Modeling Oil-Spill Detection with Multirotor Systems Based on Multi-Agent Systems

Pablo Chamoso, Alberto Pérez, Sara Rodríguez, Juan M. Corchado
Computer and Automation Control Department, Faculty of Science
University of Salamanca
Salamanca, Spain
{chamoso, alberto.pgarcia, srg, corchado}@usal.es

Mireia Sempere, Ramón Rizo, Fidel Aznar, Mar Pujol
Department of Computer Science and Artificial Intelligence
University of Alicante
Alicante, Spain
{mireia, rizo, fidel, mar.pujol@ua.es}@dccia.ua.es

***Abstract*—An increase in the frequency of oil spills in the sea, which adversely affects the maritime environment, led to the motivation for this article. Recent advances in technology have now made it possible to provide Unmanned Aerial Vehicles (UAV) systems with great processing capacity. They include all the features of a complete computer, while maintaining a really small size as required by this kind of system. This investigation takes advantage of the functionalities provided by existing Multi-Agent Systems (MAS) to accomplish tasks between UAVs. The article presents a case study that uses the capabilities to perform the detection of oil spills.**

Keywords—Unmanned Aerial Vehicle; Oil-Spill detection; Agents; Multi-Agent System; Virtual Organizations

I. INTRODUCTION

The inclusion of a current powerful Single-Board Computer (SBC) in UAV systems allows us to provide the system with the processing power required to perform tasks autonomously. Some tasks can be addressed in a distributed manner by a set of UAVs; if they are not very complex, they could be performed individually. However, there are other tasks that need to processed collaboratively (apart from additional benefits this system may entail) such as the case study developed in this article.

This way of approaching tasks from a series of collaborations between autonomous robots can be thought of as a system composed of multiple intelligent agents interacting with each other; that is, a multi-agent system. These systems allow the resolution of problems that would be difficult or impossible for an individual agent or monolithic system to solve, as in the presented case study.

These agents follow simple rules, allowing local interactions between agents lead to the emergence of a complex global behavior.

MAS have already been successfully tested in highly dynamic environments, which make it necessary for the model to evolve over time [2][6][15][9][17][16][7][8][4][26]. Some of these studies have been performed in a maritime environment [14][13][12], as with the case study of this article.

There are many tools that can provide a methodology for communication between each of the agents forming part of the system. This is the case of the Platform for Automatic Construction of Organizations of Intelligent Agents (PANGEA) [30], which allows the integration of agents in virtual organizations and provides a set of tools and rules for communication between agents.

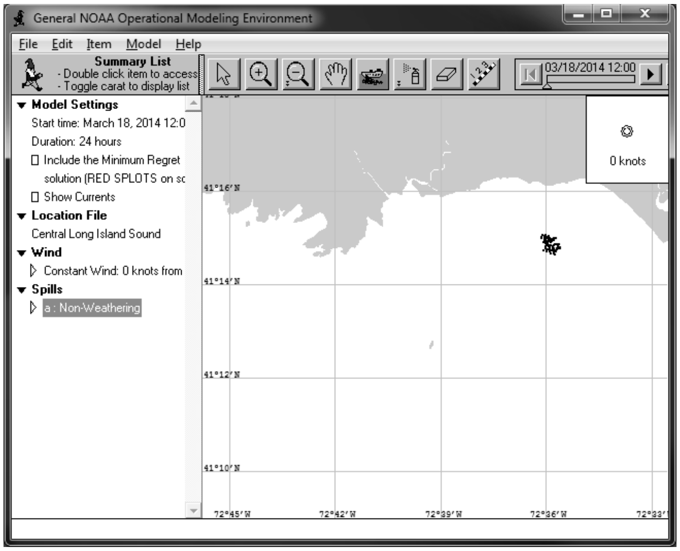
All of these tools and technologies allow addressing complex problems such as the focus of this study: the detection and monitoring of oil spills on maritime surfaces. The amount of spills produced in recent years has increased dramatically. This complex and damaging problem, which will be addressed in detail in the article, consists of certain variables that influence the way in which discharges are expanded. These variables can be sea currents, winds and even the gravitational force or surface tension of water.

Finally, once the case study has been introduced, it is necessary to keep talking about UAV systems. Of note is the multi-rotor type UAV, which is the most appropriate system to carry out this task. This is due to the great stability provided to the aircraft by the presence of several engines, and the ability to perform motions along all axes of space. Thanks to this, the trajectory of the movements is more efficient and controllable.

The article is structured as follows. First the pollutant dispersion model is explained. Subsequently, the background is described, including the case study and the integration of the system in terms of multi-agent systems and virtual organizations. A system overview is then provided for the reader to assimilate the structure of the entire system. Finally results and conclusions are given.

