Journal of Intelligent & Robotic Systems (2021) 102: 44
https://doi.org/10.1007/s10846-021-01372-8

SHORT PAPER

HMDCS-UV: A concept study of Hybrid Monitoring, Detection and Cleaning System for Unmanned Vehicles

Salima Bella[1] · Ghalem Belalem[1] · Assia Belbachir[2] · Hichem Benfriha[1]

Received: 24 July 2020 / Accepted: 18 March 2021 / Published online: 25 May 2021

Abstract

Incidents of hydraulic or oil spills in the oceans/seas or ports occur with some regularity during the exploitation, production and transportation of petroleum products. Immediate, safe, effective and environmentally friendly measures must be adopted to reduce the impact of the oil spill on marine life. Due to the difficulty to detect and clean these areas, semi-autonomous vehicles can make a significant contribution by implementing a cooperative and coordinated response. The paper proposes a concept study of Hybrid Monitoring Detection and Cleaning System (HMDCS-UV) for a maritime region using semi-autonomous unmanned vehicles. This system is based on a cooperative decision architecture for an unmanned aerial vehicle to monitor and detect dirty zones (i.e., hydraulic spills), and clean them up using a swarm of unmanned surface vehicles. The proposed solutions were implemented in a real cloud and were evaluated using different simulation scenarios. Experimental results show that the proposed HMDCS-UV can detect and reduce the level of hydraulic pollution in maritime regions with a significant gain in terms of energy consumption.

Keywords UAV (Unmanned Aerial Vehicle) · USV (Unmanned Surface Vehicle) · Swarm · Hybrid architecture · Detection · Trajectory planning

1 Introduction

Nowadays, research in swarm robotics is growing due to their robustness, parallelism and flexibility. Unlike distributed robotic systems, swarm robotics focuses on a large number of robots and promotes scalability [15, 24]. Several applications of these systems concern environmental exploration, resource detection and monitoring, cleaning, rescue, food search in agriculture, etc. [15]. Many applications, such as surveillance, perimeter / surface detection and cleaning of polluted zones, have several ecological uses.

Maritime activity has gradually increased in recent years. Many ships carry products such as hydraulics that can harm the environment. These products can produce high levels of pollution if spilled at sea. This oil/ hydraulic pollution, which may be caused by marine accidents or normal ship operations, is dangerous for the marine environment, with possible long-term effects. The extent of the damage depends on the quantity of oil spilled, geographical and environmental conditions, and the nature of marine life in the zone. The cost of cleaning and the damage caused could reach billions of euros [17]. For example, in 2010, the Deepwater Horizon oil spill in the Gulf of Mexico was considered one of the largest accidental oil spills in the history of the oil industry, with more than 210 million gallons of crude oil spilled on the ocean surface over 180,000 km^2. The oil spill cleaning process involved more than 39,000 people, 5,000 ships and 110 aircraft [20]. Also, recently in Siberia, 20,000 tonnes of diesel fuel was spilled in 2 rivers in Norilsk after a leak in an oil depot. A subsidence of the soil occurred damaging the storage tank and the melting of the permafrost is believed to be the cause. The traces of oil have been identified despite the floating dams placed in emergency and the pollution levels noted are much higher than the authorized threshold [23].

Current oil spill recovery and cleaning technology includes many techniques (physical and / or chemi-

✉ Salima Bella
bella.salyma@gmail.com

1 Department of Computer Science, Faculty of Exact and Applied Sciences, Computer Laboratory of Oran (LIO), Université Oran 1, Oran, Algeria

2 Mechatronics Department, Polytechnic Institute of Advanced Sciences, IPSA, Ivry-sur-Seine, France

cal actions) such as controlled combustion, solidifiers (absorbents), skimming, oil oxidation, manual recovery, dispersion, bioremediation / biodegradation, etc. which are largely limited by marine conditions [19, 20]. While these treatments are important to rapidly control the spread and drift of oil, they are not suitable for ecological restoration. All the methods and techniques mentioned are only relevant to the elimination of oil spills in water, but this is not an easy task, as the spill usually spreads more widely over time. The only technology used to collect the spilled oil is the use of dams, which are large floating barriers that supplement the oil spill and then lift the oil from the water between two ships [11], as shown in Fig. 1 [1,2]. This process is long and costly, as ships carrying large containers have to leave the zone several times to dispose of the recovered oil. In the case of turbulent water, the spill spreads more widely, making it difficult to complete the cleaning process. Instead of using many large barges in the working zone, a swarm of robots equipped with skimmers and dams was proposed in [11] to collect the spilled oil in one place and limit its spread. Only vessels equipped with containers can be present to collect and store the oil. Such a swarm can also be sent to prevent the oil spill from moving to the shoreline, port or other such zone and save lives more effectively.

Based on this technique, a hierarchical decision architecture is illustrated in the proposed hybrid system (HMDCS-UV) for a better management and easier control between the different unmanned vehicles. In addition, the proposed system is efficient, robust and scalable compared to the self-organized approach which hardly allows scalability with increased complexity for efficient management of its entities. Centralized management has a central node with deterministic decision-making capability and easy to implement coordination. This central node has a global view of the unmanned monitoring and cleaning vehicles activities. Distributed management begins when the monitoring vehicle is assigned to a region. It moves to its region according to a proposed trajectory planning method, detects dirty zones based on an unsupervised classification method specific to image processing, monitors and supervises its cleaning swarms in its region.

This article proposes a concept study of Hybrid Monitoring, Detection and Cleaning System (HMDCS-UV) for a maritime region using heterogeneous semi-autonomous unmanned vehicles. The HMDCS-UV is based on a cooperative hybrid architecture of an Unmanned Aerial Vehicle (UAV) to monitor and detect dirty zones (hydraulic spills) and clean them from a swarm of Unmanned Surface Vehicles (USV). A general coordinator is proposed to manage all the tasks and coordinate these vehicles. The proposed UAV is supposed to be equipped with on-board sensors that allow it to move, detect and locate dirty zones and update the nautical chart using the specific planning and detection methods. After receiving and analyzing the collected data, the general coordinator assigns a swarm of USVs with the processed information to clean each dirty zone. This swarm navigates to the assigned dirty zone and cleans it according to the proposed solutions for trajectory planning and cleaning.

The paper is organized as follows: Section 2 presents some related work from the literature; Section 3 describes the proposed system for different unmanned vehicles with methods for detection, trajectory planning and cleaning; Section 4 illustrates an example to simulate the operation of the HMDCS-UV; in Section 5, the proposed HMDCS-UV is compared with related work; finally, Section 6 concludes this paper and provides some guidance for future research.

2 Related Works

The technical literature abounds with various solutions that address the problem of pollution of land, air and / or sea environments, using methods and algorithms for monitoring, detection, reduction / mitigation and cleaning. For example, the design of autonomous units (autonomous vessels / drones) was developed in the framework of an EU-MOP [17] research project for the elimination of marine oil pollution, which are capable of mitigating and eliminating the threat of small and medium sized spills. These vessels are released in the zone of the spill, automatically monitor (using appropriate sensors) the specificities of the oil concentration of the spill and locally apply mechanical or chemical countermeasures. An alternative design of a multi-robot autonomous aquatic vehicle system was proposed in [16] for lake cleaning and fisheries maintenance. The robots use tactile sensors and wireless communications to independently navigate and collectively perform cleaning operations such as removing surface impurities, pumping oxygen into the water, spraying chemicals, distributing food to appropriate locations while measuring water quality. Then, a chemical leakage localization and cleanup method was implemented in [25] using a robotic swarm and based on a bio-inspired exploration method. This method is based on a combination of two bio-inspired behaviors: aggregation and pheromone tracking. The main objective of the robots is to follow pheromone trails to find the source of a chemical leakage and then carry out a decontamination task by aggregation at the level of the critical zone.

A new swarm robotic system was proposed in [11] to locate and collect an oil spill on the water surface (ocean, river, lake, etc.). A coordinator determines the position and

[1] https://www.itopf.org/uploads/translated/TIP3FRUseofBoomsinOilPollutionResponse.pdf

[2] https://www.arzewports.com/?pages=page&rub=13

Fig. 1 Examples of dams collecting a hydraulic spill

the center of the oil spill using a GPS receiver, then a barge carrying the robots goes to the work site. Depending on the state of the oil spill, the coordinator may send a swarm of robots to surround and collect the spilled oil, then place the barge with oil suction equipment and move it to another location to safely remove the oil. The distributed system presented in [15] allows monitoring, recovery and containment of a resource using a swarm of homogeneous drones at low cost. A microscopic model of the swarm is presented, which defines individual behavior and is capable of locating and marking the perimeter of an oil spill. The functioning of a macroscopic model is analyzed and shows the trend of the swarm for a large number of agents. The authors of [15] suggest to use the signal power (at a given frequency) for obstacle avoidance tasks.

In [18], an experimental system was proposed for the autonomous control and coordination of the automatic spill response dam towing operation when using unmanned vessels. This system comprises two ASVs (Autonomous Surface Vehicles) and a ground station. Once the operation has begun, the ASVs tow the dam towards the target, minimizing the towing effort, close to the target that the ASVs deploy and advance, and finally approach, confined to the spill; then the drive moves to the destination. Another marine robotics system was designed in [19] to locate and quantify surface water (oil-based) pollution in lakes and ponds using aerial and marine robots, taking into account the influences of nearby buildings and trees. An aerial mobile robot equipped with two cameras such as the FLIR (Forward Looking Infra-Red) thermal imaging camera to locate and detect oil spills based on water temperature day and night, and a digital camera for trajectory planning. Quantification consists of a marine robot (boat) with a spectrometer on board. This marine robot controls its movement by fuzzy logic, distinguishes the received signal from the oil and water spectrum thanks to its integrated ultrasonic sensor, and collects samples of oil spilled by the suction pump. The neural network technique analyses the images received to differentiate between the different types of pollution on the water surface.

E-drones (Environmental Drones) is a new approach, was proposed in [14] for the large-scale elimination of air pollution. The environmental drones used make it possible to independently monitor air quality, detect the presence of pollutants and measure their concentrations (in carbon dioxide (CO2), carbon monoxide (CO), etc.), implement an option for appropriate reduction at a specific altitude (E-altitude) to guarantee the elimination of these pollutants, then fly to their ground locations. When multiple environmental drones are used in different locations, custom software generates an Air Quality Health Index (AQHI) map of the region for current and long-term environmental analysis. Thus, a method of controlling air pollution based on an air purifying drone system was presented in [26] to clean or reduce the amount of pollutants present in the areas near the industries or highly populated cities. The air purifier drone will disunite pollutants by spraying water and chemicals into the atmosphere.

Another hierarchical hybrid approach for the heterogeneous cooperation of unmanned vehicles (HA-UVC) was proposed in [1]. The proposed HA-UVC allows the cooperation of an unmanned aerial vehicle (UAV) to monitor an ocean region and a swarm of UAVs to clean up dirty zones. The UAV is supposed to be equipped with on-board sensors that allow it to locate the dirty zones using the proposed methods of travel, discretize its environmental map and update it with the information collected about the dirty zones. After receiving and analyzing the data collected according to the color of the water, the general coordinator (represented by a laptop computer and guided by a human operator) assigns the explored map to the swarm SVU to clean each dirty zone. This swarm moves to the assigned dirty zone according to the proposed solutions for trajectory planning, namely "Modified-GA" and "Proposed-CCA". In addition, these solutions incorporate a method to avoid static obstacles. When this swarm arrives in the dirty zone, it starts moving and cleaning the dirty cells according to a proposed algorithm. The proposed HA-UVC is complemented by a method of managing SUV failures during the execution of its cleaning task by measuring its energy quantity according to an energy threshold. Thus, a system for monitoring, detecting and cleaning up hydraulic spills was proposed in [22] using a UAV developed with integrated software. The UAV can be used as a "flying officer" that can monitor each part of the oil platform using a light Canon XS260 camera to capture images, a light detection and ranging (LiDAR) sensor for navigation and collision avoidance, and a GPS to map the spill zone where a geo-referenced method is used. The UAV can also clean up the spill by spraying chewing microbes with oil on the ocean surface. In addition, a fluorosensor system carried by a commercial drone was built in [28] for monitoring laser-induced fluorescence from the aquatic environment. The present version of the fluorosensor system could only be used at night time, which is a clear drawback compared to pulsed lidar systems with range gating, allowing daytime use, while again still functioning better in low ambient light level conditions. Also, a new system architecture derived from the integration of a low-cost laser-based network of detectors for pollutants interfaced with a more sophisticated layout mounted on an UAV was proposed in [29] to identify the nature and the amount of a release. Once this system is in place, the drone will be activated by the alarm triggered by the laser-based network when anomalies are detected. The area will be explored by the drone with a more accurate set of sensors for identification to validate the detection of the network of Lidar systems and to sample the substance in the focus zone for subsequent analysis.

A new elite group-based evolutionary algorithm (EGEA) was presented in [13] for maximum ocean sampling by several Unmanned Marine Vehicles (UMVs). The EGEA integrates a group-based framework and two proposed elitist selection methods, GIES (Group Individual Elitist Selection) and WPES (Whole Population Elitist Selection) to facilitate the selection of preferred solutions to be passed on to the next generation. The EGEA trajectory planners are based on the Simulated Annealing (SA) algorithm and Particle Swarm Optimization (PSO) to find the trajectories of the UMVs and to collect the maximum information on the studied regions. Mixed integer linear integer programming (MILP) was used by EGEA to solve the problem of adaptive sampling. Other algorithms were introduced in [21] for the detection of oil spills using MIMO (Multiple-input multiple-output) radar remote sensing integrated on a UAV. The first Single-Frequency Single-Observation (SFSO) sensor with power reflection coefficients is used to detect the presence of oil. The results show that the performance of this type of detector is related to oil thickness values, where they operate for one range but fail for another. An improvement of this detector is the Dual-Frequency Single-Observation (DFSO) detector where two frequencies of electromagnetic waves are used. Analysis of the performance of the second detector allows accurate detection. An improvement of both detectors is the use of multiple observations.

A metaheuristic algorithm based on the simulated annealing methodology to generate the UAV movement directions, integrated in a platform was proposed in [27] to measure atmospheric pollutants and to monitor contamination sources and treat them in real time. Another pollution-driven UAV control (PdUC) algorithm was presented in [30] to guide drones equipped with off-the-shelf sensors to perform air pollution monitoring tasks. This algorithm makes it possible to autonomously monitor a specific area by prioritizing the most polluted zone. In particular, it is able to find the most polluted areas more accuracy and cover the surrounding area, thus obtaining a complete and detailed pollution map of the target region within the time bounds defined by the UAV flight time. In addition, the authors of [31] were developed a real-time detection and monitoring system for the coronavirus (COVID-19) from the thermal image integrated into a UAV based on the Internet Of Things (IoT). The proposed system can detect ground surface temperatures from a height above the ground. Furthermore, the proposed design has capability for using Virtual Reality (VR), so the live video scanning process will be monitored through the VR screen to make it realistic and less human interaction. Thus, the diagnosis of the screening process will be less time consuming and less human interactions that might cause the spreading of the coronavirus faster.

A trajectory planning control method was developed in [20] for an Autonomous Surface Vehicle (ASV) that is capable of mitigating and bypassing the propagation of an oil spill while deploying micro-organisms and nutrients (bioremediation) with the collaboration of a UAV. The UAV is responsible for detecting the zone of the oil spill with the thermographic camera and controlling internal leakage zones by spreading freeze-dried indigenous microbial consortium dust on the oil spill, while the ASV releases the product mixed with water on the boundary zones of the line. The potential field method is used so that the ASV can plan its route, cover the entire zone and at the same time robustly avoid the oil spill. In addition, a multi-resolution navigation algorithm was presented in [12] to clean up oil spills in dynamic and uncertain environments using autonomous vehicles as the sole agent. A proposed algorithm for adaptive decision making based on sensory information, provides a complete coverage of the search zone for cleaning that does not suffer from the problem of local minima using potential field methods.

It is in this sense that this present work aims to provide a new solution to the challenges of maritime pollution by proposing a concept study of Hybrid Monitoring, Detection and Cleaning System for Unmanned Vehicles (HMDCS-UV). This proposed system enables cooperation and coordination between heterogeneous semi-autonomous air-sea unmanned vehicles to monitor maritime regions and clean their dirty zones. Thus, the solutions are proposed for the trajectory planning towards the dirty region / zone, the monitoring of a region, the detection and cleaning of the dirty zones and the supervision of unmanned vehicles.

3 Proposed System

This paper aims to propose a hybrid system for the surveillance, detection and cleaning of dirty / polluted maritime zones using the cooperation and coordination of semi-autonomous unmanned vehicles (HMDCS-UV). Thus, the HMDCS-UV is seen as an improvement and an extension of the HA-UVC solution [1]. In this section, a process for preventing and combating oil pollution in the port of Arzew (Algeria) is presented. Based on this process, the HMDCS system is described with its architecture, its constituent entities, the solutions proposed for supervision and detection, trajectory planning and cleaning, and their operation.

3.1 Prevention and Fight Against the Pollution of Hydrocarbons

Marine oil pollution from ships is a real problem that threatens the port of Arzew (oil port in western Algeria). Given the impact of these harmful products on the port and industrial activity of the site, prevention has become a constant concern for the authorities. The Arzew port has considerable material (tugs, oil waste recovery barge, self-floating containment dam, etc.) and human resources (people working in the anti-pollution cell, mooring staff, etc.) to combat oil pollution, and works in collaboration with other services at the port to intervene in the event of spills.

