadbashi@ucdconnect.ie, abdollah.malekjafarian@ucd.ie

***Abstract*—Effective inspection of offshore wind farms is essential to maintain their operational reliability. UAVs provide a promising means to reduce inspection costs and duration while minimizing risks to human operators, making them valuable tools for infrastructure health monitoring. When organized as swarms, UAVs can exploit collective intelligence and coordination to improve inspection coverage and efficiency. However, offshore environments impose unique challenges, including limited accessibility and harsh weather conditions, which require the swarm to operate with high fault tolerance. In this work, we introduce FOCUS, a novel mission-level approach for offshore wind farm inspection, that leverages mobile ad hoc connectivity and distributed consensus within a UAV swarm to enable fault-tolerant operations. Emulation-based evaluations show that FOCUS doubles the inspection efficiency compared with baseline approaches while maintaining strong resilience against common failures.**

***Index Terms*—Fault-tolerance, distributed consensus, UAV swarm coordination, offshore wind farm inspection**

I. Introduction

Aligned with the EU's offshore renewable goals of achieving around 111 GW by 2030 and 317 GW by 2050, offshore wind energy is set to become a main pillar of Europe's future electricity mix [1]. However, offshore wind turbines (OWTs) endure harsh environmental conditions, where dynamic and extreme loads affect their safety and service life. According to recent surveys [2], [3], operations and maintenance (O&M) costs represent a large share of the global expenses of an OWT project, surpassing 30% of the total levelized cost of energy. Thus, reducing O&M cost is key to increasing the competitiveness of offshore wind energy in the market of renewable energy.

Traditional manual inspections for wind turbines O&M are often labor-intensive, risky, subjective, and costly [4], especially in offshore environments, where severe weather increases health and safety risks for human inspectors. To deal with these issues, remote monitoring systems, particularly unmanned aerial vehicles (UAVs), have emerged as a leading solution to reduce operational costs and enhance inspector safety, making them increasingly routine for such infrastructure inspection and maintenance. Equipped with optical cameras, infrared cameras, or X-ray scanners, UAVs are able to perform inspection of OWT by leveraging non-destructive testing techniques such as optical photography, thermography, and computer vision technologies [5].

As offshore wind farms continue to expand, the use of UAV swarms is increasingly relevant. While many emerging remote monitoring systems still rely on a single UAV for simplicity [6], [7], their limited endurance and vulnerability to harsh marine conditions often hinder continuous operation and expose the system to higher risk of failure. In contrast, UAV swarms enable more time-efficient and resilient inspections by carrying out concurrent, distributed actions [8].

Recent studies on multi-UAV and UAV swarm systems for infrastructure inspection (e.g., wind turbines, power grids, bridges) have primarily focused on path planning, task allocation, and decision-making algorithms [9]–[11], while offering limited attention to swarm-level coordination. Moreover, most existing approaches depend on centralized, ground-based control architectures [12], which restrict scalability, fault tolerance, and autonomy. To address these challenges, this work aims to leverage distributed computing, specifically distributed consensus, within UAV swarms to enable fault-tolerant coordination for such inspection missions.

Distributed consensus protocols are widely used in practical systems to ensure reliable coordination under failures and concurrency [13]. They enable a group of processes to agree on the execution order of operations, a property known as linearizability. These protocols implement fault-tolerant algorithms that provide safety and liveness guarantees based on specific assumptions about timing and failure models. As detailed by Ongaro and Ousterhout [14], such protocols enable strongly consistent data coordination, which in turn greatly simplifies the design of distributed services and configuration management in UAV swarms.

Building on this foundation, we propose FOCUS—a Fault-tolerant, Consensus-driven UAV-Swarm-based approach for offshore wind farm inspection. By integrating a distributed consensus protocol into mobile UAV swarms, FOCUS enables decentralized swarm configuration and strongly consistent inspection data replication, ensuring mission completion even in the presence of UAV failures. In small-sized swarms, data remains fault-tolerant as long as a quorum of UAVs is available, while swarm-wide replication supports autonomous workload sharing, thereby improving efficiency, resilience, and autonomy in the challenging offshore environment. Each UAV operates with consistent, up-to-date global information, further enhancing system availability in dynamic and uncertain conditions.

Additionally, to address the degraded performance of distributed consensus algorithms in mobile ad-hoc networks (MANETs), where frequent network partitions lead to data unavailability [15], FOCUS incorporates a temporary local storage mechanism. Experimental results show that combining distributed consensus with this mechanism significantly improves adaptability to dynamic network conditions.

In summary, the contributions of this work are:

- We present FOCUS, a fault-tolerant, mission-level swarm coordination approach that enables a novel application of distributed consensus protocols in mobile UAV swarms, and demonstrate its deployment in a real-world scenario—offshore wind farm inspection. The approach also incorporates a temporary local storage mechanism to address the low availability of consensus protocols in dynamic networks.
- Through extensive evaluations, we assess the feasibility and capabilities of the proposed approach. Our results show that FOCUS significantly reduces mission completion time compared to non-collaborative approaches lacking swarm coordination. Moreover, FOCUS exhibits strong fault tolerance under various failure conditions, maintaining task completion even with UAV failures (albeit with extended mission duration), and its scalability is validated through performance assessment.

