An Energy-aware Collaborative Multi-agent System for Autonomous Underwater Vehicles

Luis Felipe Zapata-Rivera, Catalina Aranzazu-Suescun and Imadeldin Mahgoub, PhD.
Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL 33431, USA
E-mail: {lzapatariver2014, caranzazusue2014, mahgoubi}@fau.edu

***Abstract*—Underwater vehicles most of the time operate in environments that normally are inaccessible to humans. They can operate in the depth of the ocean where they have challenging conditions such as: pressure, light and visibility, among others. Collaborative autonomous underwater vehicle (AUV) systems provide a possibility to create groups to work as a team and do some specific tasks that generally cannot be done by only one vehicle, saving time and energy in the network. Collaborative approaches aim to improve the response time of the task and save the energy on the network. This paper compares the efficacy and efficiency of three different approaches that use multi-agent systems. All approaches use different zones or clusters of vehicles. The first approach uses a leader AUV responsible for the message propagation, the second uses cluster heads and the third one uses a BDI model (Belief, Desire and Intention) to allow the agents to have a simple human behavior. The results show that the second and third approaches reduce the energy consumed in the network compared with the first approach (leader).**

***Index Terms*—Autonomous Underwater Vehicles, BDI, Collaborative, FIPA-ACL, Multi-agent System.**

I. Introduction

Underwater vehicles operate in environments that normally are inaccessible to humans. They can operate in the depth of the ocean where conditions such as: pressure of the water, light, visibility, are challenging problems for the development of the vehicles and for the complete the assigned tasks. Taking into account these difficulties is important to provide reliable collaboration schemes supported by the adequate combination of communication protocols, software and hardware. There are two main types of operation for underwater vehicles: crewed vehicles and non-crewed vehicles. Non-crewed vehicles consist of remote controlled vehicles and Autonomous Underwater Vehicles (AUV) that have the possibility to operate without the assistance of humans. The autonomous underwater vehicles have gained importance during the last years because they enable scientist and researchers to access previously inaccessible places in the ocean. These advances are now possible due to improvements in hardware, software and telecommunication technologies. The development of self-driven vehicles or autonomous vehicles are not exclusive for underwater vehicles, there is also a very wide area of research for other type of autonomous vehicles such as: cars, airplanes, and drones. The developments in one specific domain recurrently can be implemented in other domains. To accomplish the tasks assigned to an autonomous underwater vehicles network, the concept of collaboration has been included in recent research work [2]–[6]. Collaboration schemes, allow autonomous underwater vehicles to reach their goals in less time, with less resources and less energy. Due to the possibility of grouping multiple vehicles to perform some specific tasks, the concept of agents has demonstrated to be a good option to be implemented in this type of vehicles [7], [8]. Using agents is possible to create scenarios in which the vehicles behave and make decisions like humans, when they work as a team. Some other examples of collaboration schemes are evident in the nature, for example ants or bees that cooperate to construct their places of living.

This paper presents three approaches that use the concept of multi-agent systems to improve the communications scheme, the time to reach an objective and the energy consumption in an AUV network. The first approach includes an AUV network where a vehicle is chosen as leader after being the first to complete the task, the second approach uses cluster heads to improve the communication strategy and the third one uses a BDI model (Belief, Desire and Intention) to allow the agents to have a simple human behavior. The simulations were done using Netlogo platform. The rest of the paper is organized as follows: Section II presents the related work in the field. Section III gives a background on agents, multi-agent and FIPA-ACL language. Section IV summarizes the problem in the AUV networks. Section V presents the model proposed and the non-collaborative approach. Section VI shows the simulation environment. Section VI-A presents the results and analysis. Finally, conclusions are stated in Section VII.

II. Related works

Communication based on acoustic signals is dependent on environmental conditions and it is band and range limited. Authors of [9] use sonar to transmit the information between the vehicles and an ad-hoc network with fixed and mobile nodes on water and a base station on ground. The vehicles can dynamically decide to stay or not in the group. The group controller is the fixed node that maintains track of mobile nodes, detects compromised nodes and provides keys to encryption. The goal of the network is to protect some

978-1-5090-6468-7/17/$31.00 ©2017 IEEE

objects. The authors design an algorithm that allows the nodes to use a set of rules to be always close to the object, covering the largest area possible but being aware of its neighbors to be on their communication range. They use dead reckoning to know their location in the water. Similarly, in [3] before submerge, the vehicles synchronize the time with the GPS to send data in a periodic time. With the angle of the other vehicles, measured using a compass, each vehicle can adjust its own position in the network. Then, the vehicles use dead reckoning to estimate their own position.