- *Presentation of the port of Arzew:* the port of Arzew is a rather important port complex, it is the largest oil port in Algeria. It is composed of 2 ports, the first one is the port of Arzew which is the old port whose construction dates back to the Roman period; it underwent several modifications and extension works. Today, it receives two types of goods, general goods and hydrocarbons. The second is the recently built (1975-1978) port of Bethioua (Arzew El-Djedid), which mainly supplies the liquefaction of natural gas and oil, crude oil and condensates. Both ports are managed by the Arzew Port Company (APC), which is attached to the Directorate of Captaincy (DC)[3,4].
- *Pollution prevention:* pollution prevention is an integral part of the daily missions of the Arzew port company. This mission is reflected in particular by traffic monitoring measures enabling permanent monitoring on the VHF marine radio (Very High Frequency), monitoring the port area, identifying incoming and outgoing ships, etc. As part of the fight against marine oil pollution, the directorate of captaincy has implemented an intervention plan, this plan consists of action planning aimed at organizing the response to a spill taking into account the collaboration with various services at the port level: civil protection, towing and mooring, environment, the Management and Operating company for marine oil Terminals (MOT), and the Territorial Grouping of Coast Guards (TGCG). To this end, the appearance of an oil spill in the port (due to an accidental or deliberate spill) is directly reported to the directorate of captaincy and to the territorial grouping of coast guards[5].

3.2 Proposed HMDCS-UV

A hierarchical organization chart is proposed for a new Central Unit (CU) for the Arzew port company. The hybrid system is intended as an improvement and extension of the proposed organization chart for the directorate of captaincy. Figure 2 shows the proposed reporting structure, which is composed of:

- *Master station:* it is composed of a general coordinator and several central unit officers. The general coordinator is represented by a laptop, guided by a human operator and / or an officer. This coordinator is responsible for several tasks such as launching and control of vehicles with / without driver used, use (processing) and storage of data in the database, regulation of the movement of the navigation of vehicles and coordination with other services (police and security, armament and maintenance).
- *Security, Armament and Maintenance (SAM) department:* it comprises the three departments mentioned concerning the three services, namely security, police and ship movements, armament and naval maintenance.

[3]"APC quality manual: quality management system ISO 9001: 2015, May 2018".

[4]"Arzew Port Company (APC)". https://www.arzewports.com/?pages=page&rub=13

[5] "Marine Pollution Control Report (APC)". Source: ISO 9001-2015, Marpol 72/73, Algerian Maritime Code, Environment Code, Regulations relating to the environment and ISO 9001-2008. Verification date May 10, 2019.

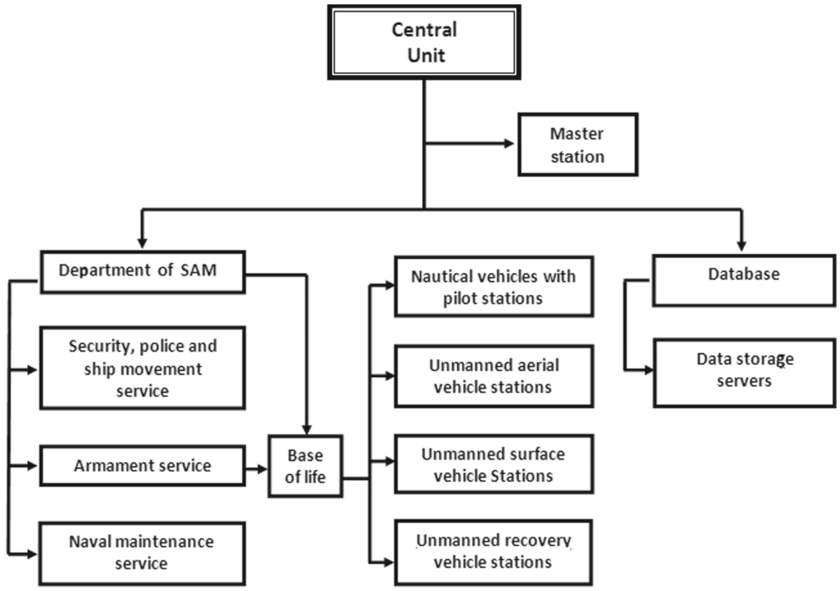


Fig. 2 Hybrid system central unit hierarchy

The SAM is responsible for various tasks such as security and safety of the central unit, preparation, checking and maintenance of marine equipment (surveillance vehicles or craft, cleaning vehicles, water or floating vehicles and others) in the living base.

– *Base of life:* it consists of four storage stations for watercraft with pilot, surveillance, cleaning and unmanned recovery vehicles represented in Fig. 2 by the Armaments Service.
– *Database:* it is represented by storage servers for all maritime space data.

i) *System architecture*

A hierarchical hybrid architecture is proposed in the HMDCS-UV (Fig. 3); it comprises a maritime force base, a central unit, a surveillance vehicle to monitor the maritime region, and a swarm of cleaning vehicles to clean a dirty zone (each swarm is guided by a leader). The maritime force represents the base of the TGCG, it includes the coastguards who work in collaboration with the port for surveillance and intervention in pollution response operations in the port. The central unit consists of a command room, a SAM department, a living base and a database. The main room includes a general coordinator who stores and consults the data in the database. The SAM department interacts with its agents (SAM_{agent}) in the living base by VHF marine radio, and with the control room by VHF and other means of communication such as WiFi. The master room interacts with the base of life and the surveillance vehicle via WiFi, and with the TGCG base via VHF and WiFi. This WiFi network also allows the exchange of messages between the surveillance vehicle and the leader of each swarm of cleaning vehicles.

ii) *Hierarchical decision of each entity*

The hierarchical decision on the proposed architecture, illustrated in Fig. 3, is made at the first level of the Maritime Force. This force includes the TGCG base, which represents the core memory of the port. It works in collaboration with the port to monitor it, intervene in pollution response operations and initiate requests to the central unit (i.e. the master's room). The latter is located on the second level, which includes a general coordinator. This coordinator represents the central memory of the central unit and has the highest decision for the execution of various tasks such as the launching and control of the used manned / unmanned vehicles that are located in the living base, the use and storage of data in the database and the coordination with SAM_{agent}. The surveillance UAV is responsible for a lower decision located at the third level. Its role is to monitor a maritime zone and supervise its swarms of clearance vehicles. These swarms are located on the fourth level and are composed of a lead vehicle and follower vehicles. Their objective is to carry out the cleaning operation of the dirty zone according to the energy availability of each member. Each leader has two necessary roles; it is responsible for the tasks of its followers, and also shares and cooperates with them in the cleaning action. Finally, each vehicle of the swarm has a local memory which constitutes the fifth level, and it can communicate with its neighbors.

iii) *Environmental modeling*

The working environment is described by the elements listed below [1]. Before defining these elements, Table 1 presents a description of some acronyms of the entities used and Table 2 shows the applied parameters in the proposed system.

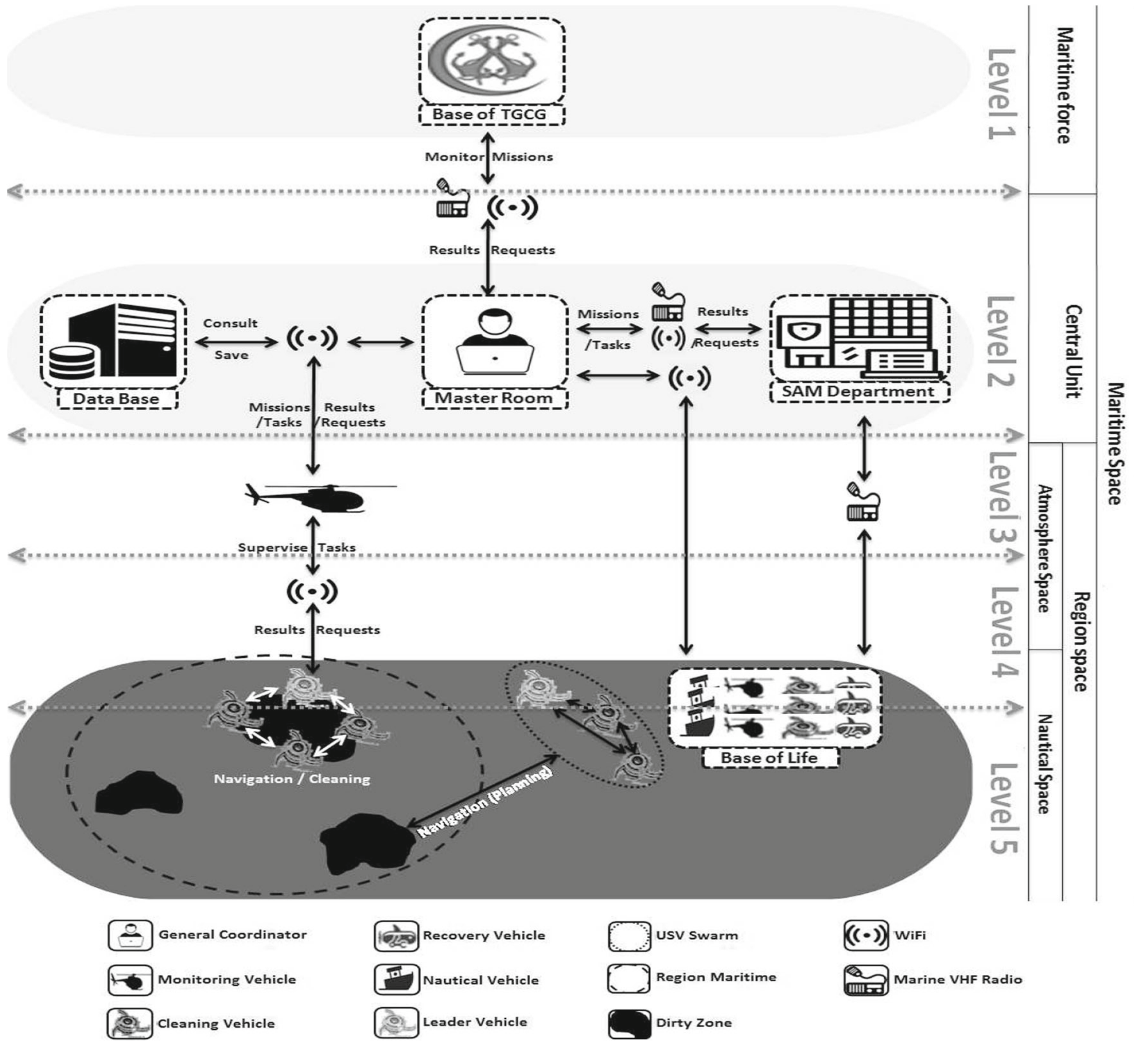


Fig. 3 Hierarchical hybrid architecture

– *Set of tasks:* these are the six high-level tasks used by the general coordinator, the SAM_{agent}, the surveillance and cleaning vehicles: Allocation task (t_a) of vehicles with / without pilot for the different regions and dirty zones; Preparation task (t_{pv}) of vehicles in the base of life; Monitoring task (t_{mr}) for a region r; Cleaning task (t_{cz}) of a dirty zone z; Cleaning supervision task (t_{sz}) of a zone z; Launch task (t_l) of the various previous tasks (allocation, preparation, monitoring, cleaning and cleaning supervision task).

Table 1 Descriptive table of acronyme

Acronyme	Description
TGCG	Territorial Grouping of Coast Guards
UAV_{mr}	Unmanned Aerial Vehicle for region monitoring
USV_{cz}	Unmanned Surface Vehicle for zone cleaning
$Leader_{cz}$	Cleaning leader (Unmanned Surface Vehicle)
$Vehicle_{rec}$	Unmanned recovery vehicle
SAM_{agent}	Security, Armament and Maintenance department agent
Sup_{mr}	Supervisor (Unmanned Aerial Vehicle)
$Vehicle_{nt}$	Nautical vehicle (boat)

– *Set of vehicles:* the different vehicles used in HMDCS-UV are the same vehicles used (UAV_{mr}, USV_{cz}, $Leader_{cz}$ and $Vehicle_{rec}$) in HA-UVC [1] except the $Vehicle_{nt}$ which is a floating object (nautical boat). It is a means of maritime transport used to embark port officers, pilots, customers, anti-pollution equipment, water supply and blasting.

– *Set of agents:* the different agents of HMDCS-UV are:

$General_{crd}$: a general coordinator is represented by a computer which contains coordination software, guided by a human operator. It is responsible for: the base of life, the data of the regions

Table 2 Descriptive table of parameters used in HMDCS-UV

Parameters	Description
$Threshold$	A fixed threshold which makes it possible to classify the coordinates of the zone according to the degrees of the cells "$List\_Degree_{cell}$"
$List_{ID}(Id_{USVcz})$	A list of identifiers of USV_{cz}
$List_{zone}(List\_Degree_{cell}, List\_Position_{cell}, Position_{zone})$	A list of coordinates of the dirty zone
$List_{threshold\_degreeZ}$	A list of the coordinates of the dirty zone compared to the dirt threshold compared to the degrees of dirt ($List\_Degree_{cell}$)
Pos_{Start}(x, y)	A starting position of USV_{cz} / UAV_{mr} from the base of life. This position is a Cartesian coordinate (x, y)
Pos_{End}(x, y)	A final position represents the arrival of USV_{cz} / UAV_{mr} at the region / dirty zone. This position is a Cartesian coordinate (x, y)
Pos_{Gol}(x, y)	A goal position means the next position of USV_{cz} / UAV_{mr} in its movement. This position is a Cartesian coordinate (x, y)
$List_{PosE}$(x, y)	A list of Cartesian coordinates (x, y) representing the final positions (to reach the dirty zone)
$List_{PosB}$(x, y)	A list of Cartesian coordinates (x, y) representing the border positions of the dirty zone
$List\_tabooPos(Id_{USV}, Pos_{Gol}$(x, y))	A list to save the positions already crossed by the USV_{cz}. It contains the USV identifier (Id_{USV}) which has already visited the position Pos_{Gol} where its Cartesian coordinate is the pair (x, y) in the grid
$List\_tabooCell(Id_{USV}, Cell(x, y))$	A list of dirty cells to memorize the cells already cleaned by USV_{cz}. It consists of the USV identifier (Id_{USV}) and its cleaned cell $Cell$(x, y). This cell represented by the Cartesian couple (x, y) in the grid
$List\_distCost(link_{ij}, Cost_{ij})$	A list of distances between positions marked as an energy cost. It is composed of a link $link_{ij}$ between the position i and the position j with its energy cost $Cost_{ij}$
$Parameters_{start-up(M)}$ (Id_{region}, $Position_{region}$, Pos_{Start}(x, y), Pos_{End}(x, y), $Map_{atmosphere\_space}$, $Map_{nautical\_space}$)	A list of parameters to start monitoring for each UAV_{mr} which represents the region's identifier and position, the start and end position, the map of the atmosphere space and the nautical space map before arriving at the region
$Parameters_{start-up(C)}$ (Id_{region}, Id_{zone}, Id_{Supmr}, $Id_{LeaderCZ}$, $Position_{zone}$, Pos_{Start}(x, y), Pos_{End}(x, y), $List_{PosE}$(x, y), $List_{PosB}$(x, y))	A list of cleaning startup parameters for each USV_{cz} which represents the identifier of its region, its zone, its supervisor, its leader and the position of the zone with the start and end positions, the list of end positions and the list of borders
$List_{characteristics}$(M) (Id_{region}, Id_{UAVmr}, $Cons_{ED1}$, $Cons_{ED2}$, $Cons_{EM}$)	List of UAV_{mr} characteristics: the identifier of a region, the identifier of a UAV (Id_{UAVmr}), the deplacement energy consumption from the base of life to the region ($Cons_{ED1}$), deplacement energy consumption in the region ($Cons_{ED2}$) and monitoring energy consumption ($Cons_{EM}$)
$List_{characteristics}$(C) (Id_{zone}, Id_{USVcz}, Id_{Supmr}, $Cons_{ED1}$, $Cons_{ED2}$, $Cons_{EC}$)	List of characteristics of USV_{cz}: the identifier of a zone (Id_{zone}), the identifier of a USV (Id_{USVcz}), the identifier of the supervisor (Id_{Supmr}), the deplacement energy consumption from the base of life to the dirty zone ($Cons_{ED1}$), the deplacement energy consumption in the dirty zone ($Cons_{ED2}$) and cleaning energy consumption ($Cons_{EC}$)

and dirty zones, the use (processing) and storage of data in the database, the launching of tasks / missions and the allocation / diffusion of tasks to vehicles and SAM_{agent}.