The paper is outlined as follows: Section II briefly introduces the background. Section III provides a comprehensive overview of FOCUS, detailing its key functionalities. Section IV describes the experimental setup. Section V presents the performance evaluation and results analysis. Section VI discusses the limitations and future perspectives. Finally, Section VII summarizes the findings and provides concluding insights.

II. Background

Currently, the visual inspection technologies, including optical photography, thermography, and computer vision are commonly mounted on UAVs for offshore wind turbines inspections [5]. UAVs equipped with functional sensors, such as high-resolution cameras, can be remotely or automatically operated to capture images and videos of critical components [7]. The collected data is subsequently processed and analyzed using intelligent algorithms for damage detection, classification, localization, and quantification [6]. Based on the analysis results, operation decisions such as condition rating and maintenance strategy formulation can then be made properly. In this work, FOCUS is solely concerned with the coordinated inspection data collection, without addressing the subsequent stages of data analysis or decision-making.

Many studies [10] [11] on multi-UAV inspection focus on optimizing path planning, task assignment, and resource allocation under complex constraints. However, such model-based optimization approaches typically operate in offline mode, relying on *a priori* knowledge (*i.e.*, a predefined environmental model). While effective under known conditions, these methods struggle to adapt to real-time changes and unforeseen failures, limiting their resilience in dynamic offshore environments. Additionally, to address UAVs' limited computational capabilities, Cao *et al.* [16] introduced a mobile edge computing-driven UAV inspection scheme, enabling local data processing or task offloading to ground stations or satellites. However, it lacks inter-UAV communication and coordination, treating UAVs as isolated units rather than a cohesive swarm leveraging collective intelligence. As a result, the system does not fully exploit the advantages of UAV swarm coordination, such as workload sharing, adaptive collaboration, and fault tolerance. To the best of our knowledge, FOCUS is the first multi-UAV inspection approach that employs distributed coordination software layer within swarm to collaboratively overcome failures and environmental uncertainties, enhancing both fault tolerance and efficiency of offshore wind farm inspections.

III. FOCUS Overview

This section first introduces the system model, then outlines two swarm intelligence-based capabilities, and finally explains in detail how cooperative inspection tasks are carried out.

A. System model

The consensus protocol investigated in this work is considered in a UAV swarm that consists of a set of N nodes, constituting a replica set, whose goal is to inspect a wind farm composed of O OWTs.

Failure model. UAVs and sensors may fail by crashing but do not experience arbitrary behaviour (i.e., no Byzantine failures). The communication network is unreliable, mostly due to packet loss, and is subject to unpredictable latencies, as well as load imbalances (e.g., peak demand), that are imposed on both nodes and the message exchange. Such imbalances may cause variations in transmission delays. However, nodes rely on reliable one-to-one communication channels, where transmitted messages can be lost (and retransmitted) but not corrupted. Therefore, the system tolerates f faulty nodes such that $f < \lceil N/2 \rceil$ (a quorum exists).

Timing model. We assume that the system is partially synchronous [17], that is, it is initially asynchronous and eventually becomes synchronous because a consensus protocol cannot be both safe and live under asynchronous assumptions, as the FLP impossibility result [18] states.

B. Strongly Consistent Data Replication

FOCUS leverages a distributed consensus protocol and mobile ad hoc connectivity to enable two swarm intelligence-based capabilities, namely distributed fleet configuration management and inspection data synchronization.

The distributed fleet configuration management permits each UAV to access critical, swarm-wide configuration data, including swarm membership, communication capabilities, and sensor status. Whenever a sensor failure occurs or a UAV completely stops working, UAVs exchange distributed consensus protocol's messages, via the wireless mesh network, to update the swarm configuration in a strongly consistent manner. It is worth noting that such swarm-wide consistent configuration state is key to inspection task coordination, proper mission execution and termination.

The second capability, inspection data synchronization, relies on distributed consensus to provide a fault-tolerant, strongly consistent distributed storage. This ensures both durability and availability of inspection data through consistent replication. To this end, inspection data should be replicated across all available UAVs via the wireless mesh network to form a replica set. Indeed, such replica set provides an atomic distributed storage, which, in turn, simplifies a lot the design of swarm intelligence-based services for UAV swarms.

To successfully replicate a new piece of inspection data or to safeguard fleet configuration consistency, a UAV must synchronize it by exchanging messages of consensus protocol with a simple majority of UAVs; otherwise, the services provided by consensus protocol are considered unavailable. In the next part, we describe how FOCUS successfully handles this intrinsic property of the distributed consensus protocol, which is further discussed in Section VI.

C. Cooperative Inspection Tasks

FOCUS enables cooperative inspection tasks at the swarm level atop of its strongly consistent data replication software layer. Actually, the impact of node mobility on consensus protocol performance is significant, as frequent network partitions can lead to data unavailability [15]. To tackle this issue, we integrate FOCUS with a simple local storage protocol. Each UAV first stores inspection data in a local database and, upon being within range of other UAVs and the replica set becomes available, it replicates the data through the distributed consensus protocol. A simple boolean variable, *"replication status"*, is introduced to track whether the data has been successfully replicated. Accordingly, FOCUS executes two procedures, namely local OWT inspection and inspection data replication.