II. POLLUTANT DISPERSION MODEL

Modeling oil spills is not a simple task since there are many factors determining the trajectory of contaminants. These factors include ocean currents, winds, gravitational force or surface tension of water. There are several tools to model and simulate pollutant spills in the sea surface. In this study, the General NOAA Oil Modeling Environment (GNOME) [1][30] application was used. This tool allows predicting how winds, currents and other elements influence the oil spill, as in [29][21].



Summary List
- Double click item to access
- Toggle carat to display list

▼ Model Settings
Start time: March 18, 2014 12:0
Duration: 24 hours
☐ Include the Minimum Regret solution (RED SPLOTS on sc
☐ Show Currents
▼ Location File
Central Long Island Sound
▼ Wind
▷ Constant Wind: 0 knots from
▼ Spills
▷ a : Non-Weathering

Fig. 1. Main view a GNOME simulation with an oil spill over the map

One of the benefits of this tool is the possibility of using external files containing information on currents, tides, etc. If this information is added, GNOME uses it to predict the trajectory of the spill in a specified region. The output of the model consists of graphics, movies and data files for post-processing in a Geographic Information System (GIS).

The model used by this tool is general and applicable to trajectory problems. It is a two-dimensional in space Eulerian/Lagrangian model; however, vertically isolated layered systems can be modeled as well. Shoreline maps are inputs to the model, which allows any area to be modeled. The model automatically handles both hemispheres, and east or west longitude.

Random spreading (diffusion) is calculated by a simple random walk with a square unit probability. The random walk is based on the diffusion value D that represents the horizontal turbulent diffusion of the spill in the water. During a spill, the value is calibrated as based on over-flight data. In this way diffusion and spreading are treated as stochastic processes. Gravitational force and surface tension effects are ignored, as these are only important during the first moments of a spill. Complex representation of sub-grid diffusion and spreading effects are ignored.

The main diffusion equation used in GNOME is presented in [28], where D_x and D_y are the scalar diffusion coefficients in x and y directions and C represents the pollutant concentration:

$$\frac{\partial C}{\partial t} = D_x \cdot \frac{\partial^2 C}{\partial x^2} + D_y \cdot \frac{\partial^2 C}{\partial y^2}$$

GNOME simulates this diffusion with a random walk with any distribution, with the resulting diffusion coefficient being one-half the variance of the distribution of each step divided by the time-step:

$$D_x = \frac{1}{2} \cdot \frac{\sigma^2}{\Delta t}$$

where σ^2 is the variance of the distribution of diffused points and Δ_t is the time elapsed between time-steps.

Spilled substances are modeled as point masses (up to 10.000[4]) called Lagrangian Elements (LEs) or "splots" (from "spill dots"). Spills can be initialized as one-time or continuous releases, as point or line sources, or evenly spaced in a grid on the map for diagnostic purposes.

A simplistic 3-phase evaporation algorithm models evaporation in GNOME, where the pollutant is treated as a three-component substance with independent half-lives [22]:

$$X_{prob} = \left(P_1 \cdot \left(2^{-\frac{t_i}{H_1}} - 2^{\frac{t_{i-1}-2t_i}{H_1}}\right) + P_2 \cdot \left(2^{-\frac{t_i}{H_2}} - 2^{\frac{t_{i-1}-2t_i}{H_2}}\right) + P_3 \cdot \left(2^{-\frac{t_i}{H_3}} - 2^{\frac{t_{i-1}-2t_i}{H_3}}\right)\right) \cdot \left(P_1 \cdot 2^{-\frac{t_i}{H_1}} + P_2 \cdot 2^{-\frac{t_i}{H_2}} + P_3 \cdot 2^{-\frac{t_i}{H_3}}\right)^{-1}$$

where t_i and t_{i-1} are the elapsed times (age; in hours) at time-step i and the previous time-step $i-1$ respectively, since the LEs release. H_1, H_2 and H_3 are the half-lives of each constituent (in hours) for the pollutant; and P_1, P_2 and P_3 are the percentages of each constituent.

Once GNOME executes a simulation, the solution is produced in the form of a trajectory. GNOME provides two solutions to an oil spill scenario: a best estimate trajectory and an uncertainty trajectory. The best estimate solution shows the model result with all of the input data assumed to be correct. The uncertainty solution allows the model to predict other possible trajectories that are less likely to occur, but which may have higher associated risks. In this paper we will use the uncertainty solution of pollutant particles (represented by its LEs) for generating a continuous pollutant map. More details of this mathematical model can be obtained in [28].

This simulation is really helpful when setting the start and end point of the mission, i.e. the takeoff and landing point of the UAVs (also called 'Home'). This point will be on a specially designed platform, with appropriate safety distances, located on the deck of the ship that transports it.