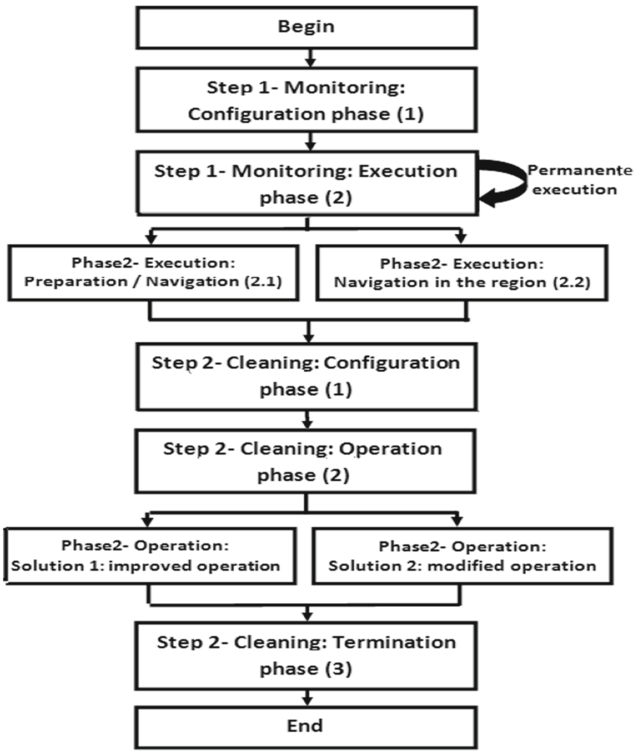


Fig. 4 Structure of the HMDCS-UV hybrid system

Sup_{mr}: a supervisor vehicle is the unmanned aerial vehicle (UAV_{mr}). It allows to: monitor regions, dirty zones and swarms of cleaning vehicles; supervise unmanned vehicles, request / inform the cleaning vehicle by tasks; return data and results to the general coordinator.

$Leader_{cz}$: it is an unmanned surface vehicle which has two roles: it is an intermediary between the supervisor Sup_{mr} and the swarm USV_{cz}, and cooperates with its followers in the cleaning operation.

SAM_{agent}: a SAM agent. This agent represents the Security, Armament and Maintenance (SAM) department for the execution of the various tasks requested by the $General_{crd}$.

– *Set of regions:* the monitored maritime space is divided into a maritime force base, a central unit and maritime regions. The region is made up of two subspaces; an atmospheric subspace where there are UAV_{mr} and a nautical subspace with swarms of USV_{cz}, $Vehicle_{rec}$, $Vehicle_{nt}$, dirty zones and can be the base of life.
– *Set of base of life:* it is an zone (which can be a boat, a ship, an island or a coast) to store a fixed number of UAV_{mr}, USV_{cz}, $Vehicle_{rec}$ and $Vehicle_{nt}$.
– *Set of database:* these are databases (servers) to store and save all the data and characteristics of the maritime space as well as the different vehicles with / without pilot used.
– *Set of dirty zones:* represents the dirty part where water pollution is found, for example hydraulic sheets (oil, gas, etc.) or plastic waste. This work focusses on oil pollution where the proposed

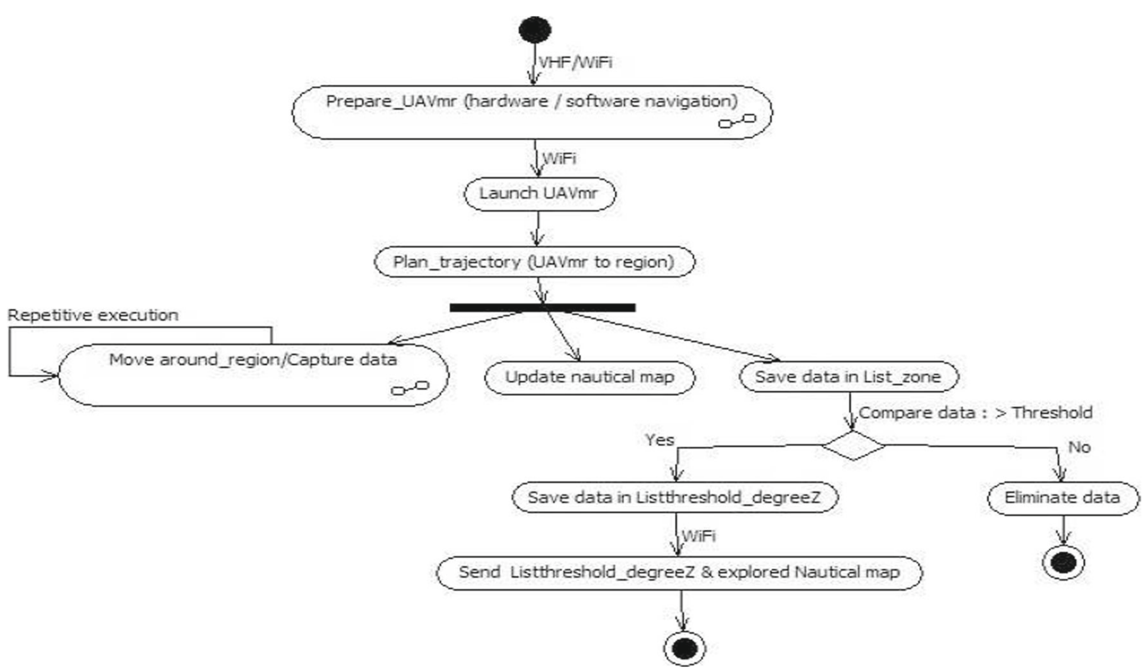


Fig. 5 Activity diagram for phase 2- Monitoring execution

measure is the degree of dirt for each zone: strong dirt, medium-strong dirt, medium dirt and weak dirt. Each zone is characterized by a list "$List_{zone}$" which delimits its borders by the coordinates; they are composed by the list of the degrees of dirt "$List_{Degreecell}$", the list of cell positions of this degree "$List_{Positioncell}$" and they are attached to a zone by "$Position_{zone}$".

iv) *Main steps of the proposed HMDCS-UV*
The proposed HMDCS-UV system is mainly based on two main steps: "monitoring" and "cleaning". Figure 4 illustrates the steps of the HMDCS-UV.
Step 1: Monitoring. This step presents the monitoring actions performed by each UAV_{mr}. These actions take place in two phases:

A) *Phase 1- Monitoring configuration:* this phase is executed when the $General_{crd}$ prepares the Monitoring drones according to the number of regions, by assigning a UAV_{mr} to each maritime region.
B) *Phase 2- Monitoring execution:* this phase is illustrated by the activity diagram (Fig. 5) and includes two sub-phases:
   – *Phase 2.1- Preparation / Navigation "base of life in the region":* the $General_{crd}$ requests via VHF the SAM_{agent} to prepare each UAV_{mr} to verify its hardware components. Then, the $General_{crd}$ sends the start parameters $Parameters_{start-up(M)}(Id_{region}, Position_{region}, Pos_{Start}$(x, y), Pos_{End}(x, y), $Map_{atmosphere\_space}$, $Map_{nautical\_space}$), an energy capacity (EC: Energetic Capacity) with a charged battery, a speed (S) and a $List_{characteristics}$(M) (Id_{region}, Id_{UAVmr}, $Cons_{ED1}$, $Cons_{ED2}$, $Cons_{EM}$) to each UAV_{mr} before starting via WiFi. Each UAV_{mr} is launched from the base of life of a starting position Pos_{Start}(x, y). It plans its path from its Pos_{Start}(x, y) to a final position Pos_{End}(x, y) using the explored atmosphere map $Map_{atmosphere\_space}$ (a discrete space of dimension 2, represented by a square grid G) and an algorithm proposed based on Cartesian coordinates "Planning towards the region" (Algorithm 1) with a rectilinear movement to reach its region (Fig. 6). Grid G is made up of identical square cells, containing free / occupied positions by UAV_{mr}. These positions construct a dynamic graph where the arcs are presented by connection links (edges) between the neighboring

Algorithm 1: Planning towards the region.

Data:
(X_s, Y_s): cartesian coordinate which represents the starting position of UAV;
(X_f, Y_f): cartesian coordinate representing the final position of UAV.
Result:
(X_g, Y_g): cartesian coordinate which represents the goal position of UAV;
$List_{PosUAV}$[SizeP]: list of cartesian coordinates traveled in a region by UAV where SizeP is the size of this list.

```
forall the ((int) r ∈ space_region) do
    X_g = X_s; Y_g = Y_s;
    add((X_g, y_g), List_PosUAV[SizeP]);
    while (Y_g != Y_f) do
        if ((X_g < X_f) && (Y_g < Y_f)) then
            X_g++; Y_g++;
        end
        else if ((X_g > X_f) && (Y_g < Y_f)) then
            X_g–;
            if (Y_g > 0) then
                Y_g–;
            end
        end
        else if ((X_g == X_f) && (Y_g < Y_f)) then
            Y_g++;
        end
        add((X_g, Y_g), List_PosUAV[SizeP]);
    end
end
```

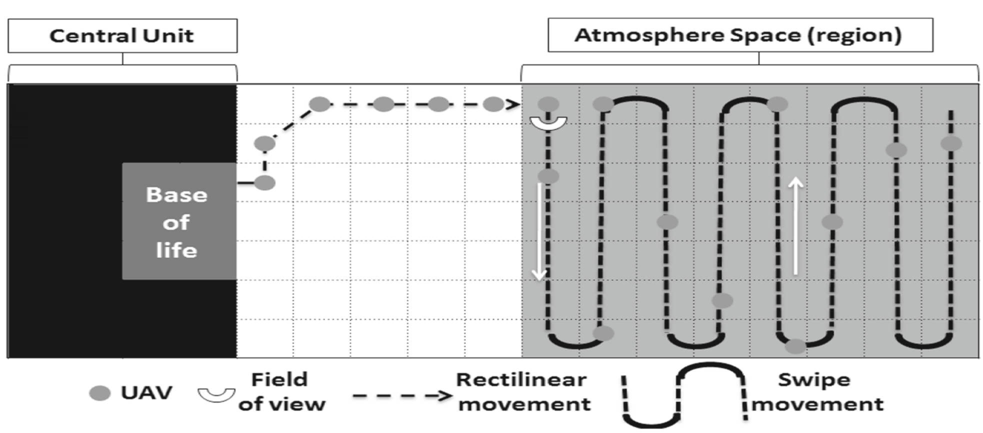


Fig. 6 UAV movements to perform

positions. The link represents the distance between the positions. This distance is represented by an energy cost that the UAV_{mr} has to consume in the displacement between the positions. The position has a maximum of eight links with neighboring positions j^{th}. After each movement between these positions, the UAV_{mr} saves the value of energy consumed in its $List_{characteristics}$(M).

– *Phase 2.2- Navigation in the region:* once the UAV_{mr} has arrived in its region, it can plan its movement. It therefore uses its sensory sensors (a camera and an ultrasonic sensor) and the $Map_{atmosphere\_space}$ card so that it can fly or move on the grid based on the Algorithm 2 "Modified Boustrophedon". This algorithm is inspired by the "Boustrophedon path" algorithms defined in [2] where four cases are proposed: The first case (A) is executed when the number of maritime space columns (nc-1) is odd and the number of the column of the goal or start position (Y_g) is odd; The second case (B) is executed when the number of maritime space column (nc-1) is even and the number of the goal position column (Y_g) is even; The third case (C) is executed when the number of maritime space columns (nc-1) is even and the number of the goal position column (Y_g) is odd; And finally, the last case (D) is realized when the number of columns of the maritime space (nc-1) is odd and the number of the column of the goal position is even. For example, the UAV_{mr} in the first case (A) crosses a new position (a goal position Y_g) which is the intersection between the second row and the first column until get to the final position which is the intersection between the last row and the last column, but the return path is to start by crossing a goal position which is the intersection between the last row and before the last column until arriving at end position which is the start position at the beginning. For each case, the UAV_{mr} scans repeatedly until the cleaning is finalized and at the same time saves the energy consumed in $List\ _{characteristics}$(M), as illustrated in Figs. 5 and 6.

A supervision and detection solution is proposed so that the UAV_{mr} can update its $Map_{nautical\_space}$ (lower level, in second square grid G 2D), where the UAV_{mr}

Algorithm 2: Modified boustrophedon.

```
Data:
(X_s, Y_s): cartesian coordinate which represents the starting position of UAV in a region; nl,
nc: number of row and column of grid;
(X_e, Y_e): cartesian coordinate which represents the end position of UAV before arriving at
the region; nl, nc: number of row and column of grid;
List_PosUAV [sizeP]: list of cartesian coordinates traveled by UAV in a region where sizeP
is the size of this list;
List_Node[sizeN]: list of maritime grid nodes where sizeN is the size of this list.
Result:
(X_g, Y_g): cartesian coordinate which represents the goal position of UAV in a region;
List_PosUAV.
forall the ((int) r ∈ space_region) do
    test_y ← even(Y_e);
    test_nc ← even(nc − 1);
    X_g = X_s; Y_g = Y_s; Y_v = 0;
    if (test_y == true && test_nc == true) then
        /* Case A: Y_g is odd and (nc − 1) is odd                        */
        /* (Instruction (A.1))                                            */
        test_y ← even(Y_g);
        if (Y_g < nc) then
            while (Y_g < nc) do
                /* (Instruction (A.2))                                    */
                if (test_y == true) then
                    add the positions (X_g < nl, X_g++) of the same column
                    Y_g in List_PosUAV;
                end
                /* (Instruction (A.3))                                    */
                else
                    add the positions (X_g > 0, X_g−) of the same column Y_g
                    in List_PosUAV;
                end
                Y_g++; test_y ← even(Y_g);
            end
        end
        /* (Instruction (A.4)                                             */
        Y_g = Y_V;
        if (Y_g != Y_s && Y_g ⩾ Y_s) then
            while (Y_g != Y_s) do
                Y_g−; test_y ← even(Y_g); (int) i = 0;
                /* (Instruction (A.5)                                     */
                if (test_y == true) then
                    add the positions (X_g < nl, X_g++) of the same column
                    Y_g in List_PosUAV and check if it is not equal to the
                    position (X_s, Y_s);
                end
                /* (Instruction (A.6))                                    */
                else
                    add the positions (X_g > 0, X_g−) of the same column Y_g
                    in List_PosUAV and check if it is not equal to the position
                    (X_s, Y_s);
                end
            end
        end
    end
    else if (test_y == false && test_nc == false) then
        /* Case B: Y_g is even and (nc − 1) is even                       */
        execute the same instructions (A.1), (A.2), (A.3) and (A.4);
        /* (Instruction (B.5))                                            */
        if (test_y == false) then
            add the positions (X_g < nl, X_g++) of the same column Y_g in
            List_PosUAV and check if it is not equal to the position (X_s, Y_s);
        end
        /* (Instruction (B.6))                                            */
        else
            add the positions (X_g > 0, X_g−) of the same column Y_g in
            List_PosUAV and check if it is not equal to the position (X_s, Y_s);
        end
    end
    else if (test_y == true && test_nc == false) then
        /* Case C: Y_g is odd and (nc − 1) is even                        */
        execute the same instruction (A.1);
        /* (Instruction (C.2)                                             */
        if (test_y == false) then
            add the positions (X_g < nl, X_g++) of the same column Y_g in
            List_PosUAV;
        end
        /* (Instruction (C.3))                                            */
        else
            add the positions (X_g > 0, X_g−) of the same column Y_g in
            List_PosUAV;
        end
        execute the same instructions (A.4), (A.5) and (A.6);
    end
    else if (test_y == false && test_nc == true) then
        /* Case D: Y_g is even and (nc − 1) is odd                        */
        execute the same instruction (A.1), (C.2), (C.3), (A.4), (B.5) and (B.6);
    end
end
```

uses an unsupervised classification method specific to image processing to process its captured data using the "swipe" movement. This proposed method is inspired by k-means clustering [3, 4]. The operation of this proposed solution is illustrated by the following phases:

a) Remote sensing; it designates the techniques allowing the acquisition of images to obtain remote information on an object, a surface or a phenomenon found on the surface of the earth, by means of a measuring instrument (for example, an airplane, a boat, a spacecraft, etc.) having no direct contact with the object studied [5]. When the UAV_{mr} detects a sudden change in the light intensity of the color of the water with its ultrasonic sensor in spatial resolution, it captures the complete image of the polluted zone and identifies its matrix points of landmark. There are different contour methods like gPb (globalized probability of boundary) [6], GraphCut road detection [7], etc. The contour method used in [8] is integrated into HMDCS-UV as a pretreatment phase, as shown in Fig. 7.

Then, the UAV_{mr} sends positions A, B, C and D of this zone directly to its $General_{crd}$ so that swarms of USV_{cz} from other regions avoid this zone in their movements (see Fig. 8a). These positions are represented in four Cartesian coordinates defined by a matrix: $\begin{bmatrix} (i_1, j_1) & (i_1, j_m) \\ (i_n, j_1) & (i_n, n_m) \end{bmatrix}$ where (i_1, j_1): the position (A) of intersection of the first line with the first column of the zone; (i_1, j_m): the position (B) of intersection of the first line with the last column of the zone; (i_n, j_1): the position (C) of intersection of the last line with the first column of the zone: and (i_n, j_m): the position (D) of intersection between the last row with the last column.

(b) Reception and segmentation of data (satellite / natural aerial image); remote sensing data is received as an image in the process of UAV_{mr}. This image is made up of many squares called pixels, as shown in the example of an oil slick in Fig. 8b.