Several authors have been working on the autonomous underwater vehicles field to use them to do tasks such as monitoring, surveillance and search of objects. To have a better performance and reach the goals faster, some authors have been working on the use of the concept of agents. Authors of [10], define two types of vehicles: Search AUV (SV), that has wide field target detection sonar with low resolution, and Classify AUV (CV) that has narrow field target detection sonar with high resolution. The target location is known a priori. The location for each vehicle is known using dead reckoning. Task allocation could be distributed or centralized. In the first one, each vehicle operates independently, but coordinated with the other vehicles. The agent-vehicle decides what to do in every moment; this is in a predefined negotiation framework. In the centralized approach, there is a coordinator that plans the mission and spread the information to the other AUVs, using Time Division Multiple Access. The authors develop a location aided auction framework that incorporates a negotiation among all the vehicles. The authors of [11] present an approach based on a hierarchical multi-agent control on the AUVs. They define 3 control levels: supervisory, that is the higher level, can make decisions, monitor the vehicle and communicate with mothership. The second level or mission, acts as referee among other levels. Finally, the third level is the vehicle level that has low level control and interacts with sensors. The authors used the architecture only in one AUV, but can be scaled to be used in missions with several cooperative vehicles. Similarly, the author of [12] presents a motion control architecture that has 4 levels. The blackboard system level, process information, manage the system and control the agent behavior. The second level is the elementary behavior agent group, where agents can be sway, yaw or heave agents. The third level is the reflection behavior agent group, used when the system has faults, and the final level are the actuators. The architecture is part of the AUV, then all level agents can communicate between then to make good decisions in the development of a certain task but the agents cannot communicate with agents of other AUVs.

Using the different classification of agents, some approaches use goal based agents and service based agents. Authors of [13] use Goal Driven Autonomy (GDA) model that allows the nodes to create and prioritize goals detecting unusual situations. The GDA model has an agent planner that interacts with an agent controller. The agent controller interacts with the environment and the state transition system. The controller detects discrepancies, search the possible causes of the discrepancies, formulate a goal and manage this goal. To detect a discrepancy they have used thresholds for every situation. On the other hand in [4] the authors use Knowledge-based database to feed the agents of the AUVs. This database has information about human skills that allows the vehicles to reason, refine and adapt to mission. Each vehicle provides basic services or functionality, so the system can know the capabilities of each node. The authors develop an architecture for the development of the agents that is capable of dynamically adapt the behavior of the agents and reconfigure them to deal with the changes on the environment. These agents are service oriented, then the agents that offer several services cooperate to perform a big process and complete a mission.

Some authors have used the model of colonies such as ant and bees colonies to model the behavior of the vehicles. The authors of [6] use an artificial ant agent to collaborate in a task. The agents have to build solutions, update pheromones and execute. The goal is to find the cheapest price in a consuming task allocation scheme. The price is based on the energy that the vehicle consumes. In recent years there has been research on the side of the software agents. The authors of [7] use collaborative agents to solve large problems. The agents have reactive behavior, internal intentions and objectives. The agents can interact between them and with the environment. The agents have some goals that could be passive (imposed) or could be active, that means that the agent can manage the actions interacting with the environment to satisfy the goal. The agents have constraints in the sensors, so they cannot perceive all the possibilities and attributes of the environment. Each agent has its own knowledge source. The nodes have some tasks that they have to plan and transmit at a certain time. The proposed research was to create a strategy to transmit the information in a flexible way using different transmission frequency that the agents must select according to the environment, but the AUV can not decide to be part of the task or not.

Our model uses multi-agent systems but is different to [7] in the use of utility based agents and clustering. We argue that using clustering and rewards in the network, it is possible to save more energy and time in the execution of a task. We proposed also a system that uses BDI (Belief, Desire and Intention) model to give the AUV a simple human behavior.