Thus, from certain known values, it is possible to predict an area where the spill is likely located.

This ability to predict the location of the spill is really useful because various missions, each lasting about 30 minutes, will be made, and measures of previous missions will be used (as from the second mission) to set the input parameters in GNOME. This way part of the error, which may be produced by the simulator, is eliminated.

III. BACKGROUND

Case study

This paper presents a case study to test the efficiency of MAS and virtual organizations [24][23][25] applied to the detection and delimitation of oil spills on the sea using a small set of multi-rotor type UAVs.

Multi-rotor UAVs are very suitable for this specific task, due to their ability to maintain a stable position, as well as their vertical takeoff capability. Eight multi-rotors are used in this case study. They are each provided with an SBC for processing, a Wi-Fi module for communicating and a series of sensors to obtain information from the environment, such as a GPS module, a network camera, a gyroscope, a barometer, etc.

UAVs are transported on a ship, which is equipped with a takeoff platform, up to the area to inspect for oil spills. When the ship has reached that position it remains stationary until the end of the mission. The UAV group must then be able to detect and follow the pollutants.

During the beginning of the mission a number of configurations are made. Specifically, the UAV is aware of the number of members in the system at all times thanks to the functionality of the multi-agent platform. In addition, everyone is aware of their position in the takeoff queue (established according to order of appearance on the platform). This turn is used to automate the task of takeoff.

When a UAV reaches its operation height it starts searching for any trace of spill in the environment. Vision agent uses the IP camera and a vision algorithm to determine the existence of any oil slick in its visual field, as presented in [18][20]. Once an oil spill is detected, the spokesman agent communicates the UAV position and velocity to the next spokesman so that it can go to that position. Finally it will try to stay within its perimeter.

The mission starts by the organized takeoff of the UAVs when they are already functional. Between each takeoff there is a time interval up to 10 seconds. The height at which each member operates is different due to security reasons, which prevents collisions in a position (*x, y*). We also assume that due to the flight altitude (about 200m above sea level) the differences in the visual field caused by different flying altitudes are negligible.

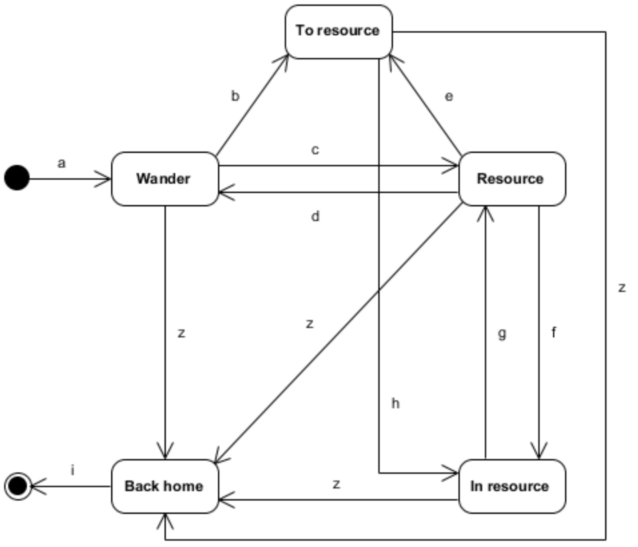


Fig. 2. Shows the Finite State Machine (FSM) governing the behavior of each robot.

The initial transition, ‘a’, to switch to 'Wander' state, refers to the takeoff, which is an automated and controlled action that allows the corresponding UAV to be positioned at the associated height, as previously described.

Once the UAV reaches that height it remains in its state of 'Wander'. This state refers to the situation where the UAV has no knowledge about the spill. At this point, all directions are equally valid and it moves randomly without repeating areas already completed. Although the slick changes its position, it does not do so fast enough to consider it in 30-minute flight. Therefore, the UAV is aware at all times of the points where it has already been in order to try to avoid repeating the same paths.

Multi-rotor velocity at a time *t*, is defined as:

$$\boldsymbol{v_w}(t) = \boldsymbol{v_w}(t-1) + \boldsymbol{ramd} \cdot \mu_1$$

where ***rand*** is a random uniform vector defined within the interval of the maximum and minimum velocity of the UAV, and μ_1 is the coefficient of variability on the current velocity. That is, with values close to 1, the robot will move in a totally random way.

Three types of transaction 'b', 'c' and 'z' may be produced from the ‘Wander’ state.

Transaction 'b' represents the passing of 'Wander' state to 'To resource' state. This transaction corresponds to the notice from another UAV system, indicating it has been positioned above the spill ('In resource' state) and should go to it.

If the UAV analyzes the image captured by its camera and finds out less than 80% of that image is actually oil spill, transaction **‘c’** is performed. That means the slick is being located and therefore the UAV should approach it until at least 80% of the image is covered.