Local OWT inspection. During the mission, each UAV periodically checks whether its current position is close enough to an OWT. If it does and the turbine has not yet been inspected by the swarm, the UAV initiates the inspection process. Upon collecting new inspection data, FOCUS stores it locally and sets the replication status of the data to *False*, meaning that the data has not yet been replicated within the swarm. Note that although the local storage does not provide swarm-wide fault tolerance, it plays an important role to handle an eventual unavailability of the replication software layer due to temporary issues in the ad hoc connectivity graph. We provide a detailed description of the local OWT inspection procedure in Algorithm 1.

Algorithm 1: Local OWT inspection procedure.

```
Input: UAV real-time coordinates, set of OWT coordinates
Output: Updated data_bank containing inspection records
new_data_entries ← [];
for each owt_key in owt_farm do
    Compute distance between UAV and the corresponding OWT;
    if distance < inspection_radius then
        if owt_key ∉ data_bank then
            Construct a data record for the inspected OWT;
            data["replication_status"] ← False;
            Append (owt_key, data) to new_data_entries;
for each (owt_key, data) in new_data_entries do
    Insert data into data_bank under key owt_key;
```

Algorithm 2: Inspection Data Replication.

```
Input: data_bank
Output: updated data_bank
unreplicated_data ← [];
for each (owt_key, data) in data_bank do
    if data["replication_status"] = False then
        Append (owt_key, data) to unreplicated_data;
for each (owt_key, data) in unreplicated_data do
    if replica_set is available then
        if replica_set.get(owt_key) = None then
            Set data["replication_status"] ← True;
            replica_set.put(owt_key, data);
        else
            Set data["replication_status"] ← True;
```

Algorithm 3: Local Storage Update.

```
Input: data_bank, replica_set.events
Output: updated data_bank
for each event in replica_set.events do
    owt_key, data ← DECODE(replica_set.get(event.key));
    if owt_key ∉ data_bank or
    data_bank[owt_key]["replication_status"] = False then
        data_bank[owt_key] ← data;
```

Inspection data replication. In order to ensure fault tolerance for the inspection mission, each UAV periodically scans its local storage for unreplicated inspection data, identifiable by a replication status of *False*. Upon detecting such data, the UAV attempts to communicate through the strongly consistent data replication software layer to perform replication. If the replication succeeds, the replication status of the corresponding data entry is updated to *True*. The detailed steps of this inspection data replication are described in Algorithm 2. Algorithm 3 handles local storage updates by continuously monitoring changes in the replica set through the consensus protocol. When a new replication event is detected, the corresponding data entry is retrieved and saved locally, ensuring the replication status is correctly updated. It is important to highlight that data synchronization plays a crucial role in mission management. The UAV swarm autonomously verifies mission completion by checking whether all OWT inspection data has been successfully stored in the replica set. Once the full dataset is available, the mission is considered complete, guaranteeing that all available UAVs have access to the necessary inspection records.

IV. Experimental Settings

In this section, we introduce the experimental settings, with Table I summarizing the key parameters.

TABLE I
MAIN PARAMETERS IN EXPERIMENTAL SETTINGS (FOR 3 UAVS INSPECTING 9 OWTS).

Parameter	Value	Unit
Number of OWTs	9	
OWT rotor diameter D	120	m
OWT spacing in the prevailing wind direction $5D$	600	m
OWT spacing perpendicular to the wind $3D$	360	m
Safety distance d_s [5]	5	m
Inspection area radium	65	m
Number of UAVs	3	
UAV communication range	300	m
UAV communication bandwidth	150	$Kbps$
UAV flight area	1400×900	m^2
UAV inspection check interval	5	s
Unreplicated data check interval	8	s
Routing protocol	B.A.T.M.A.N IV [19]	

Wind farm. We initially emulate a wind farm composed of a set of O modern OWTs, each with a rotor diameter of 120 meters, based on the Siemens SWT-3.6-120 turbines used in the London Array offshore wind farm [20]. According to Gilbert M. Masters [21], the optimal spacing between wind turbines within a row is 3–5 rotor diameters, while the spacing between rows is 5–9 rotor diameters. Therefore, we adopt a compact layout with three rotor diameters (360 m) between rows and five diameters (600 m) in the wind direction. The OWTs are evenly spaced in a rectangular layout, as illustrated in Figure 1. The inspection area of each OWT is a circle with a radius combining the rotor radius and a five-meter safety distance to ensure collision avoidance during UAV operations, as recommended in [5]. The flight area of UAVs is set according to the wind farm's size, for instance, in the experiment with three UAVs inspecting nine OWTs, the flight area is set as a 1400×900 m² rectangular region.

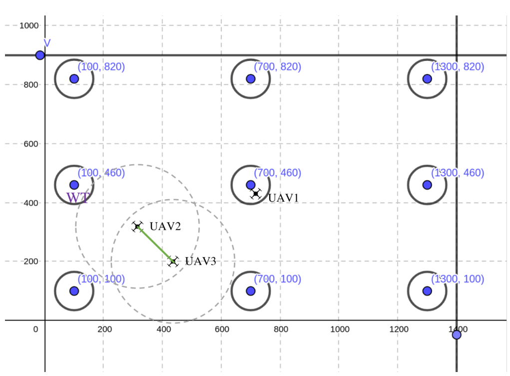


Fig. 1. A three-UAV swarm is inspecting a nine-OWT wind farm, turbine towers are marked by blue dots, inspection area of each OWT is represented as a black circle.