III. Agents, Multi-agent Systems and FIPA-ACL Language

A. *Software Agents and Multi-agent Systems*

The agents are software entities that can act in an autonomous way and make some decision based on the environment, and also can learn, cooperate and move. Russell and Norvig [16] group agents into five classes, based on their degree of perceived intelligence and capability:

- Simple reflex agents: the decisions are based on the immediate perceptions and in some rules.
- Model-based reflex agents: the decisions depend on history or ideal environment.
- Goal-based agents: have fixed goals to achieve that can be prioritized.

- Utility-based agents: have goals to achieve and can determine to continue or leave the goal based on a utility function.
- Learning agents: can perform activities according to rules, models, goals, utilities and can get feedback of the result, learn and improve its decisions.

A multi-agent system is a distributed system in which the nodes or elements of the system are agents. In these systems, the combined behavior of agents produces a smart result. Multi-agent systems coordinate intelligent behavior of autonomous agents. These agents are part of a network and can coordinate or share their knowledge, goals, skills and plans to make a decision or achieve a global goal [19]. A multi-agent system may also contain combined human-agent teams.

B. FIPA-ACL, KQML and BDI

Communication between agents is based on protocols or intercommunication schemes like *Knowledge Query Manipulation Language (KQML)* and *Agent Communication Language (ACL)*. KQML is a language and protocol for communication between agents. It was created on the early $90s$ as part of *DARPA Knowledge Sharing Effort* [17]. The KQML can be used to interact with an intelligent system by an application program or by other intelligent system. FIPA-ACL was proposed by the *Foundation for Intelligent Physical Agents* (FIPA), and is a standard language for communication between agents that was the successor of KQML [18], [19]. The implementation of software agents can be performed with the help of software models such as *The Belief-Desire-Intention* (BDI), which is a software model developed for programming intelligent agents. In general, this architecture is characterized by the use of beliefs, desires and intentions in agents, similar to human behavior. To implement multi-agent software systems we need 3 elements: Autonomy, Intelligence and Mobility.

IV. Problem Definition

The first applications with underwater vehicles used wired communications where each vehicle was connected to a mothership using cables. These cables were used also to send power from the mothership to the AUVs. However, the operation range was severely limited to only 100 to 200 meters. To give more autonomy to the vehicles, the network needs to be wireless. There are 3 options to transmit data in a wireless media, using radio, optical or acoustic signals. For the first 2 even with a wide bandwidth, MHz in radio and GHz in optical, and communication speed in the order of Mbps and Gbps, respectively, the operation range still being too short. For having an operation range above the 200 meters, the acoustic signals are the better option for medium and long communication networks.

The speed of sound in the water depends on the depth. On the surface, the speed of sound is approximately constant but, when the AUV starts to descend, the temperature decreases then, the speed also decreases. But when the AUV is in the deep ocean, because the temperature remains constant but the pressure starts to increase, also the speed increases.

TABLE I: Acoustic vs Radio signals

RADIO	**ACOUSTIC**
High Bandwidth (BW)	Low Bandwidth (BW)
Short delays	Long delays
Well understood propagation	Complicated propagation
Distance dependent BW	Distance Independent BW
White noise	Frequency dependent noise
Accepted channel models	No comprehensive channel models

Normally, the speed of sound in the water is $1500m/s$ and the transmission rate is between 5 and $100kbps$ [20]. The table I presents a comparison between the acoustic signals and the radio signals [21]. The bandwidth on the acoustic signals is only $5KHz$, and it can be shorter if the distance between transmitter and receiver is in the range of kilometers. The propagation delay could be even 1 second and due to the several interferences under the water. The propagation is complicated and there is not an accepted channel propagation model. The noise cannot be modeled as Additive White Gaussian Noise (AWGN) because it is frequency dependent. [20], [22].

Commonly, the AUV networks have been developed with a low number of vehicles, that have some specific objectives but sometimes do not collaborate with each other. This could result in a long time to reach the objective and in a high energy consumption in the network. Multi-agent systems can be implemented as a strategy to solve the collaboration problems.