Transaction 'z', starting from 'Wander' state, is produced for the same reason as other transactions referred to with a 'z' in Fig. 2. This transaction occurs when the calculation of remaining battery level exceeds the required threshold (along with a safety margin) to return to the landing point. This transaction makes the multi-rotor automatically change to 'Back home' state, regardless of the current state in use, forcing the UAV to get back to the takeoff/landing point (called home) with the help of his GPS.

The next state to be analyzed is 'To resource' state. As already mentioned, it will be accessed when a UAV of the system announces that it has been positioned on a spill. All other UAVs, regardless of their state, either 'Wander' or 'Resource', will get to an indicated point. Note that if the UAV is in 'In resource' state, it must be the one to send the warning. As a result, there is no transition from this state to ‘To resource’ state.

'Resource' state represents the moment in which a multi-rotor has detected, through optical analysis, part of the spill in the image (less than 80%). At that time, it tries to approach the spill with the intention of getting at least 80%, in order to change to the 'In resource' state by the transition 'f' as follows:

$$\boldsymbol{v} = \alpha_1 \times \boldsymbol{v_c} + \boldsymbol{v_o} + \alpha_3 \times \boldsymbol{v_r}$$

where $\alpha_1 + \alpha_2 + \alpha_3 = 1$ and these values define the intensity of each factor.

More specifically, v_c specifies the direction of the UAV. The area with the larger intensity resource average determines this direction:

$$\boldsymbol{v_c} = \gamma(S) \times \left\| \sum_{s \in S} ((\boldsymbol{pos}(s) - \boldsymbol{pos}(uav)) \cdot s) \right\|$$

where S represents the readings of the sensors detecting the resource at a given moment, $\boldsymbol{pos}(s)$ is a reading position vector and$\boldsymbol{pos}(uav)$ is the UAV vector position. We assume that the intensity of the resource is in the range [0,1], where 0 is the complete lack of resource and 1 is the unambiguous detection of it.

γ determines the direction of the velocity vector, depending on whether the UAV is outside or inside the resource. The aim of this variable is, therefore, to keep the UAV on the perimeter of the resource:

$$\gamma(S) = f(x) = \begin{cases} 1, & \frac{\sum_{s \in S} s}{|S|} \leq \eta \\ -1, & \frac{\sum_{s \in S} s}{|S|} > \eta \end{cases}$$

where η is a threshold that determines, from the quality of resource detected (0 means no resource and 1 maximum quality of resource), whether the UAV is located on the perimeter. If the main objective of the system is to cover the pollute slick, then $\gamma(S)$ will be defined as 1 for any set of readings S at a given time.

$\boldsymbol{v_o}$specifies an avoidance vector with respect to every UAV detected at a given moment.

$$\boldsymbol{v_o} = \left\| \sum_{i=1}^{|R|} (\boldsymbol{pos}(r_i) - \boldsymbol{pos}(uav)) \right\|$$

where R is the set of detected UAVs, $\boldsymbol{pos}(r_i)$ is the position of the detected UAV i and $\boldsymbol{pos}(uav)$ is the position of the current UAV.

Moreover, we will take into account the accuracy of the transmitted locations. There are several factors that could keep these locations from being optimal. We will include, therefore, a random component to model this uncertainty in the movement of the UAV: $\boldsymbol{v_r}(t) = \boldsymbol{v_r}(t-1) + \boldsymbol{rand} \cdot \mu_2$, where μ_2 is the coefficient of variability on the velocity.

Finally, the last status is '**In resource**', meaning the UAV is located inside the spill and, therefore, the borders of the resource are not detected. We assume that UAVs will develop a random wander behavior until they find water again because there is no other information about which direction is better to follow.

$$\boldsymbol{v_s}(t) = \boldsymbol{v_s}(t-1) + \boldsymbol{rand} \cdot \mu_3$$

where μ_3 is the coefficient of variability on the velocity.

UAV and multi-rotors

The advantages of intelligent approaches such as the conjunction of artificial vision and the use of Unmanned Aerial Vehicles (UAVs) have been recently emerging.

One of the most popular technological advances in recent years is the multi-rotor or multi-copter, a type of UAV capable, among other characteristics, of aerial filming. The use of this system to obtain an aerial video with sufficient quality is a tool that, in combination with the cited image analysis, allows detecting the oil slick.

Structurally speaking, multi-rotors offer such advantageous features as vertical takeoff, which allows taking off from any surface, for example from a boat, which is the case with the present case study. Another highlighting feature is their ability to maintain a stable position, in contrast to other types of UAV such as an airplane. In addition, their mechanism is much simpler than traditional helicopters and provides greater stability.