UAV swarm. Regarding UAV settings, we consider small-sized swarms only, *i.e.*, $N = \{3, 5, 7\}$, because there is a well-studied consensus protocol's trade-off between the performance and fault tolerance [15]. Thus, although bigger swarms provides higher fault tolerance, they result in poorer performance due to higher communication overhead for synchronization.

UAV communication. We assume that UAVs are equipped with wireless network interfaces to enable ad hoc mesh communication within the swarm. According to [22], it is observed that IEEE 802.11 and IEEE 802.15.4 are widely used technologies in consumer and industrial UAV systems for remote control. Based on the IEEE 802.11ah standard and in-field measurements [23], we set the UAV communication range to 300 meters and the bandwidth to 150 Kbps. In-range UAV pairs can directly exchange messages, while out-of-range pairs communicate through intermediate nodes if a reachable route exists. Routes are dynamically updated using *B.A.T.M.A.N. IV* [19], a real-world routing protocol for highly dynamic mobile ad hoc networks.

Synchronisation. During flight, the UAV checks every 5 seconds to see if it has entered the inspection radius of any OWT. During the data replication process, the UAV scans its local database every 8 seconds to detect any new inspection data that has not yet been replicated. What's more, the time required for the UAV to use onboard equipment to collect data from each OWT is ignored in the experiments, focusing the assessment solely on coordination processes.

Consensus protocols implementation. FOCUS relies on *etcd* [24], a distributed key-value store based on *Raft* [14], a well-known distributed consensus protocol known for achieving strong consistency in fault-tolerant distributed systems.

V. Performance Evaluation

The main objective of our performance evaluation through extensive emulation experiments is to investigate the capacity of a UAV swarm to inspect offshore wind farms. Thus, the evaluation focuses on three key aspects: quantifying the efficiency of FOCUS, demonstrating its fault tolerance, and assessing its scalability. We aim to answer the following questions: (1) How much faster is FOCUS compared to non-collaborative approaches that do not leverage swarm intelligence, particularly ad hoc connectivity and distributed consensus protocols? (2) How resilient is FOCUS against realistic failures scenarios? (3) How does FOCUS perform at scale, when deployed with larger UAV swarms?

A. Methodology

Baselines. We compare FOCUS to two approaches that leverage neither ad hoc connectivity nor coordination within the swarm to perform offshore wind farm inspection. Indeed, these naive approaches are attractive for their simplicity that facilitates the implementations. `Baseline 1` is an intuitive best-first approach where a predefined subset of OWTs is assigned to each UAV. For instance, if a set of N UAVs inspects a wind

TABLE II
COMPARISON OF `BASELINE 1`, `BASELINE 2` AND FOCUS SCENARIOS.

Scenarios	`Baseline 1`	`Baseline 2`	**FOCUS**
Mission assignment	UAVs are assigned to a subset of OWTs.	UAVs are assigned to all OWTs.	UAVs replicate and synchronize data via consensus protocol within the swarm.
Completion criteria e.g. 3 UAVs inspecting 9 OWTs	UAV1 inspects OWT1, OWT2, OWT3; UAV2 inspects OWT4, OWT5, OWT6; UAV3 inspects OWT7, OWT8, OWT9.	UAV1 inspects all 9 OWTs; UAV2 inspects all 9 OWTs; UAV3 inspects all 9 OWTs.	At least 1 UAV gets all 9 OWTs replication data.
Coordination	No	No	Yes
Fault tolerance	No	Yes	Yes

farm composed of a set of O OWTs, each UAV inspects a predefined, non-intersecting subset of S OWTs, where $S \approx \lceil \frac{O}{N} \rceil$. Consequently, each UAV executes its task independently by inspecting its S OWTs and the overall inspection mission is completed whenever all tasks were executed properly. The second approach, namely `Baseline 2`, is similar to the previous one, with an important improvement, it introduces redundancy by assigning intersecting subsets of S OWT. For the sake of fault tolerance, it offers the best resilience against faults without requiring coordination whenever $S = O$, in other words, if each UAV inspects all OWTs.

Metrics. We use three performance metrics. The first and most important metric is *completion time*, the amount of time for the UAV swarm to fully accomplish the inspection of the entire wind farm. The second metric is *completion rate*, which is actually a alternative of completion time, particularly relevant for assessing FOCUS's resilience. In scenarios where injected failures prevent the UAV swarm from completing the mission within the experimental time limit, this second metric captures the percentage of mission completion. The third metric is *swarm availability*, a measure of FOCUS's ability to enable distributed configuration and synchronization despite network partitions caused by the underlying mobile ad hoc networks. This metric also allows us to assess the scalability by quantifying how effectively the swarm maintains connectivity and coordination as the swarm size increases.

Experimental environment. Our experiments rely on the Mobile Ad-hoc Computing Emulator [25], *MACE*, an open-source emulation framework designed for mobile cyber-physical system (CPS) swarms. *MACE* generates a Linux-based, realistic virtual environment in terms of embedded processing, CPS's mobility and wireless communication capabilities, which allows us to run and evaluate novel distributed remote sensing applications atop mobile ad hoc networks promptly and efficiently. *MACE* emulates horizontal movements of CPS only, which is suitable for our experimental settings. Moreover, each experiment was run at least 10 times to account for the variance of our settings, mostly related to our mobility model.