Next, we present the problem definition: Given an AUV network deployed in an area A, consisting of n nodes with similar sensing components and a stationary mothership M, design an energy-efficient distributed algorithm using multi-agent systems for detecting and reporting an object O inquired by the mothership M.

V. Leader Agent-based approach, Clustering Agent-based approach and BDI Agent-based approach for multi-agent systems for Object Detection and Reporting in AUV Networks

We consider an AUV network consisting of n homogeneous vehicles N_1, N_2, ..., N_n and a stationary mothership M. The vehicles have the same communication range R_c and an initial energy E_{init}. The mothership M has communication range R_c and infinite energy.

Each vehicle has a software agent for collaboration purposes. The goal of the task is to find an object in the ocean. To model our work, the following assumptions are made:

- There is only one object to search in the area.
- The AUVs navigate in a square zone A.
- The AUVs navigate at the same depth, then all AUVs move in the same plane and they have the same movement pattern (random walk).
- The AUVs are using acoustic frequencies to communicate, so the range of coverage is bigger than using radio.
- The AUVs check the medium before sending the message to avoid collisions.

- The regular AUVs have the same resources (sensing components and processing power) to search the object, but the energy could change depending on the scenario.
- The cluster head AUV has more energy and a larger range of coverage for the communication.
- The AUVs have the same speed.
- The AUVs cannot overpass the border of the area A, then, the AUVs always stay in the square zone.

Our approaches use multi-agent collaborative model to improve the performance of the network, save energy and reduce the time to reach the goal. The communication protocol adopted for the agents was FIPA-ACL. The goal of our network is to detect one square object in the sea floor. If an AUV finds the object needs to send the information of the location of the object to its neighbors and to the mothership M. The zone A is divided in i sub-zones. All the sub-zones have the same number of AUVs. The AUVs cannot go to other zones if they have not found the object or if have not been informed of the finding.

A. *Leader Agent-based approach*

All AUVs start with the same energy. If an AUV found the object it timestamp the finding of the object, them it starts searching its neighbors to inform them about the finding. When it finds a neighbor, they exchange timestamps if both have them. The one with older timestamp, becomes the leader and will continue searching for its neighbors to inform them about the finding. The other one will acknowledge (ACK) the finding and will go to the mothership. After informing all the neighbors, the leader needs to visit the other zones and inform at least one AUV for each zone. Then, the leader goes to the mothership. The nodes informed by the leader become co-leaders and will propagate the message only in their one zones. In this case we have a leader that is in charge of collaboratively inform everybody about the finding. The leader will then be any AUV, that first find the object. The pseudocode of the object reporting mechanism is presented in *Algorithm 1*.

Algorithm 1 Leader Agent-based approach - Object Reporting (agent N_j)

```
if N_j found object == True then
    set timestamp of object found
    while neighbor's table check == not complete do
        continue navigating searching for neighbors
        if find neighbor then
            if neighbor also found object then
                Check time stamp of both nodes
                if N_j.timestamp > neighbor.timestamp then
                    N_jleader == True
                    check mark neighbor in neighbor's table
                else
                    exchange neighbor's table
                    ACK object found and go to M
                end if
            end if
        end if
    end while
    visit every other zone to inform the finding to one AUV in that zone
    Go to the mothership
else
    continue searching the object
    if find neighbor from other zone then
        if neighbor_leader == True then
            Exchange information about finding
            N_jCOleader == True
            Find neighbors to communicate finding
        end if
    end if
end if
```

Algorithm 2 Clustering Agent-based approach - Object Reporting (agent N_j)

```
if N_j found object == True then
    set timestamp of object found
    while object found message send to CH == False do
        navigate searching for Cluster head
        if find CH == True then
            inform CH about object found
        end if
    end while
    go to the mothership M
else if N_j receive CH message == object found then
    go to the mothership M
else
    continue searching the object
end if
```