Multi-Agent System

Another important quality to take into account is the possibility of providing the UAV with a Wi-Fi module. By doing so, Wi-Fi equipped multi-rotor systems can be integrated into a multi-rotor network. In this case, they can communicate to each other in order to set various collaborative behaviors to achieve a common goal, which fits within the description of multi-agent system.

There are many tools to facilitate the development of these multi-agent systems, such as PANGEA [30] or JADE (Java Agent DEvelopment framework) [30]. The former was chosen for the development of the present system due to its simplicity in the communication protocol between the parties, as well as the fact that it is based on Internet Relay Chat (IRC) [11].

Furthermore, PANGEA offers a feature by which its agents can be programmed with any programming language and run on any platform with network connectivity. This feature makes it possible to program the required agents on the multi-rotor onboard computer and connect them together.

The connection between various units makes it possible to address the underlying problem in a better way than by simply using individual units.

Apart from the creation of agents and interaction between them, PANGEA offers the ability to create virtual organizations using key concepts such as norms and roles [5]. This capability will be used to design the system as detailed in the following section.

Virtual organizations

Virtual organizations of agents are made up of a set of agents that need to coordinate their resources and services. For the current case study, a set of virtual organizations to promote communication in performing collaborative tasks was designed.

For this case, there will be a number of agents for each multi-rotor. They will all run on its processor, each one being responsible for a single well-defined task. Thus, there are several types of agents, and each type is associated to the task the agent performs.

IV. SYSTEM OVERVIEW

Hawk Multicopter

To proceed with the case study, the multi-rotors named 'Hawk Multicopter', which were developed prior to this case study, are used.

This UAV has a hexacopter structure, with six arms, six motors and six propellers. Its control system consists of two parts that must be separately detailed: the electronic part and the ground control part.

A powerful Single-Board Computer running a UNIX-based system is used as the electronic part and it is responsible for running the software that controls the multi-rotor. In a multi-rotor system, this kind of electronic component provides, apart from an intelligent control, the capacity to perform all the necessary processing tasks needed for the case study. It is also possible to exploit its Wi-Fi connectivity capacity to transfer large amounts of data bi-directionally. It represents a major innovation compared to other existing multi-rotor systems using radio frequency as the communication mechanism [27][10][3].

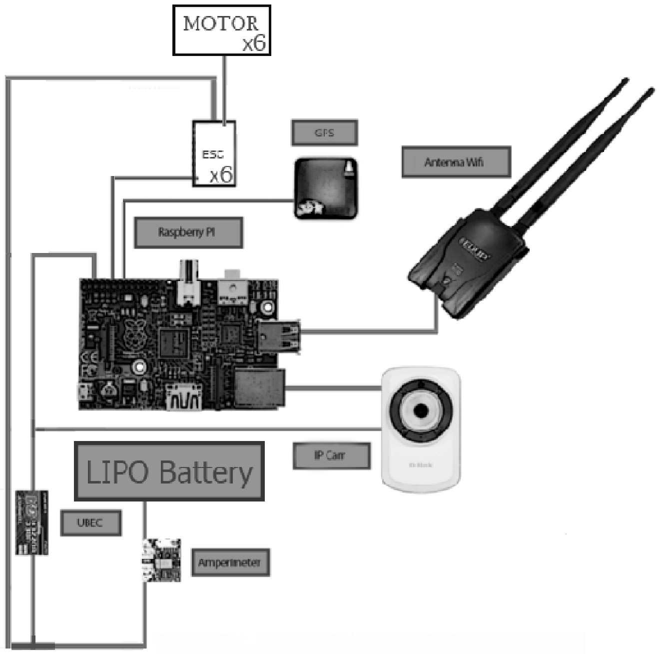


Fig. 3. Main components communication schema.

This way of communicating makes it possible to control the multi-rotor as well as to obtain information about the status of the sensors it can carry, such as the Global Positioning System (GPS) or altimeter. Of course, it can also send video images taken by its camera. Furthermore, the use of a computer instead of a radio station as a multi-rotor control element allows for communication with other components of the same network, such as other multi-rotors, and transferring information about common behaviors or goals.

Regarding the control part, 'Hawk Multicopter' multi-rotors can be controlled and monitored through a computer. This computer is responsible for executing the PANGEA platform with which each of the multi-rotors system connects.

In addition, the computer runs the necessary software to serve as a Ground Control Station (GCS). This software provides several functionalities from which the system can benefit depending on the current case study. For instance, the system will benefit from the software functionalities when new configurations or automatic behaviors are to be added to the multi-rotors.

Also, as seen in the following image, the software provides a mechanism for manual piloting. In our case, this manual mechanism is really useful for performing landing operations on the platform of the ship.