UAV swarm deployment and mobility model. We assume that the UAV swarm is dispatched from the same starting point. Each UAV is equipped with a precise positioning system and they have the geo-coordinates of all OWTs. During the inspection mission, UAVs navigate across the wind farm executing a predetermined mobility model. To ensure both generality of the experiments and reproducibility, the Random Direction Mobility Model [26] is adopted to emulate a random UAV path planning. In addition, UAVs fly within a predetermined inspection area at a speed of 5 metres per second [5]. Whenever a UAV reaches the boundary of the inspection area, it randomly selects a new direction, between 0 and 180 degrees, and continues its inspection task. For simplicity, we assume that a UAV is able to inspect an OWT when its position is within the inspection area.

B. *FOCUS's efficiency*

We compare FOCUS with the two baselines to evaluate its efficiency in terms of the mission completion time, as outlined in Table II. In this set of experiments, we assume failure-free UAV conditions with unlimited battery endurance, which allows us to focus on efficiency only, and each set of experiments is repeated 20 times to provide consistent results.

Figure 2 depicts the results of each execution (x-axis) per approach where the y-axis denotes the mission completion time. Thanks to its capacity of replicating inspections data across the swarm, FOCUS show a faster and more stable overall completion time of roughly 33 minutes on average, reducing the inspection time by 55% compared to `Baseline 1` (73 minutes) and 70% compared to `Baseline 2` (110 minutes).

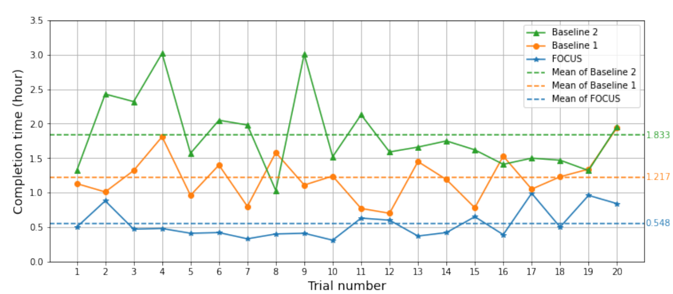
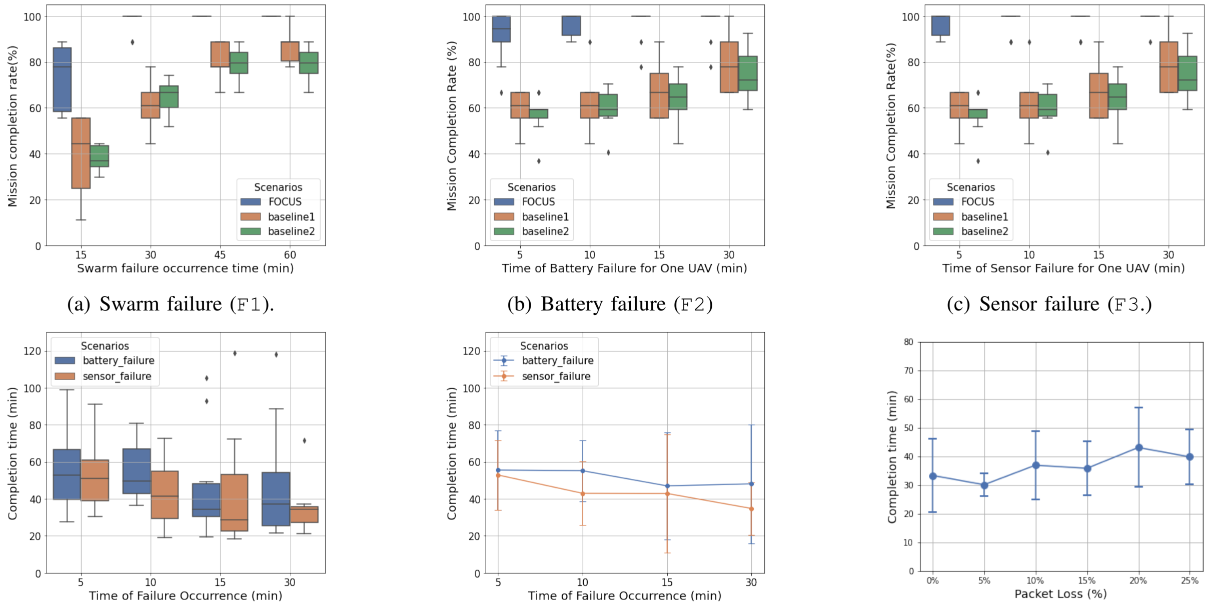


Fig. 2. Completion time comparisons between FOCUS, `Baseline 1` and `Baseline 2`.

C. *FOCUS's resilience against failure*

Building on the study of UAVs' critical failure modes by Shafiee et al. [27], we identify key failure scenarios for evaluating FOCUS's fault tolerance. These include swarm failure `F1` (complete UAV unavailability due to issues such



(a) Swarm failure (F1). (b) Battery failure (F2) (c) Sensor failure (F3.)

(d) Comparing F2 and F3. (e) Comparing F2 and F3. (f) Packet loss (F4).