B. *Clustering Agent-based approach*

In this case, all AUVs also start with the same energy. The AUVs are part of a cluster that is determined by a sub-zone and each cluster has a Cluster Head (CH). The CH is a static buoy at the center of the sub-zone that has radio communication with its CH neighbors and with the mothership M. If an AUV finds the object, it timestamps the finding and goes to the CH location to inform the location of the object. Then, the CH broadcast the information to its cluster members and to the CHs neighbors that are within its range of coverage. If the other CHs are not in its range of coverage, it is going to use a multi-hop strategy. When other CH receives the message of object found, it also broadcast the information to its nodes. When a node in a cluster receives the found message, goes to the mothership M. This case is more energy efficient because once a vehicle finds the object and informs the CH about the finding, the rest of vehicles of the cluster will receive the message and go directly to the mothership. The same situation happens for the other clusters. The problem with this scheme is that it only works for scenarios with an extra device, the buoy, and is more expensive. The pseudocode of the object reporting mechanism is presented in *Algorithm 2*.

C. *BDI Agent-based approach*

Each AUV has beliefs, desires and intentions, so we use the concept of utility-based agents [16]. The beliefs are defined as optimistic, doubtful, or pessimistic. An AUV can leave the zone and go to the mothership if any of these two conditions occurs:

- The AUV found the object or received a message that someone found the object. It confirms that its neighbors know that the object was found, sending a message to them and receiving an acknowledgement.
- It is or it becomes a pessimistic AUV

Each AUV has a table with the IDs of its neighbors. The desire of all vehicles was defined as the goal to find the object, and the intentions were defined as the actions needed to accomplish the goal. In this case we have three intentions: first, look for the object which include the functions of moving and sensing. Second, communicate with other AUVs and finally, go to the mothership. The implementation of beliefs, desires and intentions, was done using 3 stacks, each representing one of them. For example, at the beginning each AUV has the 3 intentions in the stack but executes the one corresponding to its beliefs. When the AUV changes its beliefs for example from optimistic to pessimistic, they remove the intention of ”look for the object” and execute the new intention ”go to the mothership”. In this case we implement a utility function that is evaluated by vehicles before each movement. This feature helps the agents to make decisions about continuing or stopping the search of the object at some point in the simulation. The utility function evaluation allows vehicles to change their beliefs at any point of the search. The utility function is given by:

$$U = P * R \tag{1}$$

Where U is the utility, P is a probability, R is a reward that can be configured in the simulation and E_{init} is the initial energy of the AUV, and is a value between 150 and 1000 units. the probability P is given by:

$$P = 1 - \frac{(E_{init} - CurrentEnergy)}{E_{init}} \tag{2}$$

Beliefs of the vehicles are given based on the initial energy and a reward:

- Optimistic vehicles will risk more during the search, even if their energy level is getting low. The Utility function is a value between 800 and 1000.
- Doubtful vehicles could decide to be optimistic or pessimistic at a specific point of their energy consumption. The utility function is a value between 500 and 800.
- Pessimistic vehicles do not risk their energy, so always decide to not be part of the mission, and leave the zone to go to the mothership. They can be defined also as energy saver AUV. The utility function is a value less than 500.

The communication protocol adopted for the agents was FIPA-ACL, using some of the performatives defined such as [18]:

- to receive [msg]
- to send [msg]
- create-message [performative]
- create-ack [performative msg]
- get-sender [msg]
- get-receivers [msg]
- add-sender [sender msg]
- add-receiver [receiver msg]

These performatives allow agents to communicate with each other using the standard defined by FIPA. The pseudocode of the object reporting mechanism is presented in *Algorithm 3*.

Algorithm 3 BDI Agent-based approach - object Reporting (agent N_j)

if N_j utility function ≤ 500 **then**
  Declare itself pessimistic
  Go to mothership M
else if $500 < N_j$ utility function ≤ 800 **then**
  Declare itself doubtful
  set decision to continue or give up
  if decision == give up **then**
    Declare itself pessimistic
    go to sink or mothership S
  else
    Declare itself doubtful
    search for the object
  end if
else if N_j utility function > 800 **then**
  Declare itself optimistic
  search for the object
end if
if N_j found object == $True$ **then**
  set timestamp of object found
  while neighbor's table check == not complete **do**
    continue navigating searching for neighbors
    if find neighbor **then**
      if Neighbor also found object **then**
        Check time stamp of both nodes
        if $N_j.timestamp > neighbor.timestamp$ **then**
          $N_{jleader}$ == $True$
          check mark neighbor in neighbor's table
        else
          exchange neighbor's table
          ACK object found and go to M
        end if
      end if
    end if
  end while
  visit every other zone to inform the finding to one AUV in that zone
  Go to the mothership
else
  continue searching the object
  if find neighbor from other zone **then**
    if $neighbor_leader$ == $True$ **then**
      Exchange information about finding
      $N_{jCOleader}$ == $True$
      Find neighbors to communicate finding
    end if
  end if
end if