Altitud: 789 m
Velocidad: 0.0 km/h

Posición: Lat.: 40.804245 - Long.: -6.254435 - (10)

Voltaje: 9.724 V
Amperaje: 3857 mAh
Batería: 86%

RSSI: -71 dBm
Calidad: 39/70

Comando 1: Waypoint
Latitud:
Longitud:
Altitud (m):
Esperar (segs.):
Velocidad (%):

Fig. 4. GCS of 'Hawk Multicopter' during a teleoperated flight.

For the case to be treated, this tool will provide (in addition to the previously mentioned manual landing case) an entry point when establishing the initial associated configuration. It will also provide a way to jointly or individually monitor the situation of multi-rotors as required.

Multi-Agent System

The following types of agent were defined to perform the required tasks for the case study:

- Agents associated with sensors. They are responsible for monitoring each one of the multi-rotor sensors on which they depend. They provide information to those who require it to perform a task. They can be divided into the following subtypes:
    - Position agent. Through GPS sensors (Global Positioning System) and altitude (altimeter and sonar) it is able to obtain the position at all times with a 4 meters margin of error.
    - Battery usage agent. It draws on information from the battery level and the measures taken by the voltmeter and ammeter to calculate all the information about energy intake. This information is taken into account when performing the mission and when returning to the point of landing. For the

latter task, a margin of 20% capacity is saved for safety reasons.

- Vision agent. This agent uses the onboard multi-rotor camera to implement the algorithm for the detection of spills. Furthermore, it allows determining the percentage of existing spill in the captured image. In the case that the percentage is more than 80% of the image, the spill is considered. Otherwise, it is rejected.
- Pilot agent. This agent is responsible for establishing the way forward. It initially starts from a state in which the location of the spill is unknown; its initial mission is to search for the spill in a pseudorandom manner. Once a spot of the spill has been located, the agent is guided by the information provided by the Vision Agent. Finally, to return to the landing point, the multi-rotor plots the shortest route to that point to proceed with the controlled landing later on.
- Stabilizing agent. It uses sensor data in conjunction with information about the motions to be traced. It translates all information into a set of values for the multi-rotor to remain stable, whether at a resting state or in motion.
- Spokesman agent. There is only one such agent for each multi-rotor system. It represents the point to communicate and read the information relating to the system's collaborative behavior. This information can be the internal state of the multi-rotor or the new discoveries it makes to put them together with other UAVs.

And they are distributed in the following types of virtual organizations:

- Multi-rotor state. This type of virtual organization includes each one of the agents that the multi-rotor provides. The agents in this virtual organization are capable of communicating to each other after defining the relevant norm in PANGEA, but it is not possible to establish communications between agents that only belong to external virtual organizations.
- Mission state. This organization includes a spokesman type of agent representing each one of the multi-rotors in the system, and is able to keep an updated log of the global mission state. This is achieved by knowing the relevant information of all participant multi-rotors.

Under no circumstances, are agents able to communicate to other agents not belonging to one or more of the organizations where the agent is present.

For this reason, and according to the given description, agents and virtual organizations are structured as shown in the following schema:

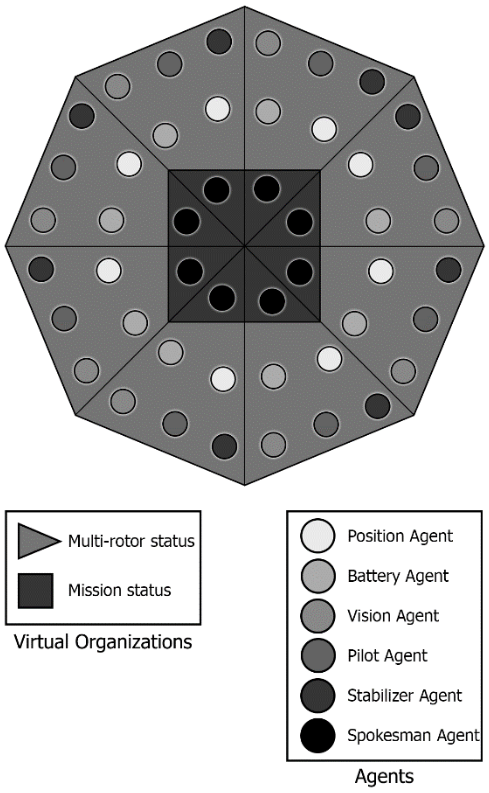


Fig. 5. Deployment of agents in the respective virtual organizations

V. RESULTS AND CONCLUSIONS

A. Use of Multi-Agent System

The use of a multi-agent platform when structuring the system offers the possibility to establish group communications, as in the case of the Mission status virtual organization. Through a single communication in the platform, the whole system of UAVs can become aware of the global state, ensure communication with a guaranteed order, and play the same role on all systems.