Image segmentation can be performed by several color space methods [9]. However, the defined method is based on the use of the RGB (red, green, and blue) color space. In this method, the pixel (a bright spot) is calculated by averaging the RGB color encodings. Each pixel in an image has a radiometric value between 0 and 255.

c) Classification of data using the k-means algorithm; clustering is a process by which discrete objects with similar characteristics can be assigned to groups. In unsupervised classification, clustering methods aim at partitioning a set $X = \{x_1, x_2, ..., x_n\}$ of n objects described by p attributes into k classes also called clusters. The basic idea is that each object must be closer in terms of similarity to objects in the group to which it belongs than any object in another group. One of the most commonly used unsupervised classification algorithms is the k-means algorithm [3, 10].

d) Cluster validity measures; many criteria have been developed to determine the validity of clusters [9, 10] such as Dunn's index, Davies-Bouldin, F-ratio (WB), etc., all with a common goal to find the cluster that gives well separated compact clusters. Since the k-means method aims at minimizing the

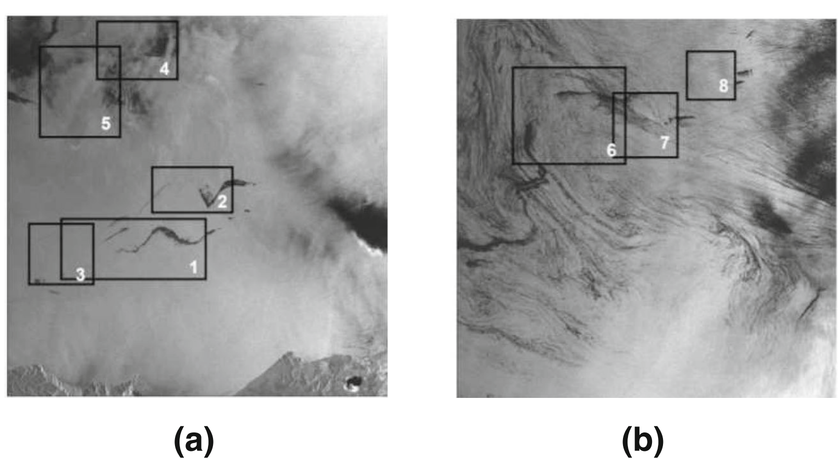


Fig. 7 **a** Selection of zones of interest in the El-Kala (eastern Algeria) and Arzew (western Algeria) images [8], **b** Result of contour detection in the El-Kala and Arzew images [8]

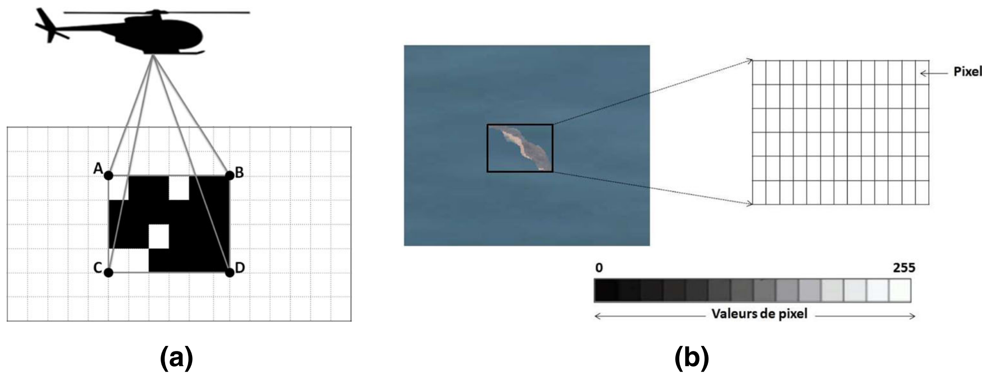


Fig. 8 **a** Spatial detection by UAV_{mr}, **b** Example of a natural image (an oil slick)

sum of the squares of the distances of all points from the center of their cluster, this should result in compact centers and thus compact clusters. To this end, the "F-ratio index validity method" can be applied.

e) K-means clustering; Algorithm 3 shows the K-means clustering applied based on the intra-cluster (1).

$$intra(k) = \sum_{i=1}^{k} \sum_{j=1}^{n} d(x_{ij}, c_i) \quad (1)$$

Where, c_i: is the center of the cluster i; n_i is the number of data points (pixels) in the cluster c_i; x_{ij} is the j^{th} data point of the cluster c_i; k is the number of clusters and d is a Euclidean distance between x_{ij} and c_i.

Algorithm 3: K-means clustering (X, k) [3].

Data:
$X = (x_1, x_2, ..., x_n)$: number of data points;
k: number of clusters.
Result:
$\{c_1, c_2, ..., c_k\}$: set of cluster centroids.
p = 0;
Randomly choose k objects and make them as initial centroids $(c_1^{(0)}, c_2^{(0)}, ..., c_k^{(0)})$;
repeat
  Assign each data point to the cluster with the nearest centroid;
  p ← p + 1;
  /* Centroid update */
  for *((int) j = 1 to k)* **do**
    Update the centroid $c_j^{(p)}$ of each cluster;
  end
until $c_j^{(p)} \approx c_j^{(p-1)}$ $\forall j = 1, 2, ..., k$;

The initialization of the k-means algorithm is based on the specified number of k clusters. These clusters contain the pixels of the segmented image. Then, the algorithm starts with an initial set of centers of gravity (or centroids) of clusters $\{c_1, c_2, ..., c_k\}$, chosen at random (an iterative process). In each iteration, each pixel of the image is assigned to the center of gravity of its nearest cluster. Then, the centroids of the cluster are recalculated. The center of gravity c of each cluster is calculated as the average of all pixels belonging to that cluster:

$$c_i = \frac{1}{n_i} \sum_{x_i \in cluster_i} x_i \quad (2)$$

The steps of the process are repeated until the centroids no longer move. The proposed algorithm allows to produce segmented images for 2 clusters up to $Kmax$, where $Kmax$ is an upper limit of the number of clusters, and then to calculate the validity measure to determine which cluster is the best cluster and, consequently, what is the optimal value of K. This work aims to produce 2 clusters and then to determine the dirty zone by the best cluster. This best cluster is not detected by the validity measure in a nautical image but is found by a proposed dirt level. This dirt level is calculated by the sum of the dirt levels found in the cluster. Thus, the cluster that has a maximum rate is defined as a dirty zone. Then, the UAV_{mr} compares the List_$Degree_{cell}$ of this detected cluster with a predefined "Threshold" to eliminate the non-dirty degrees. After the comparison, it saves the result in the $List_{threshold_{degreeZ}}$ list and sends it to $General_{crd}$ along with the explored and modified nautical chart.

Step 2: Cleaning. This step illustrates the cleaning actions in three phases:

A) *Phase 1- Cleaning configuration:* in this phase, the cleaning process is carried out as follows:

The $General_{crd}$ analyzes the data received from each UAV_{mr}. Then, the human operator or central unit officer prepares a report (PV) from the $List_{threshold\_degreeZ}$ to determine the position and zone of the dirty zone. Then, the central unit officer sends this report to the TGCG base via the SAM_{agent}, and requests the SAM_{agent} to send a nautical vehicle ($Vehicle_{nt}$) with agents to determine the cause of the pollution, check the climate, take samples and limit the pollution flow. Thus, the TGCG base sends the report to the local authorities (in this study, it focused on the city of Oran in Algeria), who represents the President of the local committee, and to the Regional Operational Centre for Surveillance and Rescue (ROCSR) of Oran. Then, the TGCG base asked the regional operational centre to send a nautical vehicle ($Vehicle_{nt}$) with an evaluation team to collaborate with the SAM_{agent}.

The assessment team and the SAM_{agent} found in the $Vehicle_{nt}$ record the information necessary for this pollution. Then the SAM_{agent} and the assessment team contact the SAM_{agent} from the SAM department and the TGCG base respectively, by VHF to send this recorded information in formation. This SAM_{agent} prepares an initial report based on the information received, and sends it to the central unit officer. The latter sends this IR1 to the TGCG base to prepare a second report also based on the information received from the evaluation team. This second report defines the position, zone, nature, cause of the dirty zone and the climatic report. The TGCG base sends the second report to the local authorities, the regional operational centre and the central unit. If the climatic conditions are not favorable, it does nothing and sends this information to the local authorities. Otherwise, the central unit officer launches a cleaning plan.

This plan begins when the central unit officer determines the number of USV_{cz} via the $General_{crd}$ for each dirty zone according to the $List_{threshold\_degreeZ}$. This action is described by Algorithm 4 for a single dirty zone in a region. Then, the $General_{crd}$ prepares a list of end positions ($List_{PosE}$(x, y)), illustrated by the dark gray color in Fig. 9, so that the swarm does not go beyond the zone borders, and a list of border positions ($List_{PosB}$(x, y)) in light gray color based on the positions received by the UAV_{mr}, so that the USV_{cz} is located near the zone cells using the regular cell decomposition.

Algorithm 4: Calculation of the number of USV_{cz}.

Data:
$List_{threshold\_degreeZ}[C]$: list of dirty cell degrees of a zone compared to $Threshold$, where C represents the length of this list;
M: represents the number of USV_{cz} in base of life (constant).
Result:
$List_{max}$[M]: list of maximum dirt levels found in $List_{threshold\_degreeZ}[C]$ received, where M represents the length of $List_{max}$;
$List_{sum}$[S]: list of the sums of the degrees of $List_{threshold\_degreeZ}[C]$ received, where S represents the length of $List_{max}$;
$List_{average}$[A]: list of average of $List_{threshold\_degreeZ}[C]$ received, where A represents the length of $List_{average}$;
Nbr_{CZ}[U]: number list of USV_{cz} for each dirty zone, where U represents the length of Nbr_{CZ}.

forall the *($List_{threshold\_degreeZ}$ received)* **do**
  calculate the max($List_{threshold\_degreeZ}$);
  save the max in $List_{max}$;
  calculate the sum($List_{threshold\_degreeZ}$);
  save the sum in $List_{sum}$;
  calculate the average($List_{sum}$, $List_{threshold\_degreeZ}$);
  save the average in $List_{average}$;
end
/* Calculate the number of USV_{CZ} for a zone in a region */
X = A;
forall the *((int) x = 0 to X)* **do**
  Nbr_{CZ}[x] ← $List_{average}$[a] * M / $List_{max}$[a];
end

In this phase, a proposed trajectory planning solution is applied by the USV_{cz} swarms, namely the Proposed-Cartesian Coordinate Algorithm (P-CCA). The P-CCA algorithm guides the USV_{cz} swarm to plan its trajectory to its dirty zone. This algorithm is already defined in [1] with adaptations and extensions for the HMDCS-UV, by adding the Pos_{End}(x, y) defined by (x_e, y_e) which represents the first position found in the $ListPosB$(x, y), and by replacing the pair of abscissas (i_1, i_2) in the dirty zone by the pair (x_e, y_e) (Algorithm 5).

When the $General_{crd}$ chooses the swarm of USV_{cz}, it sends the list of its $List_{ID}(Id_{USVcz})$ identifiers to its Sup_{mr}. Then, the $General_{crd}$ asks the SAM_{agent} to prepare the USV_{cz} forming the swarm in the base of life to check their hardware components, to prepare a $Leader_{cz}$ for each swarm with a high energy capacity compared to the other USV_{cz} and to drop them on the water by a crane. Then it transmits a set of parameters to the selected swarm, as shown in the sequence diagram in Fig. 10; and it sends only the start position Pos_{Start}(x, y) and end position Pos_{End}(x, y) to the $Leader_{cz}$. These fields are empty for the other USV_{cz} of the same swarm. Thus, the $General_{crd}$ launches via WiFi

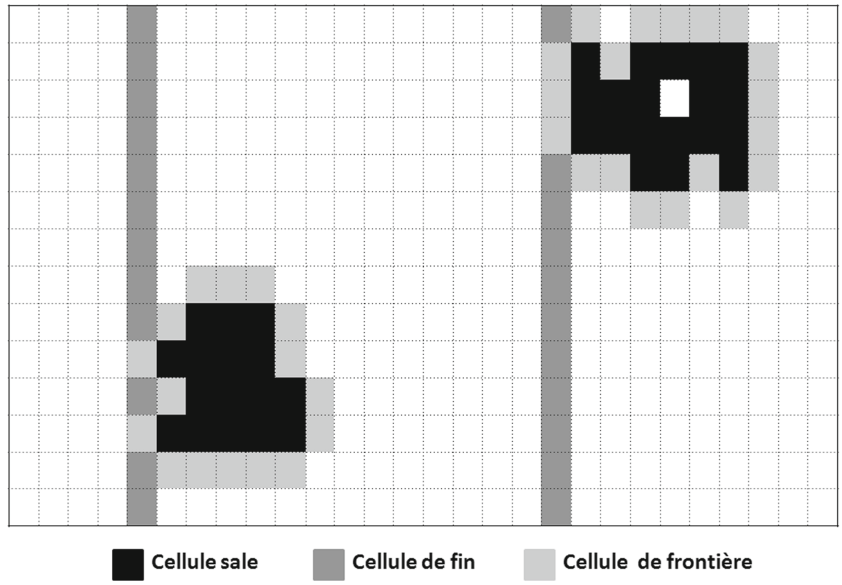


Fig. 9 End positions and boundaries of the dirty zone

the swarm with the execution of the P-CCA; and each $Leader_{cz}$ swarm plans its trajectory until it reaches the final position. The USV_{cz} of the swarm move together in the grid in a rectilinear form, and at the same time they change their positions in their $Parameters_{start-up(C)}$. Each USV_{cz} swarm follows its neighbor USV_{cz}.

B) *Phase 2- Cleaning operation:* this phase allows the USV_{cz} swarm to move into the dirty zone and clean it. The nautical zone is already discretized in a square grid (G). Each cell of the G can be dirty or clean. The USV_{cz} moves on G and perceives a detection zone R_s nearby (Fig. 8 [1]), and can clean the dirty cells. For this purpose, two solutions are proposed and applied by the USV_{cz} to move and clean the dirty zone.

– *Solution 1*: the first solution is an improvement of the proposition (Algorithm 2) defined in [1]. The novelty of this solution is that the USV_{cz} are located around the dirty zone with a step between each USV_{cz}. This solution eliminates collisions between the USV_{cz} and reduces the cleaning time. Then, a final position adjustment function is executed by the USV_{cz} to reach one of

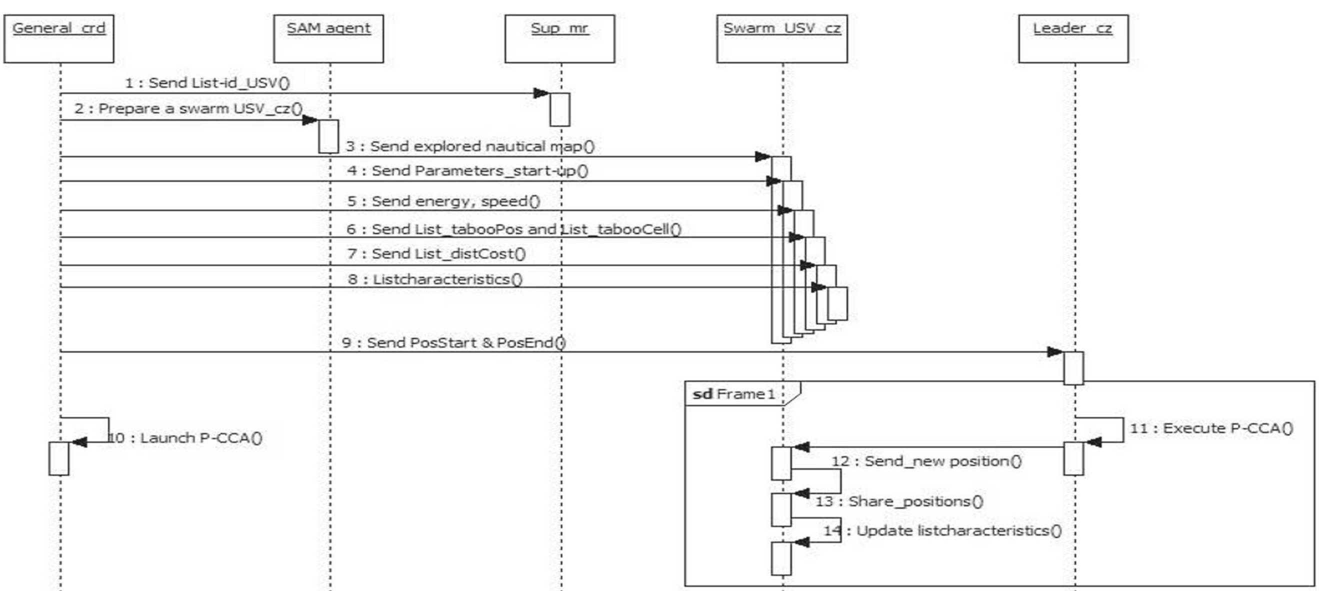


Fig. 10 Sequence diagram of "P-CCA-based triggering process for the swarm of USV_{cz}"

Algorithm 5: P-CCA-based trajectory planning.