VI. Simulations

We conducted simulations using Netlogo 5.3.1 [1]. NetLogo is based on the Logo programming language and it has a simple and powerful interface that allows a large variety of simulations and modeling phenomena. The models in Netlogo can have a high degree of complexity and the networks

TABLE II: Parameters of the simulation

Simulation time	$2hr$
Length of the side of $A.L$ (m)	1000
Number of Clusters or zones	4
Area of the object	$100m^2$
Initial energy	$1000units$
speed of the AUV (m/s)	2.5

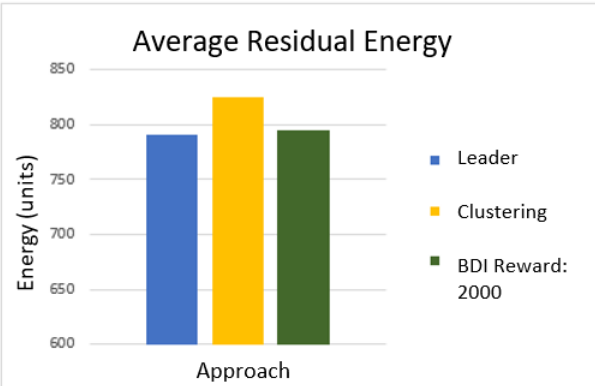

Fig. 1: Average residual energy per vehicle.

can have hundreds of agents. NetLogo was designed by Uri Wilensky in 1999, director of the University Northwestern and is written in Java and Scala and runs on Java virtual machine. NetLogo is part of a series of multi-agent systems modelers which began with StarLogoT [23]. The parameters chosen for the simulations were defined taking into account values of normal AUVs real life mission. The speed of the vehicles is the average speed of an AUV given by [24]. In the simulations, the user can design the environment adding obstacles and can define the number of AUVs in the area. However, in this case, we do not add any obstacle in the area. The parameters for the simulation are presented in the Table II.

The AUV network is deployed into a square area with side length $A.L = 1000m$, and the object to search is randomly deployed in the field at the beginning of the simulation.

For each approach, 100 runs of the simulation were completed. The three approaches include a squared area divided in 4 zones. Each zone has a certain number of AUVs that can be modified. The zones can have some obstacles which the AUVs are able to avoid. For the simulation of the models some assumptions of the scenarios and AUVs were made. The search area is 2 dimensional (X and Y axis). All vehicles are the same type of vehicles in terms of movement patterns (random walk), the consumption of energy depends exclusively of the amount of movement, the power consumption of the sensor and the communication devices are not included in the calculations. No AUV could exit the zone to go to the mothership if it did not find the object or if it did not receive a message that other AUV found the object. The initial energy of the AUVs is $1000units$ in all the scenarios.

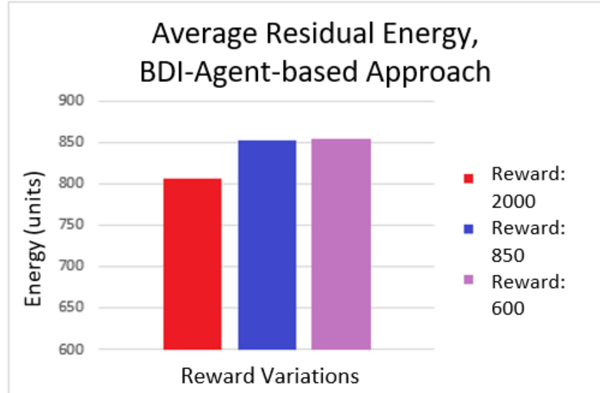

Fig. 2: Average residual energy of the network BDI Agent-based approach.