Fig. 3. Evaluation results of FOCUS's resilience.

as battery depletion or propeller malfunction), single UAV battery failure F2, and single UAV sensor failure F3 (e.g., camera malfunction). Additionally, to reflect the realistic challenges of deploying consensus protocols in MANETs, such as unpredictable message loss and network partitions, we also include transmission packet loss as an injected failure scenario, denoted as F4. The injected failure modes are summarized in Table III and fully detailed below. Each experiment is repeated 10 times and since some settings may fail to complete the inspection mission, we compute the completion time rate.

TABLE III
FAILURE INJECTION SCENARIOS.

Scenarios	Baseline 1	Baseline 2	FOCUS
F1: Swarm failure. All UAVs stop working at the specified times.	✔	✔	✔
F2: One UAV battery failure. A subset of UAVs stops working at the specified times.	✔	✔	✔
F3: One UAV sensor failure. A subset of UAVs stops inspecting at the specified times.	✔	✔	✔
F4: Packet loss. Different packet loss impacts communication performance	✗	✗	✔

*1) **F1** - swarm failure:* We emulate a swarm failure setting in which all UAVs stop working at predefined moment of the mission, *i.e.*, 15, 30, 45 and 60 minutes after stating the inspection mission. The results, shown in Figure 3(a), show that FOCUS outperforms baselines by inspecting more OWTs achieving the highest completion rate for all settings, *e.g.*, almost 80% of OWT were inspected on average within 15 minutes. While Baseline 2 shows a slight improvement over Baseline 1, both remain significantly less efficient than FOCUS. Thanks to data synchronization, FOCUS is likely to complete the mission if the swarm failure occurs after 30 minutes, in line with findings detailed in SectionV-B, whereas both Baseline 1 and Baseline 2 mostly fail to accomplish the mission if the failure occurs before the first hour of inspection.

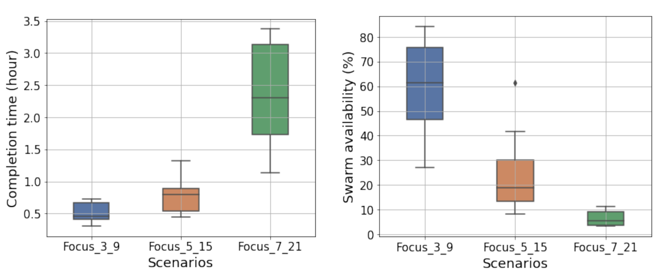
*2) **F2** - one UAV's battery failure:* This setting emulates a single-UAV battery failure, injected at 5, 10, 15, and 30 minutes, since the average mission completion time for FOCUS is approximately 30 minutes. Figure 3(b) depicts the mission completion for this setting. For simplicity and to ease the performance comparisons, the experiment duration of each run is set to 60 minutes. The findings show that despite one UAV stops inspecting, FOCUS leverages the mission coordination and synchronization capability of the remaining two UAVs to complete the mission more efficiently than Baseline 1 and 2, demonstrating its strong fault tolerance. Moreover, even when a single-UAV battery failure occurs at only 10 minutes after the start of the mission, FOCUS still completes the mission within one hour duration for most of the experiments.

*3) **F3** - one UAV's sensor failure:* In this setting, we are interested in evaluating the impact of the failure of one UAV's sensor. It is important to note though that in this case the UAV continues flying and relaying inspecting data within the swarm. The overall performance on F2 and F3 (Fig. 3(c)) shows similar trends, despite FOCUS performs slightly better when the latter failure is injected. To further investigate the performance gains achieved by FOCUS, we compared the results under F2 and F3 failures. As depicted in Figures 3(d) and 3(e), the capacity of the third UAV of relying messages using FOCUS has a minor effect on the overall completion time. This finding suggests that most of the performance gains of FOCUS using a small replica set (*i.e.*, three) stems from two UAVs, yet the third UAV remains key to guarantee the fault tolerance at the swarm level. Moreover, it highlights the feasibility of using FOCUS to inspect wind farms efficiently.

*4) **F4 - packet loss impact**:* To evaluate the impact of packet loss of MANETs on FOCUS, we conducted experiments with increasing packet loss rates: 5%, 10%, 15%, 20%, and 25%. As presented in the Figure 3(f), our findings reveal that packet loss has a moderately little impact on the completion time thanks to FOCUS's fault-tolerant capability. Upon further investigation on the availability of the replica set that enables such a fault tolerance, we observe that FOCUS is unable to synchronize data through it roughly 40% of the mission duration on average, regardless of the packet loss rate. This result helps us to better understand the low sensitivity of FOCUS to packet loss and highlights the importance of our simple local storage protocol to ensure the durability of unreplicated inspecting data whenever the synchronization through the replica set is unavailable. The availability of the replica set is further investigated in Subsection V-D.

D. FOCUS's Scalability

To examine the scalability of FOCUS, we increase the UAV swarm size from three to five and seven. Consequently, we expand the number of OWTs proportionally, *i.e.*, from 9 to 15 and 21. To accommodate a bigger number of OWTs, the inspection areas are scaled up to 1400×1600 m² and 1400×2300 m² respectively. Figure 4(a) shows the comparison of the three different scenarios regarding mission completion time, while Figure 4(b) illustrates the swarm availability in percentage of the mission duration time.



(a) Completion time (b) Swarm availability

Fig. 4. Performance evaluation for FOCUS's scalability.