Data:
$ListNvPos\_USV$[sizePU]: list of the new Cartesian coordinates (x_u, y_u) of each swarm USV of each dirty zone, where sizePU represents the length of $ListNvPos\_USV$;
$List_{DirtyPos}$[sizeD]: list of Cartesian coordinates (x_d, y_d) of the dirty degrees positions of each dirty zone, where sizeD represents the length of $List_{DirtyPos}$;
$List_{PosE}$[sizeP]: list of cartesian coordinates of the end position of the dirty zone, where sizeP represents the length of $List_{PosE}$;
$Pos_{End}(x_e, y_e)$: end position.
Result:
(x_g, y_g): Cartesian coordinate representing the new position of each swarm USV;
$ListNvPos\_USV$[sizePU].

fill_list($ListNvPos\_USV$[sizePU], (x_u, y_u), z);
forall the *((int) z ∈ space_region)* **do**
  while *(x_u != x_e)* **do**
    chercher_Pos_Fin((x_u, y_u), z, $List_{PosE}$[sizeP]);
    /* **Condition (1) if** $(x_u > x_e)$ **then**
      /* **Instructions (1.1)** */
      search_Node_Close((x_u, y_u), z, $List_{DirtyPos}$[sizeD], $ListNvPos\_USV$, (x_g, y_g));
      $x_u = x_g$; $y_u = y_g$;
      /* **Instructions (2.1)** */
      send((x_u, y_u), followers_USV);
      edit((x_u, y_u), $ListNvPos\_USV$[sizePU]);
    end
    */
    /* **Condition (2)** */
    else if *($x_u < x_e$)* **then**
      execute the same instructions (1.1) and (2.1);
    end
    /* **Condition (3)** */
    else
      while *(y_u != y_e)* **do**
        y_u++;
        execute the same instruction (2.1);
      end
    end
  end
end

the positions found in the $List_{PosB}$(x, y) list for each solution. This function is executed on the list of USV_{cz} ($List\_USV$) in two parts with two leaders, primary and secondary. The primary leader of the first part who arrived first at Pos_{End}(x, y), it starts searching for two successive positions found in $List_{PosB}$(x, y) with an upward move and modifies its last Pos_{End}(x, y) found in $Parameters_{start-up(C)}$, then it sends its old position to its neighbor, and the latter also sends its third position before the last one to its neighbor, and so on (Fig. 11a). Then the secondary leader who is the last USV_{cz} of the second part performs the same instructions with its followers but with a downward movement using the step concept. In the end, the primary leader is considered a leader for the whole swarm because it has consumed less energy than the secondary leader. Then, the $General_{crd}$ launches Algorithm 2 "Cleaning operation" on the USV_{cz}. Each USV_{cz} follows the algorithm so that it can avoid clean cells and select dirty cells to clean them.

– *Solution 2:* this solution is based on the method presented in [11], where the authors proposed to place USV_{cz} around the dirty zone in a circular pattern and each USV cleans a slice from its starting point of

Algorithm 6: Modified cleaning operation.

Data:
$List_{Crd\_USV}$[sizeU]: list of cartesian coordinates (x_u, y_u) of each USV where sizeU represents the length of $List_{Crd\_USV}$;
$ListCrd_{threshold-degreeZ}$[sizeD]: list of cartesian coordinates of dirty cells in a zone (x_d, y_d) where sizeD represents the length of this list;
Nb_zone: number of zones;
FI_min: first line index of a zone;
FI_max: last line index in a zone.
Result:
$(Newx_u, Newy_u)$: cartesian coordinates representing the new position of each USV;
$List_{TabooCelU}$[sizeC]: list which represents the memory of dirty cells already cleaned by USV.

while *((int) z < Nb_zone)* **do**
  (int) x_{d1} = (FI_min); (int) x_{d2} = (FI_max);
  (int) nb = (int)((FI_max - FI_min) / 2);
  (int) u1 = 0, u2 = (sizeU / 2);
  while *(x_{d1} < ((FI_min + nb) + 1))* **do**
    $y_d = y_u$;
    if *(u_x != x_{d1})* **then**
      $Newx_u = x_{d1}$; $Newy_u = y_d$;
      /* **(Instructions (A)** */
      modify(u1, $(Newx_u, Newy_u)$, $List_{Crd\_USV}$[sizeU]);
      delete((x_{d1}, y_d)), $ListCrd_{threshold-degreeZ}$[sizeD]);
      save(u1, $(Newx_u, Newy_u)$, $List_{TabooCelU}$[sizeC]);
    end
    u1++;
    if *(u == sizeU / 2)* **then**
      u1 = 0; x_{d1}= (x_{d1} + 1);
    end
  end
  while *(x_{d2} ⩾ ((FI_min + nb) + 1))* **do**
    $y_d = y_u$;
    if *(u_x != x_{d2})* **then**
      $Newx_u = x_{d1}$; $Newy_u = y_d$;
      execute the same instructions (A) with u2;
    end
    u2++;
    if *(u2 == sizeU)* **then**
      u2 = (sizeU / 2); x_{d2} = (x_{d2} -1);
    end
  end
  z++;
end

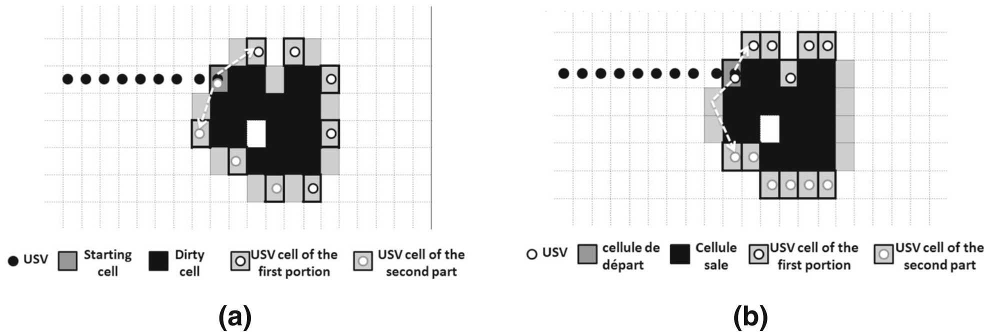


Fig. 11 Examples of initialization of a cleaning operation: **a** Improved operation, **b** Modified operation

position to the center of the circle. In addition, the improved operation proposes to locate half of the swarm of USV_{cz} at the top of the dirty zone and the rest at the bottom of the dirty zone in a straight line. Then, the dirty zone is divided into two parts. The first part is cleaned by the USV_{cz} at the top and the second by the USV_{cz} at the bottom of the zone, as shown in Fig. 11b. Next, the $General_{crd}$ runs Algorithm 6 on the USV_{cz}. Each USV_{cz} follows Algorithm 6 to avoid clean cells and to select dirty cells for cleaning.

C) *Phase 3- Cleaning termination:* this phase consists of identifying the termination of the cleaning process [1]. When the USV_{cz} finishes its cleaning task, it informs its $Leader_{cz}$ of the end of its mission by sending a message with the $List_{characteristics}$(C) via WiFi. Upon receipt of the message, the $Leader_{cz}$ informs the USV_{cz} to return to the base of life. When the USV_{cz} arrives at the life base, it informs its $Leader_{cz}$ by sending a message. Then, this $Leader_{cz}$ sends the $List_{characteristics}$(C) of the USV_{cz} to its Sup_{mr} for registration. Upon receipt of the message, the Sup_{mr} adds the information on the displacement energy consumption between the zone and the base of life to the $List_{characteristics}$(C) of the USV_{cz}, and sends it to the $General_{crd}$ for registration. When the cleaning operation is completed by all the USV_{cz}, the $General_{crd}$ prepares a initial final report containing all the information received from the Sup_{mr} and the SAM_{agent} compared to that of the USV_{cz} and other equipment used in this operation, and then sends it via WiFi to the TGCG base via the central unit. The TGCG prepares a second final report based on the second report and initial final report and the information received by the evaluation team, then sends it to the local authorities, the regional operational centre of the city of Oran and the central unit for registration.

4 Experimental Study

This section presents an example to simulate the operation of the HMDCS-UV. The evaluation of this simulation is based on the following measurements: displacement energy consumption plus monitoring energy consumption of UAV_{mr} in the region, total UAV_{mr} energy consumption, displacement energy consumption plus cleaning of the USV_{cz} swarm in the dirty zone, total energy consumption of the USV_{cz} swarm, total system energy consumption, speed of the USV_{cz} swarm, cleaning rate and efficiency of the USV_{cz} swarm. Furthermore, it is compared with a second modified proposal to study the behavior of the HMDCS-UV proposal and to analyze the results obtained from the simulation. The first proposal (P1) includes the first improved cleaning solution. However, the second proposal (P2) includes the second modified cleaning solution. A series of simulations were carried out using different parameters. Before starting the experiments, a real environment of the Gulf of Arzew (of the city of Oran, Algeria) with the positions of the most frequent hydraulic pollutions is presented, in addition to a virtual environment to apply the HMDCS-UV proposal.

4.1 Real Environment

The real environment of the Gulf of Arzew includes the port of Arzew, Béthioua and the offshore installations of Mers El Hadjadj (Fig. 12). The Gulf of Arzew is located on average on the Greenwich meridian and the latitude 36° N and extends from Cape Ivi (36° 37′ N - 000° 54′ W) to Cape Carbon (35° 54′ N - 000° 20′ W). These two capes form the boundaries of the Gulf of Arzew[3,4]. The port complex extends along the maritime fringe of the western part of Arzew Bay for about 22 km in latitude and 22 km in longitude (Fig. 12). The Arzew basin may have sources of marine pollution such as oil leaks from a ship loading station

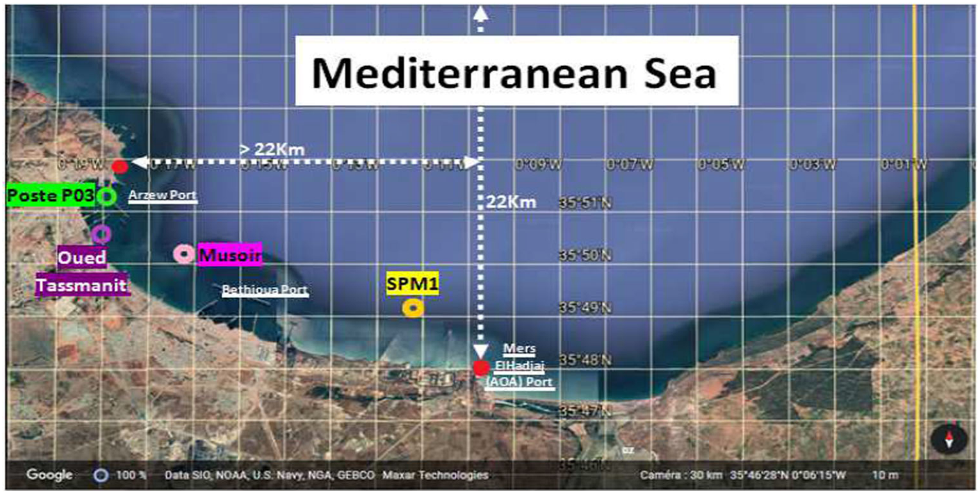


Fig. 12 Location of water pollution in the Arzew basin (Satellite image by Google Earth, March 2020)

pipe or the underwater pipeline of a liquid hydrocarbon; uncontrolled or accidental spills after loading a ship with hydraulic products, for example. The most frequent marine pollution in this basin is (Fig. 12)[6]:

"SMP 1, Single Point Mooring 1" Buoy Station, for loading crude oil located in the harbor;

"Musoir" pier end, it consists of three secondary points S1, S2 and S3 for loading and unloading crude oil, fuel oil and bitumen;

"Post P03", for loading crude oil and fuel oil;

"Oued Tassmanit", the presence of a black oil slick and H / C (hydrocarbon) traces along this Oued and the main axis towards the sea. This pollution is the result of leaks from industrial water discharged from one of the petroleum gas liquefaction and petroleum refining complexes.

4.2 Virtual Environment and Discretization

An example of a virtual environment is proposed, based on the maritime space of the Gulf of Arzew and the most frequent marine pollution presented. The virtual environment (maritime space) is represented by a central unit and a region "Region 1" (Fig. 13). The central unit is composed of a base of life, and the region constitutes two levels (maps) "atmosphere" and "nautical". The atmospheric level includes a UAV_{mr}. The nautical level includes a dirty zone "Z11" treated by a classification of an aerial natural image proposed below. This dirty zone is illustrated by a real crude oil slick in the vicinity of the SMP1 station in an exercise of the Tel-Bahr plan, proposed on 14 May 2019 at 06:00 local time when an oil tanker "ALPHA" loaded with crude oil collided with a vessel of type "RO / RO" during its exist maneuver of the Gulf of Arzew[7].

Before defining the discretization step, the "K-means clustering (X, k)" algorithm is applied to a reduced natural image of an oil spill captured in 2010. This slick of nearly 800 million liters of oil is spilled in the Gulf of Mexico[8] (see Fig. 14).

After reading the image, an RGB color space method is applied. A matrix is constructed based on this method which includes the coding of three colors (red, green and blue) forming the pixel, the Cartesian coordinate (x, y) of each pixel and the number of its cluster. This cluster number is filled in after performing K-means clustering, as shown in the example shown in Fig. 15a. After the RGB matrix, the K-means algorithm is executed to classify the reduced image, whose number of clusters is set to 2, and then the dirt rate of each constructed cluster is calculated. The cluster with the maximum rate represents the polluted zone, as shown in Fig. 15b and c of the previous example.

Next, the proposed environment is randomly defined and adds the data of this classification. Then, a grid is obtained where the grey cells represent the dirty zone and the white cells represent the borders of this zone. Figure 16 shows the discretization of the previous example.

4.3 Results

The application was produced with the open source Java language and the simulations were executed on a virtual machine (VM) in a heterogeneous Cloud, built with the middleware VMWare vCloud Suite 6, with a VM created

[6] "Report on marine pollution in the industrial area of Arzew (APC)". Presentation in 2014. http://www.arzewports.dz/

[7] "Demonstration exercise in combating marine pollution on the west seafront (Arzew 2019)". Proposed by the Ministry of National Defense, Command of the Naval Forces and the National Coast Guard Service. http://www.arzewports.dz/

[8] https://wol.jw.org/en/wol/d/r1/lp-e/102015330

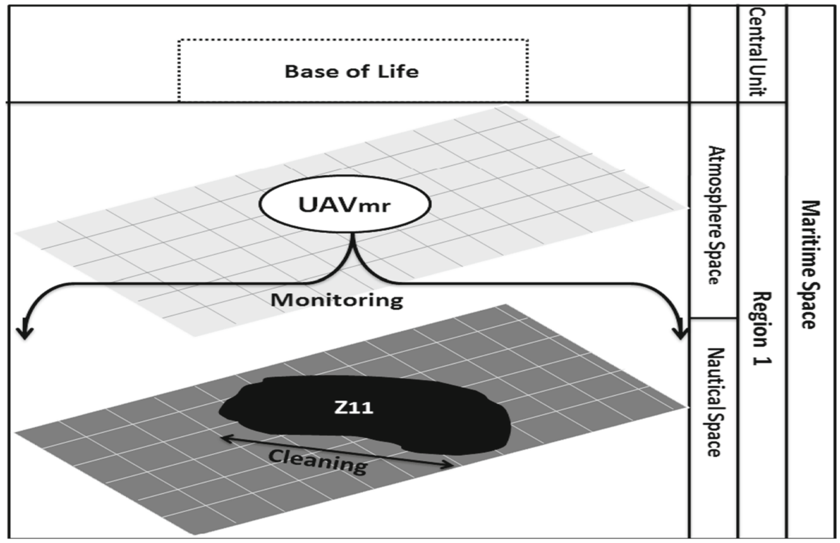


Fig. 13 Virtual environment (presentation of 3D level maps)

from 16G RAM and two Xeon (R) CPU E5-2620 v2 processors (@2.10GHz, @2.09GHz) running under the Windows 7 operating system. The proposed HMDCS-UV was implemented for a maritime region that includes a surveillance vehicle (UAV_{mr}), a swarm of cleaning vehicles (USV_{cz}) and a dirty zone. This zone does not present any solid obstacles that could prevent the vehicles from navigating and planning their trajectories. Based on a 78 × 48 cell metric map (Grid), the UAV_{mr} plans its trajectory from the base of life to its region according to the proposed Algorithm 1 “Planning toward region”.

When the UAV_{mr} arrives in its region, it begins to plan its trajectory and collect nautical level data using the “Modified Boustrophedon” algorithm and the unsupervised natural image classification method. When it detects a dirty zone with the contour detection method, it processes it with the k-means clustering method. The captured data is measured against a water color metric. These colors are distributed over four intervals:]0, 25],]25, 50],]50, 75] and]75, 100] in which they contain white (light dirt), light brown (medium dirt), dark brown (medium-strong dirt) and black (strong dirt) cells. The UAV_{mr} classifies this data (degrees) of the cells by comparing the predefined threshold value (equal to 26% of the degree of dirt) with the $Degree_{cel}$ of each cell. The number of USV_{cz} containing the swarm used in the simulations varies between 34, 47, 60, 73 and 86 USV_{cz} according to the enhanced cleaning algorithm (first proposal (P1)), and is equal to 136 USV_{cz} according to the modified cleaning algorithm (second proposal (P2)). The ranked images for each simulation are shown in Fig. 17. Each swarm of USV_{cz} plans its movement along the optimal trajectory (using the Proposed-Cartesian Coordinate Algorithm (P-CCA)) to reach its dirty zone and clean it. The P-CCA was executed by the $Leader_{cz}$ of each USV_{cz} swarm where each $Leader_{cz}$ of the classified zone starts from a starting position equal to (0, 0), and a final position equal to the first position in $List_{PosB}$(x, y) of each dirty zone. When the $Leader_{cz}$ finds a goal position, it shares it with it followers.

