A Hybrid Implementation Model to Develop Cooperative Controllers for Team-Based Operations of UAV/AUS-MAUVs Group

Mô hình thực thi lai để phát triển bộ điều khiển phối hợp cho hoạt động theo đội hình của nhóm UAV/AUS-MAUVs

Ngo Van Hien

Hanoi University of Science and Technology, No. 1, Dai Co Viet, Hai Ba Trung, Hanoi, Viet Nam

Received: November 27, 2017; Accepted: March 28, 2018

Abstract

A novel hybrid control model is proposed to implement cooperative controllers, which permit an Unmanned Aerial Vehicle (UAV) coordinated with the Autonomous Unmanned Ship/Multiple Autonomous Underwater Vehicles (AUS/MAUVs) team to effectively perform missions of ocean exploration in the wide range. This model is based on hybrid automata and the Real-Time Unified Modeling Language (Real-Time UML) for capturing the whole development lifecycle of cooperative controllers. The paper shows out stepwise the main research contents as follows: the coordinated structure and scenarios are define to gather the requirements of control analysis; hybrid automata's features are specialized to model the coordination behaviors of UAV/AUS-AUVs; the real-time communication pattern is created by using the 'capsules, ports and protocols' notation of Real-Time UML for depicting in detail the design components. The detailed design components are then converted into the implementation model by using open-source platforms such as OpenModelica in order to quickly simulate the cooperative controller. Following this proposed model, a cooperative controller permits a quadrotor UAV combined with a pair of small-scale AUS/03-AUVs to perform pre-determined search scenarios with the coordination mechanisms for ocean exploration, was designed and simulated with good reliability and feasibility.

Keywords: UAV/AUS/MAUVs, Cooperative Control, Team-Based Operations, Hybrid Automata, Real-Time UML.

Tóm tắt

Một mô hình điều khiển lai mới được đề xuất trong thực thi bộ điều khiển phối hợp; nó cho phép một phương tiện bay không người lái (UAV) liên kết với nhóm tàu thủy không người lái tự hành và đa phương tiện không người lái tự hành dưới nước (AUS/MAUV) thực hiện một cách hiệu quả các tác vụ thăm dò đại dương trong phạm vi rộng. Mô hình này dựa trên Automate lai và ngôn ngữ mô hình hóa hợp nhất trong thời gian thực (Real-Time UML) để mô tả toàn bộ vòng đời phát triển của bộ điều khiển phối hợp. Bài báo trình bày từng bước các nội dung nghiên cứu chính như sau: Cấu trúc và kịch bản phối hợp được xác định nhằm đưa ra các yêu cầu về phân tích điều khiển; Các đặc trưng Automata lai được cụ thể hóa nhằm mô hình hóa ứng xử phối hợp của UAV/AUS-MAUVs; Mẫu kết nối truyền đạt trong thời gian thực được thiết lập thông qua sử dụng gói, cổng và giao thức của Real-Time UML nhằm mô tả chi tiết các thành phần thiết kế. Các thành phần thiết kế chi tiết sau đó được chuyển đổi thành mô hình thực thi bằng cách sử dụng các nền tảng mã nguồn mở như OpenModelica để mô phỏng nhanh chóng bộ điều khiển phối hợp. Dựa theo mô hình đề xuất này, một bộ điều khiển phối hợp đã được thiết kế và mô phỏng với độ tin cậy và tính khả thi cao; nó cho phép một quadrotor UAV kết hợp với một cặp AUS/03-AUVs cỡ nhỏ thực hiện kịch bản tìm kiếm xác định trước theo cơ chế phối hợp trong khảo sát đại dương,

Từ khóa: UAV/AUS/MAUVs, Điều khiển phối hợp, Hoạt động theo đội hình, *Automate* lai, Real-Time UML.

1. Introduction

The study of oceans needs underwater vehicles such as AUS/MAUVs with concrete aims to enhance the effectiveness of civil society in economic as well as in other naval facilities, e.g. the biological discovery of ocean resources, disaster and tsunami warnings, self-operated underwater military means, etc. In fact, the AUS/MAUV development is often limited to the sensors and underwater communications, so the information processing speed, autonomy duration and zone of actions of AUS/MAUVs are also restricted. In addition, UAVs have seen unprecedented levels of growth over the last decade. Even though UAVs have been mainly used for military applications, there is a considerable and increasing interest for civilian applications. It is

* Corresponding author: Tel.: (+84) 904.255.855
Email: hien.ngovan@hust.edu.vn

postulated that UAVs will be used in the future extensively for environmental monitoring, search and rescue, etc. UAVs with their ability to travel at greater speeds and can be used to cover a large region; but they can only gather the information through the surface and cannot provide insight into the ocean life, so that needs to develop new control mechanisms and system structure for improving the mission performance. Therefore, we could build a UAV and AUS/MAUVs team, which cooperatively function in order to achieve this goal.