Compared to inspecting 9 OWTs with three UAVs, increasing the swarm to five UAVs for 15 OWTs resulted in a 49.6% increase, on average, of the mission duration (from 31 minutes to 46 minutes) and a 59.3% decrease in swarm availability (from 59.2% to 24.1%). Despite the significant drop in swarm availability, the mission time increased by about 50% only, which is lower than the 77% increase in inspection area, suggesting that FOCUS still scales well. Whenever the swarm size increases to seven UAVs inspecting 21 OWTs though, FOCUS experiences substantial performance degradation, raising the average mission completion time to 142 minutes, an order of magnitude longer. Additionally, swarm availability dropped drastically to an average of only 6.5%. In summary, the results clearly show that the smaller the swarm size, the better is the performance both in terms of mission completion time and swarm availability. Unsurprisingly, our findings are in line with previous works on distributed consensus, which highlight scalability as one of its major performance issue, as described by Ailijiang *et al.* [13].

VI. DISCUSSION

This section discusses the limitations of FOCUS, and a few issues that represent corner-cases to its design.

The integration with an advanced path planning algorithm would boost FOCUS performance. For simplicity and generality though, FOCUS currently relies on a random mobility model for multi-UAV inspection, which represents the worst case in terms of the path planning. As surveyed by Foster *et al.* [28], a clear distinguishing is made between decision making (*i.e.*, path planning) and coordination in multi-robot coverage for OWT inspection. Decision making corresponds to the collective choices defined by specified objectives made by a multi-robot system. The applicable studies were analyzed using the model/non-model distinction by Almadhoun *et al.* [29], or the task decomposition, task assignment, and motion planning framework by Yan *et al.* [30]. Coordination, on the other hand, addresses mechanisms necessary to manage environmental dynamics and real-time operational challenges, emphasizing communication strategies to enable effective collaboration. While distinct, decision making and coordination are highly interdependent, as most decision-making approaches require effective coordination to improve system utility. FOCUS prioritises coordination, employing distributed computing techniques to enable UAV cooperation. It also uses data synchronisation to track progress and determine mission completion. Our research specifically investigates how mobile ad hoc networks and distributed consensus protocols can effectively coordinate UAV swarms for offshore wind farm inspection. As a result, path planning optimization is beyond the scope of this work, as our primary goal is to establish an effective coordination strategy rather than optimizing UAV trajectories.

Distributed consensus in MANETs. The communication cost of distributed consensus is high and, according to the CAP theorem proved by Gilbert and Lynch [31], there is a fundamental trade-off between availability and consistency. Therefore, whenever network partitions occur and no majority of UAVs exists, the data synchronization becomes unavailable. Unfortunately, it turns out that network partitions are frequent in mobile ad hoc networks with randomly moving nodes. Indeed, our performance evaluation shows that UAVs are unable to synchronize data for roughly 40% of the total mission duration on average, with a high coefficient of variation (32%). These results show that, unsurprisingly, the replica set for distributed fault tolerance is frequently unavailable, which highlights the importance of FOCUS's simple local storage protocol to temporarily ensure the durability of unreplicated inspection data. Moreover, the well-studied scalability issues of distributed consensus [13] are further complicated by the time-varying communication graphs that arise from MANETs.

For this reason, small-sized UAV swarms are more suitable for offshore wind farm inspection that requires fault tolerance.

Energy consumption required for communication is negligible compared to propulsion energy. Although FOCUS requires more energy for communication than non-coordination approaches, communication-related energy is generally negligible compared to propulsion energy. For instance, communication consumes only a few watts, whereas UAV propulsion typically requires hundreds [32]. With an average packet size of about 123 bytes, FOCUS maintains low communication energy consumption, but confirming this potential advantage requires further investigation.

VII. Conclusion

We propose FOCUS, a fault-tolerant, mission-level approach for efficient offshore wind farm inspection using a UAV swarm. By leveraging the distributed consensus protocol and wireless ad-hoc connectivity, FOCUS is able to make progress for the inspection mission in a fault-tolerant way as long as a majority of the swarm's UAVs are available. Meanwhile, the distributed techniques also enable the UAVs to cooperate autonomously as they keep track of the swarm configuration. An extensive evaluation of FOCUS based on emulation techniques shows empirical evidence of its efficiency as compared to the baselines without coordination capabilities, as well as its resilience to faults and its scalability, suggesting its suitability to enable autonomous unmanned systems in renewable wind energy infrastructure inspection.

Acknowledgments

We thank the anonymous reviewers for their insightful feedback. This work was supported in part by the ULYSSES PHC programme, operated by the French Embassy in Ireland and the Research Council in Dublin, with ADEME and SEAI.

References

[1] "Offshore renewable energy," European Commission, 2024. [Online]. Available: https://energy.ec.europa.eu/topics/renewable-energy/offshore-renewable-energy_en

[2] E. Marino, M. Gkantou, A. Malekjafarian, S. Bali, C. Baniotopoulos, J. van Beeck, R. P. Borg, N. Bruschi, P. Cardiff, and E. Chatzi, "Offshore renewable energies: A review towards floating modular energy islands—monitoring, loads, modelling and control," *Ocean Engineering*, vol. 313, p. 119251, 2024.

[3] M. Vieira, E. Henriques, B. Snyder, and L. Reis, "Insights on the impact of structural health monitoring systems on the operation and maintenance of offshore wind support structures," *Structural Safety*, vol. 94, p. 102154, 2022.