There are no distinctions between the UAVs, which allows them to reach a common goal much more quickly. This is because the area that eight multi-rotors are able to cover is much larger than the area that only one multi-rotor is able to cover. Furthermore, by using the multi-agent platform to communicate, UAVS will not fly over areas that have been previously analyzed by a different UAV.

In addition, each robot member of the system has its own virtual organization (Multi-rotor status), which includes agents performing multiple distributed and parallel tasks. Thus, allowing increased scalability makes it possible to apply the designed system to another scenario or to add new modules to increase its functionality, such as the addition of new sensors.

Also, thanks to PANGEA, if different agents with the same IP address are present in a virtual organization, the message bypasses the host server and is delivered directly. This causes a minimal delay (less than 2 milliseconds) between messages occurring in a single UAV and not directed to the outside.

B. Optimized communication protocol

One of the main problems the use of multiple UAVs presents is the communication among them. This communication, as detailed in our case, is done via 802.11n Wi-Fi with powerful omnidirectional antennas able to reach distances of several kilometers. Although the communication standard physical layer supports a speed of 300Mbps, depending on the environment, the perceived speed in reality is much lower.

Therefore, even though the bandwidth can appear to be sufficient without requiring major optimizations, in practice some optimizations are necessary to avoid reception problems and avoid generating more traffic, thus ensuring the delivery of messages.

Because of the reason explained in the previous paragraph, the actual agents in the system communicate to each other using an optimized communication protocol that is encapsulated within the IRC frames, which does not guarantee the ordered delivery of messages.

This protocol is based on Mavlink [19], which is a protocol oriented to air vehicles communication that allows the addition of new user-defined messages.

These frames are defined in the structure shown in the following table:

TABLE I. MESSAGE PACKET STRUCTURE

Field name	***Index (Bytes)***	***Purpose***
Payload length	0	Length of the following Payload field.
Packet sequence	1	Count of the sequence.
Message ID	2	Identification of the message type. This determines the format of the payload.
Payload	2 to (n+2)	The data to be transmitted.
CRC	(n+3) to (n+4)	Check-sum of this packet.

'Payload' encapsulates the information to be transmitted. Since this information may be of different types, each message is assigned one ('Message ID'). Therefore, its structure directly depends on this type, which is one of the 18 communication message types that have been created.

Pollutant dispersion model results improvement

Thanks to the multiple short flights produced throughout the course of a task, there has been an improvement in the estimates produced by the simulation software, GENOME. This is because the inputs from the second execution, which are fed into the simulator, use the positions of the spill that have been detected in the last flight.

ACKNOWLEDGMENT

This work has been carried out by the project Sociedades Humano-Agente: Inmersión, Adaptación y Simulación. TIN2012-36586-C03-03. Ministerio de Economía y Competitividad (Spain). Project co-financed with FEDER funds.

REFERENCES

[1] Beegle-Krause, J. (2001, March). General NOAA Oil Modeling Environment (GNOME): a new spill trajectory model. In International Oil Spill Conference(Vol. 2001, No. 2, pp. 865-871). American Petroleum Institute.

[2] Bresson, G., Aufrere, R., & Chapuis, R. (2013, July). Consistent multi-robot decentralized SLAM with unknown initial positions. In *Information Fusion (FUSION), 2013 16th International Conference on* (pp. 372-379). IEEE.

[3] Buss, H., & Busker, I. (2008). Mikrokopter.

[4] CI Pinzón, J Bajo, JF De Paz, JM Corchado. S-MAS: An adaptive hierarchical distributed multi-agent architecture for blocking malicious SOAP messages within Web Services environments. Expert Systems with Applications 38 (5), 5486-5499

[5] Corchado, J. M., Villarrubia, G., De Paz, J. F., Rodríguez, S., Zato, C., de la Prieta, F., & Bajo, J. (2013). Practical Applications of Virtual Organizations and Agent Technology. In Highlights on Practical Applications of Agents and Multi-Agent Systems (pp. 17-23). Springer Berlin Heidelberg.

[6] DI Tapia, A Abraham, JM Corchado, RS Alonso. Agents and ambient intelligence: case studies. Journal of Ambient Intelligence and Humanized Computing 1 (2), 85-93. 2010.

[7] DI Tapia, JF De Paz, S Rodríguez, J Bajo, JM Corchado. Multi-agent system for security control on industrial environments. International Transactions on System Science and Applications Journal 4 (3), pp. 222-226. 2008.

[8] DI Tapia, RS Alonso, JF De Paz, JM Corchado. Introducing a distributed architecture for heterogeneous wireless sensor networks. LNCS 5518, pp. 116-123. 2009.