Starting from the above considerations, we have developed a cooperative control model, which permits a UAV combined with the AUS/MAUVs group to be deployed for performing quickly missions in the wide range of actions in order to improve the efficiency of ocean exploration and survey. In our model, the physical control structure and coordination scenarios are specified to gather the requirements of control system; Hybrid Automata's (HA) [1-3] features are specialized to model the behaviors of UAV and AUS/MAUVs coordination, as well as the real-time capsule collaboration performed by using the Real-Time UML [4] in order to indicate the detailed design model. Then, this design model is converted into the implementation model with open-source platforms such as *OpenModelica* [5] based on *Modelica* language [6] to quickly carry out the simulation model for the controller. Finally, a cooperative controller of a quadrotor UAV combined with a pair of small-scale AUS/03-AUVs was designed and quickly simulated to perform predetermined cooperative scenarios for ocean exploration and search.

2. Control configuration of a coordination of UAV and AUS/MAUVs team

2.1. Coordinated Structure and Scenarios

Fig. 1 shows out a coordinated structure for presenting the cooperative model to implement the controller of a UAV and AUS/MAUVs team. Here, the Command and Control Station (CCS) periodically requires the gathered information from the UAV and also commands the AUS/MAUVs to survey some particular regions of interest. The MAUVs carry out the exploration mission and periodically provide the information to an AUS. Then, a UAV will be flying over the AUS. Once the information is transferred from AUS to UAV, the UAV may provide a new path to the MAUVs through the AUS for exploration. The UAV periodically meets the AUS/MAUVs, collects the information, and returns to the CCS to provide the acquired information. The communication links between the CCS, UAV and AUS can be carried out by *RF XTend* combined with the Differential Global Positioning System (DGPS) [7]. Furthermore, the AUS is also considered as an acoustic navigation vehicle combined with one higher cost central AUV (Master AUV) based on DGPS Intelligent Sonobuoys (DIS) to provide several different types low-cost AUVs (Slave AUVs) with navigation information. Using the underwater DGPS concept together with a set of intelligent surface sonobuoys, the precise position of the master AUV carrying an acoustic pinger, could be estimated by the measured time of arrival of acoustic signals and the DGPS positions of sonobuoys. Hence, the AUS always conveniently moves above to the master AUV that permits the master AUV to remain inside the projected area of communication of the AUS, and to get the precise position from the AUS. With this coordinated structure and scenarios, the master AUV could get accurate position from the CCS, UAV and AUS, without coming up to the surface. The above coordinated structure and scenarios also permit the master AUV to calibrate its positions (e.g. the trajectory-tracking) which would severely disturb or even deteriorate the whole strategy of the team coordination and formation, besides the unwanted energy consumed to emerge to the surface [7].

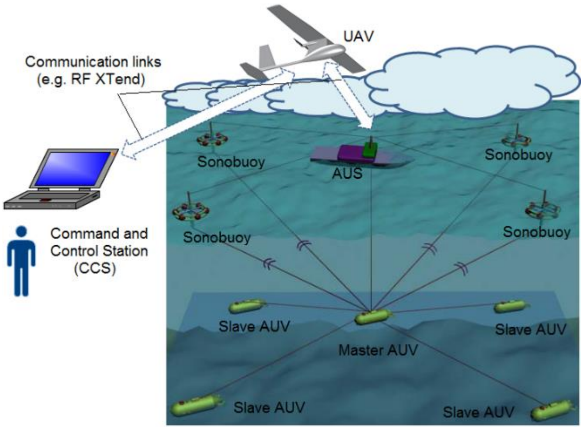

Fig. 1. Coordinated structure of a UAV and AUS/MAUVs team.

2.2. Cooperative control architecture of UAV and AUS/MAUVs team

Control systems of actual machines or actuators generally take account of models with discrete events and continuous behaviors that are called Hybrid Dynamic Systems (HDS) [2]. These behaviors are distributed on different operating modes, which are associated with processes related to the interactivity with users. Furthermore, controlled systems do not always have the same behavior because they are associated with validity hypotheses to check at any moment. In the industrial control context, a HDS can contain two parts with theirs interactions that are the HDS controller and the controlled HDS. These parts mutually exchange periodic signals and episodic events, which are either external or internal. Fig. 2 shows out the block diagram of an Industrial HDS (IHDS). Here, E_o and E_i are respectively output and input events; S_o and S_i are respectively output and input signals; ΔT is a sampling period of the evolution model for control; and $Actor_1$, $Actor_2$, …, $Actor_m$ are descriptions of a coherent set of roles that users (i.e. persons or involved external systems) play when they interact with the developed IHDS.