A. Simulation Results

Figures 1 and 2 show the residual energy of the network for $n = 20$ nodes, 5 in each zone, and length of the area $A.L = 1000m$.

In Figure 1, the residual energy for the first and third approaches is lower than the second approach, because the communication strategy requires one of the vehicles share the information that the object was found with the neighbors in the zone and with one vehicle in the other zones. The second approach uses a static node that acts as a cluster head whose role is to collect the information and share this information with the mothership and the other cluster heads. This setup reduces the energy consumption of the vehicles overall. For the case of reward equals to 2000 in the BDI approach, the simulation runs similarly to the Leader Agent-based and Cluster Agent-based approaches. There is no evident advantage due to the use of the BDI architecture, because the reward is high enough to keep all the vehicles active during the whole mission.

Using the BDI approach the results for the energy varies significantly according to the rewards for the vehicles. In Figure 2, when the reward is small (600) about 50% of the vehicles declare themselves pessimistic, compromising the success of the mission but saving more energy. When the reward is 850, more vehicles are motivated to keep looking for the object, then the residual energy is similar to the scenario with reward of 600, but is higher than in the case of a reward of 2000. This is because, when the reward is 2000, more optimistic vehicles are searching for the object, consuming energy, but when the reward is 850 some vehicles are looking for the object and some are saving energy being pessimistic.

Tables III and IV present the results for the average time to find the object. As shown in the tables, the three approaches have similar times to find the object. Only in the BDI Agent-based approach when the reward is 600, more vehicles abandon the mission, and less cars are searching for the object, which affects the time to reach the goal.

Tables V and VI show the percentage of missions com-

TABLE III: Average time to complete the mission

Approach	Time (min)
Agent-based	72.78
Clustering-Agent-based	59.65
BDI Agent-based, reward=2000	69.9

TABLE IV: Average time to complete the mission approach 3

Reward variation	Time (min)
BDI Agent-based, reward=2000	69.9
BDI Agent-based, reward=850	67.58
BDI Agent-based, reward=600	81.1

pleted. For Table V, the approaches have similar behavior and approximately 83.5% of the times, the object is found. This is due to the use of a random walk model in the simulation. Similar to the time comparison, the case with reward of 600 has less percentage of mission success due to more vehicles abandoning the mission.

TABLE V: Percentage of missions completed approach 1 and 2

Approach	Percentage (%)
Leader Agent-based	84
Clustering Agent-based	82
BDI Agent-based, reward=2000	83.3

TABLE VI: Percentage of missions completed approach 3

Reward variation	Percentage (%)
BDI Agent-based, reward=2000	83.3
BDI Agent-based, reward=850	83
BDI Agent-based, reward=600	58

Table VII shows the average number of cars that abandon the mission in the BDI-Agent-based approach. There is a direct relation between the number of cars leaving the mission and the reward. This is, when the reward is bigger, such as 2000, all the cars are optimistic, so they continue searching the object. Now, when the reward is small, more cars decide to leave the mission, and it is less likely to find the object.

VII. Conclusions

This paper presents three strategies based on agents for autonomous underwater vehicles. Based on the simulation results, the clustering agent-based is the most energy efficient approach, additionally it saves time in the communication stage of the mission. However, the disadvantage is that it requires additional infrastructure for the cluster heads.

Better success rates and less time in the missions could be obtained implementing movement strategies different than random walk, for example avoiding the AUVs to pass through the same place twice.

The BDI agent-based approach is an important model because it gives the system a decision making behavior similar to humans. The beliefs, desires and intentions provide three levels of configuration for the agents, whose combinations give the system sometimes an unpredictable result.

TABLE VII: Average number of cars abandoning mission approach 3

Reward variation	Average number of cars
BDI Agent-based, reward=2000	0
BDI Agent-based, reward=850	6.8
BDI Agent-based, reward=600	9.1

More sophisticated knowledge data bases can be integrated in the future to provide the agents with better criteria for decision making. Additionally, learning capabilities can be given to the agents to have a better performance in the missions.

References

[1] Netlogo, Last access July 25, 2017. (Online): https://ccl.northwestern.edu/netlogo/

[2] A. Caiti, V. Calabro and A. Munafo, Underwater sensor networks with mobile agents: experience from the field. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. 2013.