[9] DI Tapia, S Rodríguez, J Bajo, JM Corchado. FUSION@, a SOA-based multi-agent architecture. International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008), 99-107. 2008.

[10] Haehnel, H. (2008, November). Remote controlled flying robot platform. In Digital Infor-mation Management, 2008. ICDIM 2008. Third International Conference on (pp. 920-921). IEEE.

[11] Hudson, J. M., & Witt, P. L. (2008). Internet Relay Chat (IRC). Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications, Volume 3, 889-897.

[12] J Bajo, JF De Paz, S Rodríguez, A González. Multi-agent system to monitor oceanic environments. Integrated Computer-Aided Engineering 17 (2), 131-144. 2010.

[13] J Bajo, JF De Paz, S Rodríguez, A González. Multi-agent system to monitor oceanic environments. Integrated Computer-Aided Engineering 17 (2), 131-144. 2010.

[14] J Bajo, JM Corchado. Evaluation and monitoring of the air-sea interaction using a CBR-Agents approach. Case-Based Reasoning Research and Development, 50-62. 2005.

[15] JA Fraile, J Bajo, JM Corchado, A Abraham. Applying wearable solutions in dependent environments. Information Technology in Biomedicine, IEEE Transactions on 14 (6), 1459-1467. 2011.

[16] JM Corchado, J Aiken, N Rees. Artificial intelligence models for oceanographic forecasting. Plymouth Marine Laboratory. 2001.

[17] JM Corchado, J Aiken. Hybrid artificial intelligence methods in oceanographic forecast models. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 32(4), pp. 307-313. 2002.

[18] Karantzalos, K., & Argialas, D. (2008). Automatic detection and tracking of oil spills in SAR imagery with level set segmentation. International Journal of Remote Sensing, 29(21), 6281-6296.

[19] Meier, L., Camacho, J. F., Godbolt, B., Goppert, J., Heng, L., & Lizarraga, M. (2009). Mavlink: Micro air vehicle communication protocol.

[20] Nirchio, F., Sorgente, M., Giancaspro, A., Biamino, W., Parisato, E., Ravera, R., & Trivero, P. (2005). Automatic detection of oil spills from SAR images. International Journal of Remote Sensing, 26(6), 1157-1174.

[21] Ortiz-Pena, H. J., Sudit, M., Hirsch, M., Karwan, M., & Nagi, R. (2013, July). A multi-perspective optimization approach to UAV resource management for littoral surveillance. In *Information Fusion (FUSION), 2013 16th International Conference on* (pp. 492-498). IEEE.

[22] P. D. Boehm, D. L. Flest, D.Mackay, and S. Paterson, “Physicalchemical weathering of petroleum hydrocarbons fromthe ixtoc I blowout: chemical measurements and a weathering model,” Environmental Science and Technology, vol. 16, no. 8, pp. 498– 505, 1982.

[23] S Rodríguez, B Pérez-Lancho, JF De Paz, J Bajo, JM Corchado. Ovamah: Multiagent-based adaptive virtual organizations. Information Fusion, 2009. FUSION'09. 12th International Conference on, 990-997. 2009.

[24] S Rodriguez, V Julián, J Bajo, C Carrascosa, V Botti, JM Corchado. Agent-based virtual organization architecture. Engineering Applications of Artificial Intelligence 24 (5), 895-910. 2011.

[25] S Rodríguez, Y de Paz, J Bajo, JM Corchado. Social-based planning model for multiagent systems. Expert Systems with Applications 38 (10), 13005-13023. 2011.

[26] Tapia, D. I., Alonso, R. S., García, Ó., Corchado, J. M., & Bajo, J. (2013, July). Wireless Sensor Networks, Real-Time Locating Systems and Multi-Agent Systems: The Perfect Team. In *Information Fusion (FUSION), 2013 16th International Conference on* (pp. 2177-2184). IEEE.

[27] Tretyakov, V., & Surmann, H. (2008). Hardware architecture of a four-rotor UAV for USAR/WSAR scenarios. In Workshop Proceedings of SIMPAR 2008-International Con-ference on Simulation, Modeling and Programming for Autonomous Robots.

[28] United States. National Ocean Service. Office of Response and Restoration, General Noaa Operational Modeling Environment (Gnome) Technical Documentation, U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, National Ocean Service, Office of Response & Restoration, 2012.

[29] Wojtaszek, K. (2003). Application of transport model for building contingency maps of oil spills on the North Sea (Doctoral dissertation, MS Thesis, Delft University of Technology).

[30] Zato, C. Villarrubia, G., Sánchez, A., Bajo, J., & Corchado, J. M. (2013). PANGEA: A New Platform for Developing Virtual Organizations of Agents. International Journal of Artificial Intelligence, 11 (A13), pp. 93-102