It is important to note that the trajectory of the UAV_{mr} or of USV_{cz} swarm obtained from the starting position (base of life) to the final position (the region or dirty zone) in the second proposition is the same as the one generated in the first proposition. The cost value between grid cells is

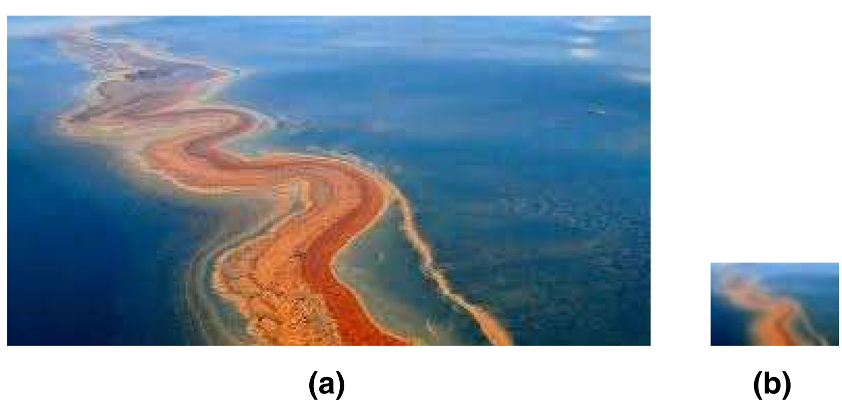
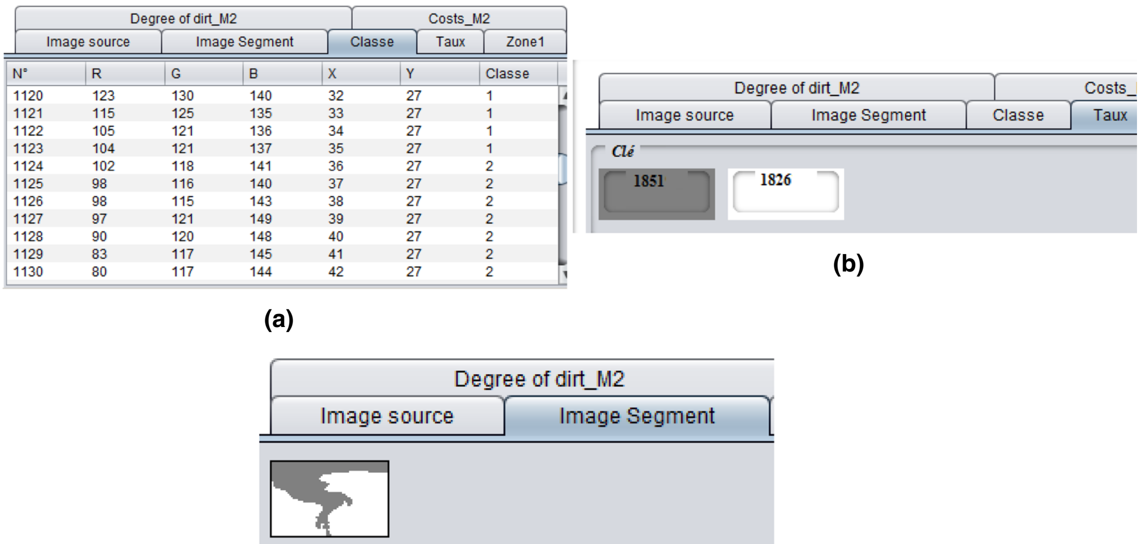


Fig. 14 Image presentation (zone): **a** Original image (310 × 159) and **b** Reduced image (68 × 44)



N°	R	G	B	X	Y	Classe
1120	123	130	140	32	27	1
1121	115	125	135	33	27	1
1122	105	121	136	34	27	1
1123	104	121	137	35	27	1
1124	102	118	141	36	27	2
1125	98	116	140	37	27	2
1126	98	115	143	38	27	2
1127	97	121	149	39	27	2
1128	90	120	148	40	27	2
1129	83	117	145	41	27	2
1130	80	117	144	42	27	2

(a)

(b)

(c)

Fig. 15 Example of a processed image: **a** RGB matrix of the classified image, **b** Dirt classification rate and **c** Classified image with k = 2

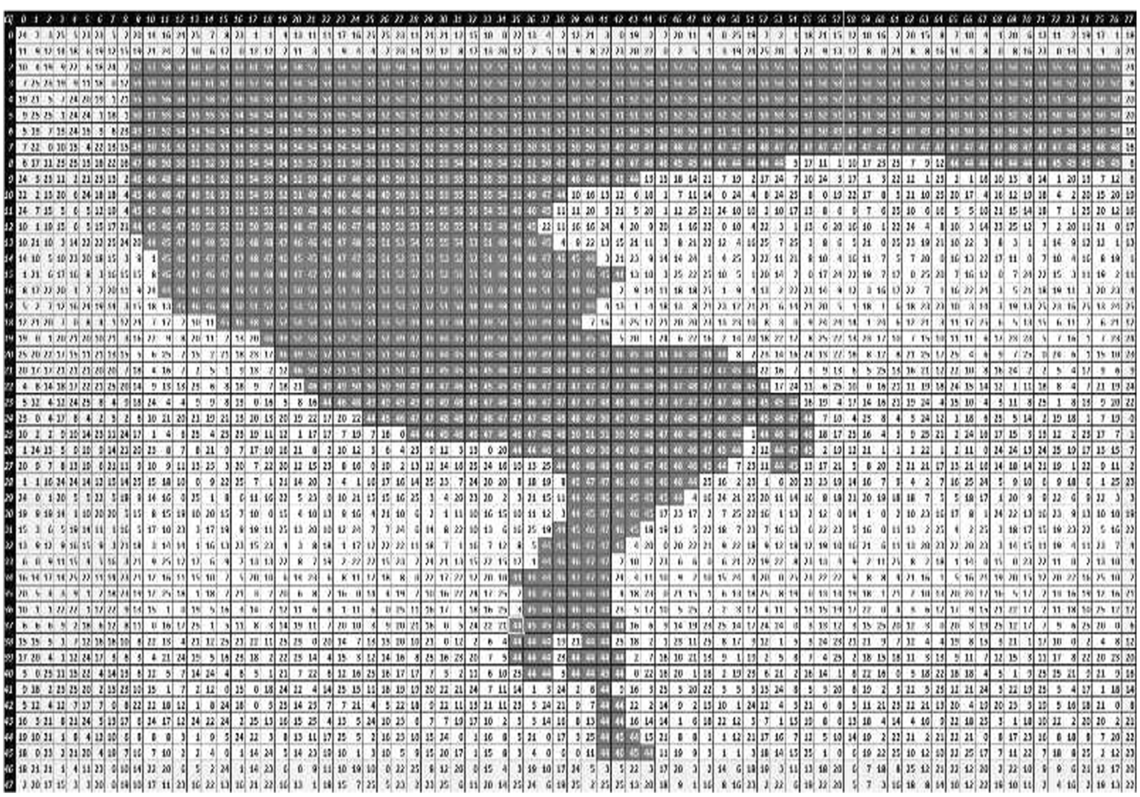


Fig. 16 Discrimination matrix for the nautical level of the environment

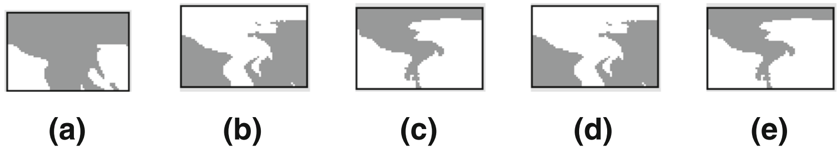


Fig. 17 Classified images: **a** (P1: 34 USV_{cz} and P2: 136 USV_{cz}), **b** (P1: 47 USV_{cz} and P2: 136 USV_{cz}), **c** (P1: 60 USV_{cz} and P2: 136 USV_{cz}), **d** (P1: 73 USV_{cz} and P2: 136 USV_{cz}), **e** (P1: 86 USV_{cz} and P2: 136 USV_{cz})

randomly generated between 5 and 10. The energy required by the USV_{cz} and UAV_{mr} to turn left / right from its current position to another position is 0.2%, 0.01%, and 0.05%, 0.001% respectively to continue directly. When the USV_{cz} swarm arrives in its zone, it starts cleaning the dirty cells. The energy required to monitor and capture data from a cell is 0.0001% for the second proposal using P-CCA (P2_PCCA). The proposed energy required to clean a black cell is 0.9%, for a medium-strength dirty cell is 0.5%, for a medium-strength dirty cell is 0.2%, and for a clear cell is 0.07%. And finally, the time required to clean a black, dark brown, light brown, light cell is 25, 15, 10, and 3 min respectively.

A formula was proposed to calculate the total system energy consumption (TEC) for each proposal using P-CCA.

$$TEC = TEC\_UAV + TEC\_SUSV \quad (3)$$
$$TEC\_UAV = DEC\_UAV + DMEC\_UAV \quad (4)$$

$$TEC\_SUSV = DEC\_USV + DCEC\_SUSV \quad (5)$$

Where, TEC_UAV: Total Energy Consumption of UAV_{mr}; DEC: Displacement Energy Consumption of UAV_{mr} from the base of life to the region; $DMEC\_UAV$: Displacement plus Monitoring Energy Consumption of UAV_{mr} in the region; TEC_SUSV: Total Energy Consumption of a swarm of USV_{cz} (SUSV); DEC_SUSV: Displacement Energy Consumption of SUSV from the base of life to the zone; $DCEC\_SUSV$: Displacement plus Cleaning Energy Consumption of SUSV in the zone. The monitoring consumption includes the data detection / processing consumption and the supervision consumption.

The formulas for the swarm speed metric of USV_{cz} ($Speed\_swarm$), the cleaning rate ($Cleaning\_rate$) and the efficiency of USV_{cz} swarm ($Swarm\_efficiency$) were calculated as follows:

$$Speed\_swarm = \frac{NbCells\_traversed/cleaned}{Total\_time} \quad (6)$$
$$Total\_time = Traveltime\_toZ + Traveltime\_inZ + Cleaningtime\_inZ \quad (7)$$

$$Traveltime\_toZ = (TotalNb\_traversedcells\_toZ \times Swarm\_size) \quad (8)$$

$$Traveltime\_inZ = (NbCells\_traversed/cleaned \times Traversetime\_cell) \quad (9)$$

$$\begin{aligned} Cleaningtime\_inZ &= (\textstyle\sum NbCells\_white \times Cleaningtime\_whitecell) \\ &+ (\textstyle\sum NbCells\_lightbrown \times Cleaningtime\_lightbrowncell) \\ &+ (\textstyle\sum NbCells\_darkbrown \times Cleaningtime\_darkbrowncell) \\ &+ (\textstyle\sum NbCells\_black \times Cleaningtime\_blackcell) \end{aligned} \quad (10)$$

$$Cleaning\_rate = \frac{NbCells\_traversed/cleaned}{TotalNb\_traversedcells} \quad (11)$$

$$Swarm\_efficiency = \frac{(Cleaning\_rate \times Swarm\_size)}{TotalSwarm\_size} \quad (12)$$

Where, $NbCells\_traversed/cleaned$: Number of cells traversed or cleaned in the dirty zone; $Total\_time$: Total time consumed by the swarm; $Traveltime\ \_toZ$: Travel time consumed between the base of life and the zone by the swarm; $Traveltime\_inZ$: Travel time consumed by the swarm in the dirty zone; $Cleaningtime\_inZ$: Cleaning time consumed by the swarm in the dirty zone; $TotalNb\_traversedcells\_toZ$: Total number of cells traversed to the zone by the swarm; $Swarm\_size$: Size of a swarm (or the number of USV_{cz} forming the swarm); $Traversetime\_cell$: Time needed to traverse a cell which is equal to 10 min; $NbCells\_white$: Number of clear cells cleaned by the swarm; $NbCells\_lightbrown$: Number of light brown cells cleaned by the swarm; $NbCells\_lightbrown$: Number of dark brown cells cleaned by the swarm; $NbCells\_black$: Number of black cells cleaned by the swarm; $Cleaning\ time\_whitecell$: Time needed to clean a white cell which is equal to 3 min; $Cleaningtime\_lightbrowncell$: Time needed to clean a light brown cell (10 min); $Cleaningtime\_darkbrowncell$: Time needed to clean a dark brown cell (15 min); $Cleaningtime\_blackcell$: Time needed to clean a black cell (25 min); $TotalNb\_traversedcells$: Total number of cells traversed by the swarm; $TotalSwarm\_size$: Total size of swarms used.

4.3.1 Simulation Results

This section presents the results of monitoring simulations in the region based on the unsupervised classification method for monitoring and detection, planning the swarm's trajectory to its dirty zone using "Proposed-Cartesian Coordinate Algorithm (P-CCA)" and cleaning the dirty zone (classified image). The results of these simulations give the curves of UAV_{mr} displacement and monitoring energy consumption in the region ($DMEC\_UAV$), total UAV_{mr} energy consumption (TEC_UAV), displacement energy consumption plus cleaning of the USV_{cz} swarm ($DCEC\_SUSV$) in the dirty zone, total energy consumption of the USV_{cz} swarm (TEC_SUSV), total system energy consumption (TEC), speed of the USV_{cz} swarm, cleaning rate ($Cleaning\_rate$) and the efficiency of the USV_{cz} swarm ($Swarm\_efficiency$) for both proposals.

- *Result of $DMEC\_UAV$:* Fig. 18a shows the results of the $DMEC$ of UAV_{mr} in its region. The first proposal was compared to the second in terms of displacement and monitoring. The $DMEC$ represents the energy consumption of the UAV_{mr} when it plans its trajectory in its region, detects the classified dirty zone and monitors the swarms in this zone that have already planned their trajectories on the basis of a P-CCA. It can be noted that the UAV_{mr} consumes less $DMEC$ in the first proposal (P1_PCCA) by moving and monitoring the 34, 47 USV_{cz} in each dirty zone compared to the second P2_PCCA by moving and monitoring the 136 USV_{cz} swarms in each zone. In addition, the $DMEC$ of P1_PCCA increases by monitoring one swarm of 60, 73 and 86 USV_{cz} relative to P2_PCCA. As a result, the UAV_{mr} consumes less energy to plan its trajectory in its region, detect the different proposed zones and monitor the swarm in P1_PCCA with a gain of + 35% compared to P2_PCCA.
- *Result of TEC_UAV:* Fig. 18b shows the result of the total energy consumption of UAV_{mr} in the two proposals (P1_PCCA and P2_PCCA). The TEC combines the displacement energy consumption of the UAV_{mr} from the base of life to its region and the $DMEC$. As a result, the UAV_{mr} in the first proposal gained +0.9% of TEC compared to the second proposal for displacement, detection and monitoring.
- *Result of $DCEC\_SUSV$:* Fig. 19a presents a simulation that evaluates the behavior of the USV_{cz} swarm in the $DCEC$ in the classified dirty zone for both proposals. It can be concluded that the displacement energy consumption increases much more than the cleaning energy consumption when the number of USV_{cz} containing the swarm increases. In addition, the $DCEC$ increases in P2_PCCA with the triple of 136 USV_{cz} compared to P1_PCCA with 34, 47, 60, 73 and 86 USV_{cz}. The HMDCS-UV proposal is generally better with 300 USV_{cz} than the P2_PCCA of 680 USV_{cz} in the $DCEC$ with a gain of +52%.
- *Result of TEC_SUSV:* the two previous results were used to calculate the TEC_SUSV based on the P-CCA for the two proposals. Figure 19b shows that the dark gray columns of the proposed dirty zone are greater in P2_PCCA than the white columns of TEC in P1_PCCA. As a result, the proposal P1_PCCA with 300 USV_{cz} consumes less energy with a gain of +53% compared to P2_PCCA with 680 USV_{cz} .
- *Result of TEC of the system:* this consumption combines the total energy consumption of UAV_{mr} (TEC_UAV) and the total energy consumption of USV_{cz} swarm (TEC_SUSV). Figure 19c shows that P1_PCCA consumes less TEC with a UAV_{mr} in the proposed region and a swarm of 34, 47 USV_{cz} in each zone compared to P2_PCCA with a UAV_{mr} in the region and 136 USV_{cz} in each swarm. In contrast, the white columns of P1_PCCA with a UAV_{mr} and a swarm of 60, 73 and 86 USV_{cz} are greater than the black columns of P2_PCCA. It can be noted that the increase in TEC with a swarm of 60, 73 and 86 USV_{cz} is due to the increased energy consumption of UAV_{mr} in displacement and monotoring these swarms to find and clean unoccupied and dirty cells. Therefore, the proposal with a UAV_{mr} and a swarm of 34, 47, 60, 73 and 86 USV_{cz} is better than the second proposal with a UAV_{mr} and a triple swarm of 136 USV_{cz} for a gain of + 1.23%.
- *Result of $Speed\_swarm$:* Fig. 20a shows the results of USV_{cz} swarm speed in the cleaning step for both proposals. It can be seen that the swarm speed increases in P1_PCCA when the number of USV_{cz} forming the swarm is equal to 34 and 47, then decreases when the swarm size is equal to 60 and 73 and remains almost stable at 86 USV_{cz} and in P2_PCCA. It can be noted that the speed of the swarm decreases when searching for a dirty and unoccupied cell, and when cleaning. As a result, swarms of USV_{cz} used in P1_PCCA clean dirty zones with an average increase in speed of +5.44% compared to P2_PCCA with 680 USV_{cz}.
- *Result of $Cleaning\_rate$:* Fig. 20b presents the results of the cleaning rate in three situations for the two proposals. The three situations are different depending on the swarm size used in each proposal. It is important to note that the black curve of P1_PCCA is above the gray curve of P2_PCCA in different situations, and the rate decreases as the swarm size increases. Moreover, the swarm lose a lot more energy in moving instead of in cleaning. Therefore, the proposal with 300