[3] L. Paull , M. Seto and J. J. Leonard, Decentralized Cooperative Trajectory Estimation for Autonomous Underwater Vehicles. *International Conference on Intelligent Robots and Systems (IROS 2014), IEEE*. 2014.

[4] C. C. Insaurralde, Y. R. Petillot, Capability-Based Engineering for Maritime Robotics Service-Oriented Agent Technology in Unmanned Marine Vehicles. *Proceedings of the Global Virtual Conference*. 2013.

[5] S. Gupta, J. Hare and Z. Shengli, Cooperative coverage using autonomous underwater vehicles in unknown environments. *Oceans Conference*. 2012

[6] J. J. Li, R. B. Zhang and Y. Yang, Meta-Heuristic Ant Colony Algorithm for Multi-Tasking Assignment on Collaborative AUVs. *International Journal of Grid Distribution Computing*. 2015.

[7] A. L. Laureano-Cruces, T. Ramirez-Gonzlez, L. Sanchez-Guerrero and J. Ramirez-Rodriguez, Multi-Agent System for Real Time Planning Using Collaborative Agents. *International Journal of Intelligence Science*. 2014.

[8] Z. Shi, J. Zhang, Jinpeng Yue and X. Yang, A Cognitive Model for MultiAgent Collaboration. *International Journal of Intelligence Science*. 2014.

[9] A. Caiti, V. Calabro, G. Dini, A. Lo Duca and A. Munafo, Secure Cooperation of Autonomous Mobile Sensors Using an Underwater Acoustic Network. *Sensors Journal*. 2012.

[10] Y. Deng, P. P. J. Beaujean, E. An and E. Carlson, Task Allocation and Path Planning for Collaborative Autonomous Underwater Vehicles Operating through an Underwater Acoustic Network. *Journal of Robotics*. 2013.

[11] T. Yew Teck and M. Chitre, Hierarchical Multi-Agent Command and Control System for Autonomous Underwater Vehicles. *IEEE/OES Autonomous Underwater Vehicles (AUV)*. 2012.

[12] L. Zhang, D. Jiang, J. Zhao and S. Ma, An AUV for Ocean Exploring and its Motion Control System Architecture.*The open mechanical engineering journal*, 2013.

[13] M. A. Wilson, J. McMahon and D. W. Aha, Bounded Expectations for Discrepancy Detection in Goal-Driven Autonomy. *Association for the Advancement of Artificial Intelligence (www.aaai.org)*. 2014.

[14] M. E. J. Newman. Spread of epidemic disease on networks. *Physical Review E*. Jul 2002.

[15] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers and R. Evans, Software Agents: A review. *Department of Computer Science, Trinity College Dublin, Tech.*. 1997.

[16] S. Russell, P. Norving, Artificial Intelligence: A Modern Approach. 3rd Edition. *Pearson*, 2009.

[17] T. Finin, R. Fritzson, D. McKay and R. McEntire, KQML as an agent communication language. *Proceedings of the third international conference on Information and knowledge management*. 1994.

[18] FIPA ACL, Message Structure Specification. (Online):http://www.fipa.org/specs/fipa00061/

[19] L. Zapata Rivera and J. D. Zapata Rivera, Collaborative Multiagent Systems. Undergraduate Thesis. Eafit University 2010.

[20] M. Stojanovic, Underwater wireless (acoustic) Communication. Northeastern University. Uvigo video conference. October 2010. (Online): http://tv.campusdomar.es/video/561429c61f56a88005b66665

[21] G. Leus, Multicarrier Acoustic Underwater communication. Delft University of Technology. Uvigo video conference. March 2011. (Online): http://tv.campusdomar.es/video/561429d11f56a88005b6670d
[22] F. Mosca, G. Matte and T. Shimura, Low-frequency acoustic source for AUV long-range communication. *MAST Europe*. 2011.
[23] StarLogoT, Last access July 25, 2017. (Online): https://ccl.northwestern.edu/cm/StarLogoT/
[24] Autonomous Underwater Vehicle, REMUS 100, Last access July 28, 2017 (Online): https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/D241A2C835DF40B0C12574AB003EA6AB?OpenDocument