[4] K. Zhang, V. Pakrashi, J. Murphy, and G. Hao, "Inspection of floating offshore wind turbines using multi-rotor unmanned aerial vehicles: literature review and trends," *Sensors*, vol. 24, no. 3, p. 911, 2024.

[5] Y. Liu, M. Hajj, and Y. Bao, "Review of robot-based damage assessment for offshore wind turbines," *Renewable and Sustainable Energy Reviews*, vol. 158, p. 112187, 2022.

[6] A. Khadka, B. Fick, A. Afshar, M. Tavakoli, and J. Baqersad, "Non-contact vibration monitoring of rotating wind turbines using a semi-autonomous uav," *Mechanical Systems and Signal Processing*, vol. 138, p. 106446, 2020.

[7] A. Khadka, A. Afshar, M. Zadeh, and J. Baqersad, "Strain monitoring of wind turbines using a semi-autonomous drone," *Wind Engineering*, vol. 46, no. 1, pp. 296–307, 2022.

[8] G. Skorobogatov, C. Barrado, and E. Salamí, "Multiple UAV systems: A survey," *Unmanned Systems*, vol. 8, no. 02, pp. 149–169, 2020.

[9] A. J. I. Foster, M. Gianni, A. Aly, H. Samani, and S. Sharma, "Multi-robot coverage path planning for the inspection of offshore wind farms: A review," vol. 8, no. 1, p. 10. [Online]. Available: https://www.mdpi.com/2504-446X/8/1/10

[10] S. Ivić, B. Crnković, L. Grbčić, and L. Matleković, "Multi-uav trajectory planning for 3d visual inspection of complex structures," *Automation in Construction*, vol. 147, p. 104709, 2023.

[11] T. Fan, L. Fu, C. Guo, Y. Zhang, and L. Sun, "Multi uav inspection optimization for offshore wind farms considering battery exchange process," *IEEE Transactions on Intelligent Vehicles*, 2024.

[12] S. Bernardini, F. Jovan, Z. Jiang, S. Watson, A. Weightman, P. Moradi, T. Richardson, R. Sadeghian, and S. Sareh, "A multi-robot platform for the autonomous operation and maintenance of offshore wind farms," ser. AAMAS '20, Richland, SC, 2020, p. 1696–1700.

[13] A. Ailijiang, A. Charapko, and M. Demirbas, "Dissecting the performance of strongly-consistent replication protocols," in *MOD*, 2019, pp. 1696–1710.

[14] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *USENIX ATC*, 2014, pp. 305–319.

[15] J. Chen, M. Vacheron, B. C. Ferreira, and G. Silvestre, "Assessing distributed consensus performance on mobile cyber-physical system swarms," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2023, pp. 650–656.

[16] P. Cao, Y. Liu, C. Yang, S. Xie, and K. Xie, "Mec-driven uav-enabled routine inspection scheme in wind farm under wind influence," *IEEE Access*, vol. 7, pp. 179 252–179 265, 2019.

[17] C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the presence of partial synchrony," *JACM*, vol. 35, no. 2, pp. 288–323, 1988.

[18] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *JACM*, vol. 32, no. 2, pp. 374–382, 1985.

[19] D. Johnson, N. S. Ntlatlapa, and C. Aichele, "Simple pragmatic approach to mesh routing using batman," 2008.

[20] "London Array - The world's largest offshore wind farm," Online, London Array, April 2013, accessed: 2025-02-07. [Online]. Available: http://www.londonarray.com/downloads/london_array_brochure.pdf

[21] G. M. Masters, *Renewable and Efficient Electric Power Systems*. John Wiley & Sons, 2013.

[22] L. Shi, N. J. H. Marcano, and R. H. Jacobsen, "A review on communication protocols for autonomous unmanned aerial vehicles for inspection application," *Microprocessors and Microsystems*, vol. 86, p. 104340, 2021.

[23] B. Bellekens, L. Tian, P. Boer, M. Weyn, and J. Famaey, "Outdoor IEEE 802.11 ah range characterization using validated propagation models," in *GLOBECOM*. IEEE, 2017, pp. 1–6.

[24] "etcd project homepage," https://etcd.io/, accessed on 2024-11-02.

[25] B. C. Ferreira, G. Dufour, and G. Silvestre, "Mace: A mobile ad-hoc computing emulation framework," in *ICCCN*, 2021, pp. 1–6.

[26] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[27] M. Shafiee, Z. Zhou, L. Mei, F. Dinmohammadi, J. Karama, and D. Flynn, "Unmanned aerial drones for inspection of offshore wind turbines: A mission-critical failure analysis," *Robotics*, vol. 10, no. 1, p. 26, 2021.

[28] A. J. Foster, M. Gianni, A. Aly, H. Samani, and S. Sharma, "Multi-robot coverage path planning for the inspection of offshore wind farms: A review," *Drones*, vol. 8, no. 1, p. 10, 2023.

[29] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai, "A survey on inspecting structures using robotic systems," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 1729881416663664, 2016.

[30] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.

[31] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *Acm Sigact News*, vol. 33, no. 2, pp. 51–59, 2002.

[32] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Transactions on wireless communications*, vol. 16, no. 6, pp. 3747–3760, 2017.