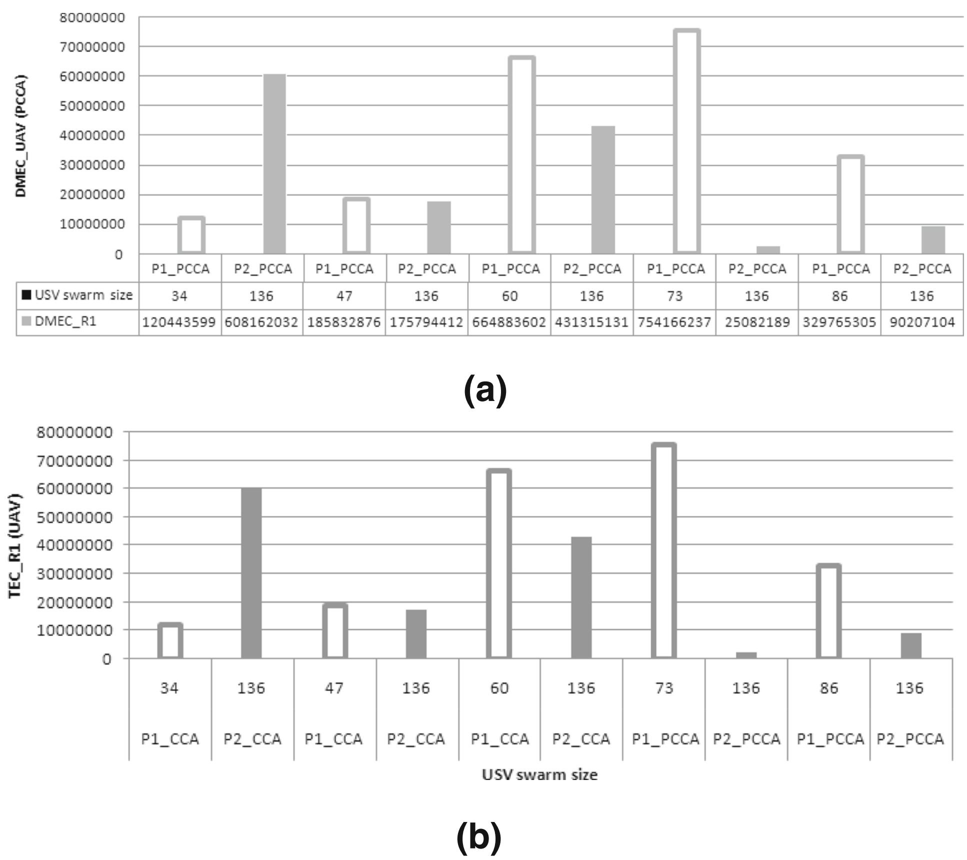


Fig. 18 Simulation results of: **a** $DMEC$ of UAV_{mr}, **b** TEC of UAV_{mr}

USV_{cz} used has a high cleaning rate with an average percentage of +67% compared to P2_PCCA with 680 USV_{cz}.

- *Result of Swarm_efficiency:* Fig. 20c shows the results of the efficiency of USV_{cz} swarm in cleaning for the two propositions. It can be noticed that the efficiency of swarm increases in P1_PCCA when the number of USV_{cz} forming the swarm increases and remains almost stable in P2_PCCA. The curve of P1_PCCA is above the gray curve of P2_PCCA in all three situations, and the cleaning efficiency increases as the swarm size increases. Consequently, the USV_{cz} swarms used in P1_PCCA cleans dirty zones with an average efficiency of +52% compared to P2_PCCA with 680 USV_{cz}.

5 Comparative Analysis

This section positions our HMDCS-UV solution in relation to the related work cited above (section 2). Each mentioned study has its own characteristics / parameters which differentiate it from the others. Tables 3 and 4 present a comparative study of these studies and the strengths of the proposed HA-UVC.

For example, the UAV_{mr} of the HMDCS-UV system moves to its region using a proposed Cartesian Coordinate Planning Algorithm (P-CCA), locates and detects the positions of the dirty zone in its region on the basis of its sensors (a camera and an ultrasonic sensor) and with a proposed solution. This solution applies an unsupervised classification method of natural / satellite image processing (K-means clustering) such as the UAV developed in [22] which makes it possible to monitor an oil platform using a camera and a LiDAR sensor for navigation and collision avoidance, and a GPS to map the spill area. The aerial mobile robot of [19], for example, is equipped with two cameras, a FLIR thermal imaging camera to locate and detect oil spills, and a digital camera to plan the route. In addition, the authors of [21] have introduced algorithms for the detection of oil spills using MIMO radar remote sensing integrated on a drone. Another sensor integrated into a commercial drone, called fluorosensor, was built in [28] for monitoring laser-induced fluorescence from the aquatic environment and also at night which is an obvious drawback compared to pulsed LiDAR systems with beach porting. Another sensor integrated into a commercial drone, called fluorosensor, was built in [28] for monitoring laser-induced fluorescence from the aquatic environment and also at night which is an obvious drawback compared to pulsed LiDAR systems with beach porting. Thus, the authors of

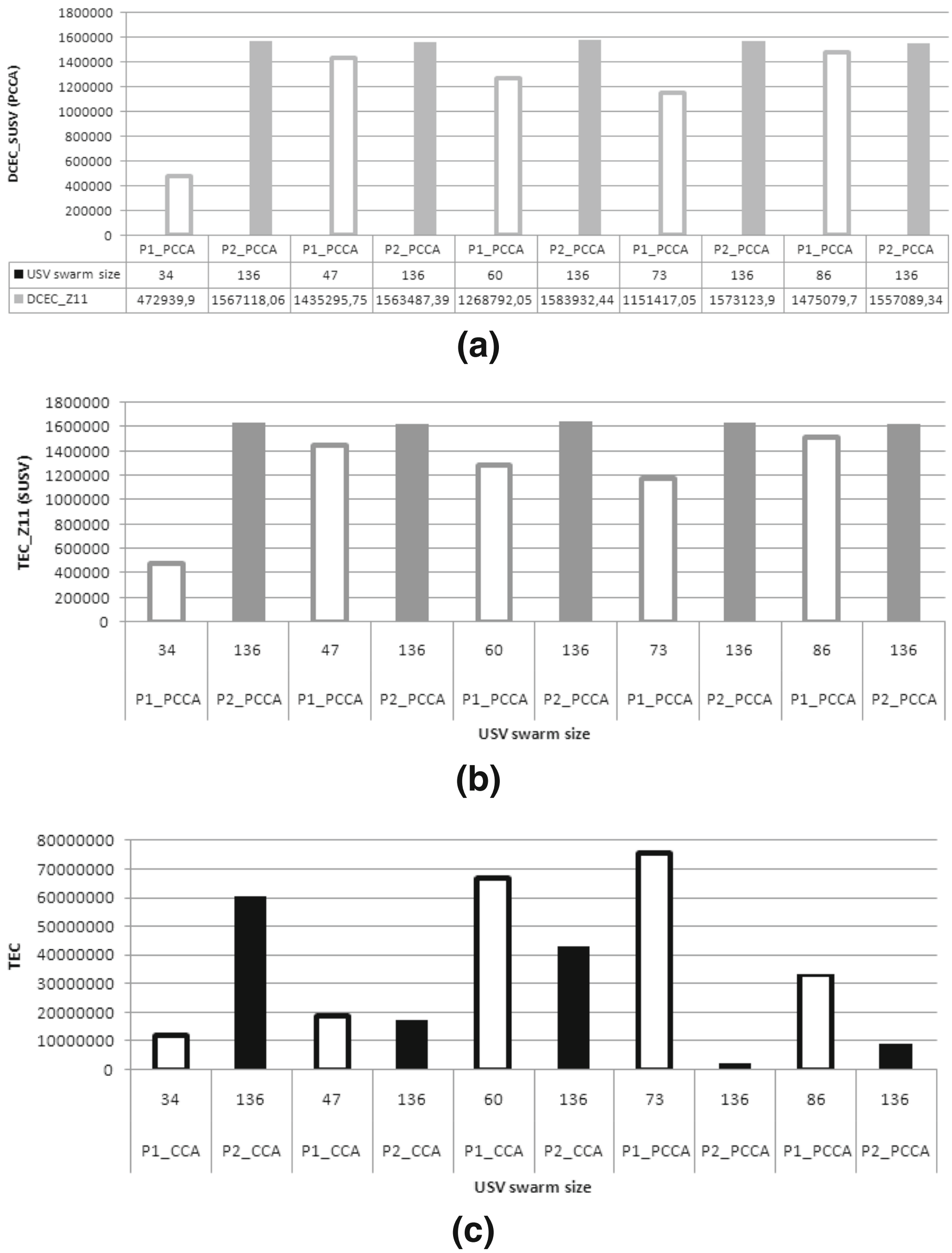


Fig. 19 Simulation results of: **a** $DCEC$ of USV_{cz} swarm, **b** TEC of USV_{cz} swarm, **c** TEC of the system

[29] proposed a new system architecture derived from the integration of a low cost laser array of pollutant detectors mounted on a UAV to identify the nature and amount of a release. The authors of [12] proposed an adaptive decision-making algorithm based on sensory information from autonomous vehicles that provides complete coverage of the search area for oil spill cleanup. On the other hand, the homogeneous drone swarm of the distributed system proposed in [15] makes it possible to monitor, locate and mark the perimeter of an oil spill and surround it, thus, avoid obstacles using the intensity of the signal (at a given frequency). In addition, the drone integrated in a system proposed in [31] makes it possible to monitor and detect in real time the temperature of the coronavirus (COVID-19) from the thermal image based on the IoT. The system of [15] allows a large-scale evolution like the HMDCS-UV, [1] and [25] and not in other works.

Subsequently, the UAV_{mr} sends the nautical chart to its $General_{crd}$ after discretization. Based on the data received, the $General_{crd}$ assigns the explored nautical chart to the swarm USV_{cz} to clean each dirty zone. For this purpose, two cleaning solutions are proposed in HMDCS-UV which were applied to the swarm USV_{cz} where each swarm can simultaneously move and clean dirty cells from its dirty

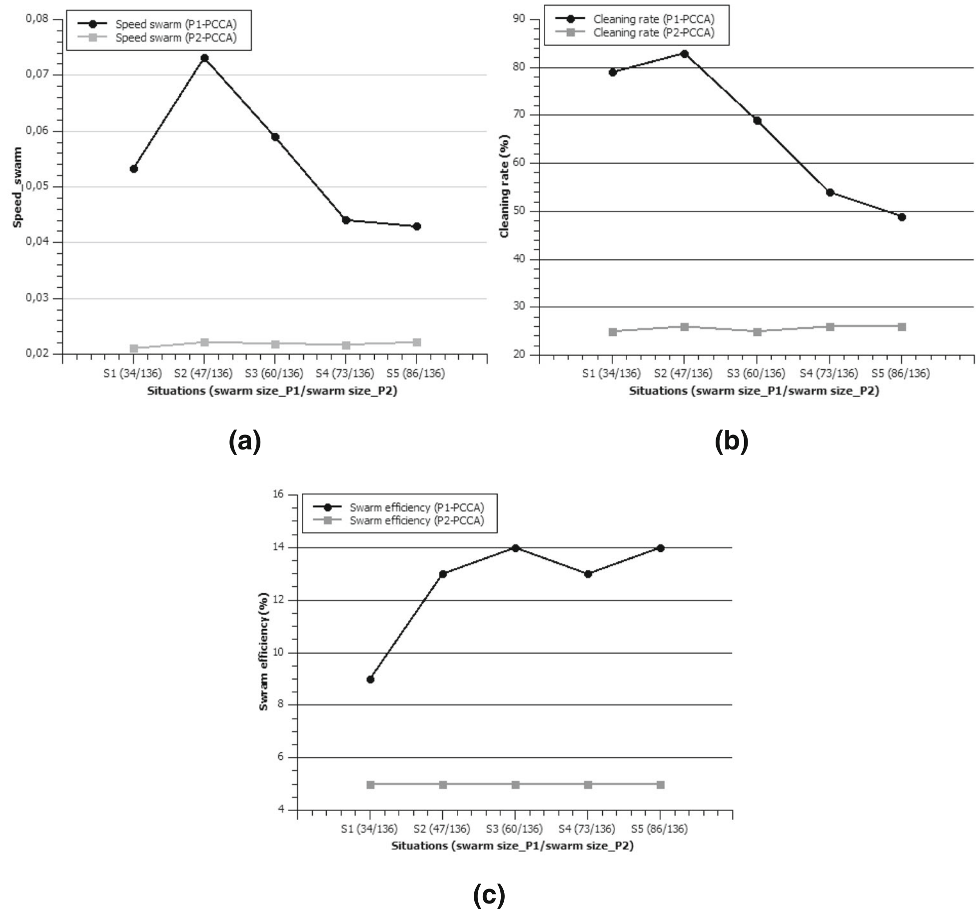


Fig. 20 Simulation results of: **a** $Speed\_swarm$, **b** $Cleaning\_rate$, **c** $Swarm\_efficiency$

zone without specifying how to remove the dirt (hydraulic spill). Thus, each USV_{cz} measures its quantity of energy by an energy threshold, then sends its information to the supervisor via its leader. On the other hand, the swarms of robots in the work [11] can place the barge with oil suction equipment and move it to another location to safely remove the oil. This barge is also used by the APC in the event of the presence of hydraulic pollution at the level of the port of Arzew, such as the [18] work which proposes an experimental system for the automatic towing operation of a dam in spill case using two ASVs and a ground station. Thus, a design of autonomous units (autonomous ships / drones) was developed in a research project EU-MOP [17] for the elimination of pollution of marine hydrocarbons, which are able to mitigate and eliminate the threat resulting from small and medium spills. Another design of a multi-robot system of autonomous aquatic vehicles was proposed in [16] for removing surface impurities, pumping oxygen into water, spraying chemicals, distributing food to appropriate places while measuring water quality. Thus, a method of controlling air pollution based on an air purifying drone system is presented in [26] to clean or reduce the amount of pollutants by spraying water and chemicals into the atmosphere. The works [17, 18] and [26] allow a feasibility in their systems and not in the HMDCS-UV.

6 Conclusion

This paper proposed a concept study of HMDCS-UV, allowing the monitoring, detection and cleaning of polluted marine zones, based on the cooperation of several semi-autonomous unmanned vehicles (a UAV_{mr} and a swarm of USV_{cz}) and their coordination by a general coordinator. Thus, this cooperation allows the swarms to clean the polluted zones from the map explored by the drone using a detection method based on a K-means clustering algorithm. In addition, an effective cleaning method for USV swarms is proposed so that they can move around and clean polluted zones in maritime regions (oceans / sea). This method is better in terms of energy by comparing it to another modified method which is inspired by the method proposed in Zahugi, E. M. H. and al. (2013) [11].

The proposed HMDCS-UV uses a UAV_{mr} for each maritime region and a swarm of USV_{cz} to clean up dirty zones.

Table 3 Comparison between related works

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[17]	Design of autonomous units (ships)	Mitigate and eliminate small and medium oil spills	Autonomous ship lawyers	Marine environment	BMS (Battery Management System) functions, mechanical or chemical countermeasures, zigzg movement	Ability to rotation of single-barrel units for different loading conditions and speeds, evaluation of the initial turning capacity on the turning circle, etc.	Design of the lawyer	-	√	-	√	√
[11]	New robotic swarm system	Cleaning of oil spills	Swarm of ship robots	Marine	Navigation, swarm operation, cleaning method	Consumption of money and time	Design robots (GPS module, communication, etc.)	√	-	-	-	√
[16]	Design of a multi-robot system of autonomous aquatic vehicles	Removing surface impurities, pumping oxygen into the water, spraying chemicals and distributing food in the appropriate places	Aquatic robot	Lakes	Recruitment algorithm, modules of the robot design	Water quality and dumping the collected waste on the ground	C++ using the Enki robotic library and Physical engine, MATLAB	√	-	-	√	√

Table 3 (continued)

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[12]	Multi-resolution navigation algorithm	Clean up oil spills	Autonomous vehicles	Port example (dynamic and uncertain)	Local and global navigation algorithm, decision-making, potential field method	Efficiency of the algorithm proposed with the benchmark algorithm: total time consumed to cover the entire search area, time consumed to clean up all oil spills, etc.	Pay / internship simulator	-	-	-	-	√
[15]	Distributed system	Monitor, recover and surround the perimeter of the spill	Swarm of homogeneous drones	Oil spill	Macroscopic and microscopic model, signal strength	Check the validity of the model presented in various cases	GNOME	√	-	√	-	√
[18]	Experimental autonomous control and coordination system	Automatic towing of a dam to contain a spill and recover it	ASV	Oil spill	Individual navigation control, towing method, training and detection of the spill	ASV lateral and heading errors during the full experiment, speed and distance between ASVs to notice a barrage	ASV design, multicontroller system, Google Maps, GPS track	-	√	-	√	√

Table 3 (continued)

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[30]	UAV-Based systems to monitor air pollution	Find the most polluted areas more accurately and cover the surrounding area	UAV	Areas	Pollution-driven UAV Control (PdUC) algorithm	Cumulative distribution function of the time spent at covering the complete area and relative error comparison between models mobility and at different times	Model of UAV, model mobility (Billiard, Spiral, and PdUC), OMNeT ++ simulator	-	-	-	√	√
[27]	Design andImplementation of an UAV-based platform for air pollution monitoring and source identification	Surveille air pollutants and tracking sources of contamination with a UAV and real-time processing using a meta-heuristic algorithm	Quadricopter	Area	Simulated annealing, monitoring and traking method	PM (Particulate Matter) indoor measurements	Model of UAV, proposed design of HMI, web application	-	-	-	√	-
[22]	Hydraulic spill monitoring, detection and cleaning system	Capturing images, navigating, avoiding obstacles, georeferencing positions and degrading the spill	UAV	Oil platform	Method spraying of chewing oil microbes on the ocean surface, georeferenced method and obstacle avoidance	Cost of cleaning consumed	UAV design	√	-	-	√	√

Table 3 (continued)

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[20]	Planning method of the trajectory planning control law	Detect, mitigate and circumvent the spread of an oil spill	UAV, ASV	Oil spill	Bioremediation method, field methods potential proposed to navigate and avoid the obstacle	Performance of the control law of each method, compare the trajectories of each method according to the distance	Gazebo	√	-	-	-	√
[21]	Algorithms for the detection of oil spills using MIMO	Detect the presence of oil based on a single and double frequency detector	Drone	Oil spill	Theoretical calculation of reflectivity, detection algorithms	Probability of detection	Monte Carlo in MATLAB	-	-	-	-	√

Table 4 Comparison between related works (the continuation)

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[25]	Locating and cleaning chemical leaks method based on a swarm and a bio-inspired exploration method	Follow pheromone trails to find the source of a leak and then decontaminate it by aggregation	Mona robots	Area	Motion Control, simulated model, interaction with environment	Impact of population size, speed of movement, intensity of environmental signals and group consistency, on the effectiveness of the cleaning operation	Simulated model of a Mona robot, Webots software	-	√	√	-	√
[28]	Monitoring system using a drone-based fluorosensor	Building a fluorosensor for fluorescence-based airborne remote sensing	Commercial drone (DJI M600 Pro)	Aquatic environment	Methods of measurements used fluorosensors	Intensity of the FIL spectrum of water in different cases	Model of drone, CW lidar system based on the Scheimpflug principle, high-power CW semiconductor laser	-	√	-	-	√

Table 4 (continued)

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[19]	Marine robotic system	Locate and quantify surface water pollution	UAV, marine robot	Lakes and ponds	Detection method based on a FLIR camera and digital camera for planning of the UAV path, fuzzy logic for movement, detecting the oil spectrum based on the ultrasonic sensor and neural network to differentiate the different types of pollution	Performance of the sensors used and the fuzzy logic method for navigation and detection	Design UAV and marine robot, MATLAB	√	-	-	√	√
[14]	New approach E-drones for large-scale elimination of air pollution	Autonomously monitor air quality, detect the presence of pollutants and measure its concentration, eliminate pollution	UAV	Air	Appropriate pollution reduction method at a specific altitude, air quality index (AQHI) map	AQHI data by e-drone	UAV design	√	-	-	-	√
[13]	New evolutionary algorithm based on elite groups for maximum ocean sampling	Planning the trajectory and taking samples	UMV	Ocean	Elite selection methods, simulated annealing algorithm, PSO, MILP	Cost, gain and convergence to find the best trajectory for the different methods	Matlab R2016b, Monte Carlo simulations	√	√	-	√	√

Table 4 (continued)

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[1]	Cooperative hybrid approach	Monitoring of a region based on UAV, cleaning of a dirty zone based on the swarm, trajectory planning, avoidance of static obstacles and fault tolerance	Semi-autonomous UAV, semi-autonomous USV swarm	Atmosphere / maritime (zone with high and low dirt)	Calculates the necessary number of USV, M-GA, P-CCA, cleaning algorithm, situation 1, 2 and 4	DEC, $D - CEC$, TEC of USV swarm	Implemented in java	√	-	√	-	-
[26]	An air pollution control method based on an air-purifying drone system	Clean up or reduce the amount of pollutants	Small drone	Areas close to industries, densely populated cities	Purification method	Smoke concentration (mg/m3) and time (Sec.). for Vehicle, Dhoop and Cigarette smoke	Model of Air purifier drone, chemicals used in air pollution control	-	√	-	-	√
[29]	A new system architecture derived from the integration of a laser network of pollutant detectors mounted on an UAV	Identify the nature and amount of a release	Rotatory wing UAS	Polluted areas	Detection (COLI , PID and IMS) and sampling (Arduino microcontroller) methods	Operation of detection and identification functions	COLI system, model of USV, SMITH detection instrument LCD3.3, PC, PID, IMS	√	-	-	√	√

Table 4 (continued)

Work / Criteria	System type	Main objective	Gear type	Environment	Algorithms / Functions	Metrics to measure	Simulator	Robustness	Feasibility	Scalability	Stability	Efficiency
[31]	A real-time coronavirus detection and monitoring system based on drone and IOT	Detect the coronavirus from the thermal image integrated in a drone based on the IoT	Drone	Air	Detection method based on a thermal camera controlled by virtual reality system	Surface temperature of the ground at a height above the ground	Detection system of UAV (GPS, thermal and optical camera, VR, Arduino IDE, etc), smartphones	-	-	-	-	√
HMDCS-UV	Hybrid monitoring, detection and cleaning system	Monitoring and detection of a dirty region based on UAV, cleaning based on the USV swarm, trajectory planning	Semi-autonomous UAV, semi-autonomous USV swarm	atmosphere / nautical space	Calculate the necessary number of USV, Planning towards the region, k-means Clustering, Modified Boustrephedon, P-CCA-based trajectory planning, modified cleaning operation	$DMEC\_UAV$, TEC_UAV, $DCEC\_SUSV$, TEC_SUSV, TEC, $Speed\_swarm$, $Cleaning\_rate$ and $Swarm\_efficiency$	Implemented in java in a heterogeneous Cloud	√	-	√	-	√

This new solution is seen as an improvement and extension of the HA-UVC solution [1] which defines a solution for fault tolerance and scaling up. The results of the simulations carried out are very encouraging, allowing a very significant reduction in energy consumption in monitoring, movement and cleaning. Additionally, the results indicate the effectiveness of deploying heterogeneous unmanned vehicles in a surveillance, detection and cleaning task in a partially known maritime environment.

On the basis of the good results obtained, in future work, the authors intend to study the influence of speed variations of unmanned vehicles in the control and cleaning phase, and also to implement an intelligent planning approach for swarms using other cooperative techniques avoiding obstacles such as line of sight, GPS intelligent buoy, fuzzy logic, etc. specific to predictable environments [24], and the methods such as RRT, virtual bodies and artificial potential, etc. for unpredictable environments [24]. They thus plan to improve the proposed process by using other physical properties of hydrocarbons such as density, viscosity or pour point [8]. Finally, they will think of collaborating with the port of Arzew (of the city of Oran, Algeria) in the future to develop a real drone for maritime surveillance such as the "Patroller [33]" system, and a drone for cleaning hydraulic layers such as the design of the "bio-Cleaner [32]".

Author Contributions Salima Bella and Ghalem Belalem conceived and designed the study. Salima Bella and Hichem Benfriha performed the experiments. Salima Bella, Ghalem Belalem and Assia Belbachir wrote the paper. Salima Bella, Ghalem Belalem, Assia Belbachir and Hichem Benfriha reviewed and edited the manuscript. All authors read and approved the manuscript.

Funding The research has not been funded.

Code Availability All code generated or used during the study are available from the corresponding author by request.

Declarations

Ethical Approval The authors declare that the rules of ethics (International, National) are respected in the conception and the realization of this work.

Consent to Participate Informed consent was obtained from all individual participants included in the study.

Consent to Publish Does not concern.

Competing Interests No conflict of interest exists in the submission of this manuscript, and the manuscript is approved by all authors for publication.

References

1. Bella, S., Belbachir, A., Belalem, G.: HA-UVC: Hybrid approach for unmanned vehicles cooperation. J. Multiagent Grid Syst. **16**(1), 1–45 (2020)
2. Manerikar, A., Das, D., Banerjee, P.: Optimal control for constrained coverage path planning. J. arXiv preprint, arXiv:1708.03055, pp 1–12 (2017)
3. Soheily-Khah, S.: Generalized K-Means Based Clustering for Temporal Data under Time Warp (Doctoral Dissertation). University of Grenoble-Alpes, France (2016)
4. Usman, B.: Satellite imagery land cover classification using k-means clustering algorithm: computer vision for environmental information extraction. Elixir J. Comput. Sci. Eng. **63**, 18671–18675 (2013)
5. Zou, Y., Li, G., Wang, S.: The fusion of satellite and unmanned aerial vehicle (UAV) imagery for improving classification performance. In: 2018 IEEE International Conference on Information and Automation (ICIA). Wuyishan, pp. 836–841 (2018)
6. Crommelinck, S., Bennett, R., Gerke, M., Yang, M.Y., Vosselman, G.: Contour detection for UAV-based cadastral mapping. Remote Sen. **9**(2), 1–13 (2017)
7. Zhou, H., Kong, H., Wei, L., Creighton, D., Nahavandi, S.: Efficient road detection and tracking for unmanned aerial vehicle. J. IEEE Trans. Intell. Transp. Syst. (T-ITS) **16**(1), 297–309 (2014)
8. Smara, Y., Bouchaib, S., Salvatori, L., Frate, F.D., Lichten-Neger, J.: Détection et identification des nappes d'hydrocarbures à partir des images satellitaires radar SAR. In: 3ème Symposium International des Hydrocarbures et de la Chimie. Ghardaia, Algérie, pp. 1-10 (2006) (In French)
9. Ray, S., Turi, R.: Determination of number of clusters in kmeans clustering and application in colour segmentation. In: 4th International Conference on Advances in Pattern Recognition and Digital Techniques. Calcutta, pp. 137–143 (1999)
10. El Asmar, S.: Contributions à la segmentation non supervisée d'images hyperspectrales: trois approches algébriques et géométriques (Doctoral dissertation). University of rochelle, France (2016)
11. Zahugi, E.M.H., Shanta, M.M., Prasad, T.V.: Oil spill cleaning up using swarm of robots. In: Proceedings of the Advances in Computing and Information Technology. Chennai, vol. **178**, 215–224 (2013)
12. Jin, X., Ray, A.: Navigation of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments. Int. J. Control **87**(4), 787–801 (2014)
13. Xiong, C., Lu, D., Zeng, Z., Lian, L., Yu, C.: Path planning of multiple unmanned marine vehicles for adaptive ocean sampling using elite group-based evolutionary algorithms. J. Intell. Robot. Syst. **99**, 875–889 (2020)
14. Rohi, G., Ofualagba, G.: Autonomous monitoring, analysis, and countering of air pollution using environmental drones. Heliyon **6**(1), 1–10 (2020)
15. Aznar, F., Sempere, M., Pujol, M., Rizo, R., Pujol, M.J.: Modelling oil-spill detection with swarm drones. In: Proc. Abstract Appl. Anal. **2014**, 1–14 (2014)
16. Agrawal, P., Bhattacharya, B.: Aquatic multi-robot system for lake cleaning. In: 16th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines. University of Technology Sydney. Australia, pp. 171–178 (2013)
17. Turan, O., Konovessis, D., Ventikos, N.P., Lemesle, P., Kakalis, N.M.P., Ventikos, Y.P.: Design and operation of small to medium scale oil-spill cleaning units. In: International Conference on Towing and Salvage of Disabled Tankers Safetow, UK (2007)
18. Giron-Sierra, J.M., Gheorghita, A.T., Angulo, G., Jimenez, J.F.: Preparing the automatic spill recovery by two unmanned boats towing a boom: Development with scale experiments. Ocean Eng. **95**, 23–33 (2015)
19. Mushtaq, Z., Ali, I., Shah, R., Sani, S.S., Su, S.F.: Detection, localization and analysis of oil spills in water through wireless thermal imaging and spectrometer based intelligent system. Wirel. Person. Commun. **111**(2), 679–698 (2020)

20. Pedrosa, D., Dias, A., Martins, A., Almeida, J., Silva, E.: Control-law for oil spill mitigation with an autonomous surface vehicle. In: 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO). Kobe, pp. 1–6 (2018)
21. Hammoud, B., Faour, G., Ayad, H., Ndagijimana, F., Jomaah, J.: Performance analysis of detector algorithms using drone-based radar systems for oil spill detection. In: Multidisciplinary Digital Publishing Institute Proceedings vol. **2**(7):370 (2018)
22. Dewi, A.A., Susanto, C.D., Ariansyah, M.R., Rahayu, M.Y.: Resheniye Drone, the Answer of Digital Oil Spill Recovery. In: Oil & Gas Seminar and Competition. Balikpapan (2018)
23. Vikulova, O.: The river runs red – catastrophic oil accident in the Russian Arctic. GreenPeace. https://www.greenpeace.org/international/story/43553/oil-spill-russian-arctic/. Accessed 10 June 2020
24. Panda, M., Das, B., Subudhi, B., Pati, B.B.: A comprehensive review of path planning algorithms for autonomous underwater vehicles. Int. J. Autom. Comput. **17**(3):321–352 (2020)
25. Amjadi, A.S., Raoufi, M., Turgut, A.E., Broughton, G., Krajník, T., Arvin, F.: Cooperative pollution source localization and cleanup with a bio-inspired swarm robot aggregation. arXiv:1907.09585 (2019)
26. Singh, A.P., Singh, S.P., Singh, A., Gupta, S., Raj, V., Kumar, S., Shankar, R., Kumar, B.: Application of Air Purifier Drone to Control Air Pollutants in Domestic and Industrial Areas. In: 2020 International Conference on Electrical and Electronics Engineering (ICE3). Gorakhpur, pp. 676–679 (2020)
27. Yungaicela-Naula, N.M., Garza-Castanón, L.E., Mendoza-Domınguez, A., Minchala-Avila, L.I., Garza-Elizondo, L.E.: Design and implementation of an uav-based platform for air pollution monitoring and source identification*. In: International Conference on Congreso Nacional de Control Automático. Monterrey, Nuevo León, pp. 288–293 (2017)
28. Duan, Z., Li, Y., Wang, J., Zhao, G., Svanberg, S.: Aquatic environment monitoring using a drone-based fluorosensor. Appl. Phys. B. **125**(6), 1–108 (2019)
29. Fumian, F., Di Giovanni, D., Martellucci, L., Rossi, R., Gaudio, P.: Application of miniaturized sensors to unmanned aerial systems, a new pathway for the survey of polluted areas: Preliminary results. Atmosphere **11**(5), 1–15 (2020)
30. Alvear, O., Zema, N.R., Natalizio, E., Calafate, C.T.: Using UAV-based systems to monitor air pollution in areas with poor accessibility. J. Adv. Transp. **2017**(8204353), 1–15 (2017)
31. Mohammed, M.N., Hazairin, N.A., Al-Zubaidi, S., AK, S., Mustapha, S., Yusuf, E.: Toward a novel design for coronavirus detection and diagnosis system using iot based drone technology. Int. J. Psych. Rehab. **24**(7), 2287–2295 (2020)
32. Turner, T.: Sea cleaning drones. Yankd Design. https://www.yankodesign.com/2012/05/21/sea-cleaning-drone/. Accessed 25 January 2017
33. Safran: Patroller tactical drone: a modular design to address the multiple needs of maritime surveillance. Safran. https://www.safran-electronics-defense.com/media/patroller-tactical-drone-modular-design-address-multiple-needs-maritime-surveillance-20181023. Accessed 25 December 2018

Salima Bella received PhD degree in computer science from Department of Computer Science, Faculty of Exact and Applied Sciences, University Oran 1 Ahmed Ben Bella (Algeria, 2020). She is a member of the EDCLC project entitled a "Ecosystéme Digital pour une nouvelle Chaîne Logistique à base de Cloud" (2018), a member of the mixed project: "Technologie et Santé: "IA-Respir"" (2021), at University Oran 1, Algeria. Her current research interests are distributed system, cloud computing, multi-agent systems, robotics mobile, decision support.

Ghalem Belalem graduated from Department of computer science, Faculty of exact and applied sciences, University of Oran1 Ahmed Ben Bella, Algeria, where he received PhD degree in computer science in 2007. His current research interests are distributed system; grid computing, cloud computing, replication, consistency, fault tolerance, resource management, economic models, energy consumption, Big data, IoT, mobile environment, images processing, Supply chain optimization, Decision support systems, High Performance Computing.

Assia Belbachir received a BEng in Computer Engineering (Algeria, 2006), a MSc in Computer Science from UCBL (France, 2007), and a PhD in AI from INPT, LAAS-CNRS (France, 2011). She worked as a postdoc at IFSTTAR (Paris, France) on the evaluation of advanced assistance driving systems. Then, she worked as a postdoc at ESIGELEC (Rouen, 2012) on the multi-robots cooperation. She is actually working at IPSA as a researcher on cooperative drones and also an associate professor at LIP6, team SMA. Her research interests lie in autonomous robots, cooperative systems, task planning and machine learning.

Hichem Benfriha is a computer science teacher in the Department of Technical Sciences, University of Mascara Mustapha Stambouli, Algeria. He is currently a Research Member of Laboratory of Computer Science of Oran. He is currently a PhD candidate in the Computer Science Department of Oran 1 University (Algeria). He received his Master of Science degree in 2012 from the same university. His research interests focus on CBR, data Mining, text mining, information extraction, information retrieval, natural language processing, machine learning and Multi-label classification areas.