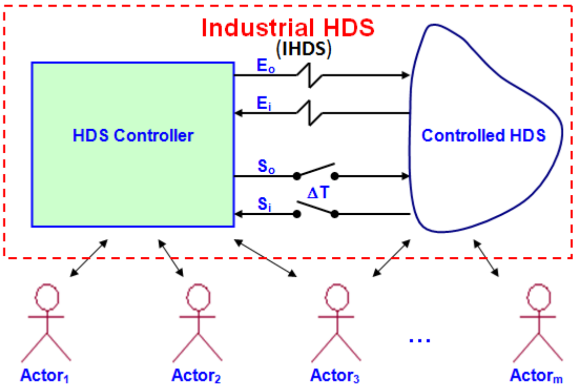

Fig. 2. Block diagram of IHDS.

From the above coordinated structure and scenarios of a UAV and AUS/MAUVs team, the dynamic models for control of the individual UAV, AUS and AUV described in [8, 9] together with the above characteristics of IHDS, we find that controllers of the UAV and AUS/MAUVs team are IHDS whose dynamic behaviors can be modeled by HA. These controllers have the *continuous/discrete* parts and their interactions such as the motional components of each vehicle in the team, the external interacting events from the CCS, guidance/navigation system and environment disturbances. The behaviors of such systems are thus complex; they can be modeled by HA [3, 10] for modeling completely requirements in the development lifecycle of these systems.

3. Model-driven development of cooperative controllers for a UAV and AUS/MAUVs team

3.1. Hybrid Control Model (HCM) for a UAV and AUS/MAUVs team

Starting from the above discussed points, the problem of coordinated UAV and AUS/MAUVs team must have a hybrid control characteristic [11, 12], which has both global discrete model combined with the coordination strategy and global continuous model issued from local *continuous/discrete* parts and their interactions related to the individual UAV, AUS and AUV.

The global continuous model of this team is generally built by considering a set $\boldsymbol{F} = \{F_1, F_2,..., F_n\}$ of $n > 3$ Autonomous Vehicles (AVs) comprising at least the *01* UAV, *01* AUS, *01* master AUV and *n-3* other slave AUVs; the dynamic properties of F_i can be not similar as that of F_j, $(i, j = 1,2,...,n)$, i.e. these n AVs also set up a heterogeneous system in the UAV and AUS/MAUVs team. The dynamic model for control of each AV can be modeled as the following nonlinear system (1).

$$F'_i(t) = f_i(F_i(t), u_i(t)) \qquad (1)$$

Here, $F_i(t)$, $u_i(t)$ and f_i are respectively the continuous state, the admissible control value or state feedback and a vector field which defines the dynamic model of the i^{th} individual AV. With the soft computing technique combined with various control laws [8, 11], AVs could arrive at the desired position from one waypoint to another.

The global discrete model of a UAV and AUS/MAUVs team can be realized by an event-based controller, which has an applicable state machine issued from the coordinated scenarios described in Section 2.1. This model generates a set $W=\{W_1, W_2,..., W_n\}$ of waypoints. The team coordination is defined and updated by the following law [12]:

$$W_i(t+1) = \Psi(W_i, t, e) \qquad (2)$$

Where: e is an event that is triggered when all AVs arrive at the desired position; t is the time step; $W(t+1)$ indicates the next value of W; finally, Ψ is the team coordination strategy, e. g. the coordinated scenarios. The control u_i is derived for the i^{th} AV based on $W_i(t)$ and $W_i(t+1)$.

An interaction between the global discrete and continuous models can be carried out by the control u_i [12] because it depends on both the continuous behaviors and the state of in the discrete model; the interaction is determined by event e as well as providing a set of coordination commands (3) corresponding to waypoints W.

$$u_i = \varphi_i(W_i, e) \qquad (3)$$

Here, φ_i is the interaction function in the team coordination strategy. It should be noted that all AVs observe the same enabling event e which is triggered when all AVs have reached their previously computed waypoints.

3.2. Hybrid Automata (HA) specialization for a UAV and AUS/MAUVs team

The evolution of the above defined HCM for a UAV and AUS/MAUVs team can be carried out by using the HA's formalism because HA has only one global continuous behavior at time given, contains the *invariant* notation to verify hypotheses on the continuous state, is derived from an automaton modeling also the dynamic behavior of interactive software systems, and can be verified with proof tools such as *HyTech*, *CheckMate* [13] and *OpenModelica* [5]. A Hybrid Automata (HA) of HCM is defined by equation (4):

$$H_{HCM} = (Q, X, \Sigma, A, Inv, \Phi, q_o, x_o) \qquad (4)$$

Where:

- Q is a set of states describing operational modes of H_{HCM}, e.g. the *System Coordination*, *Reconfiguration*, *Motion*, *Stop* and *Idle*, which are combined with a state machine issued from the coordinated scenarios (i.e. the team coordination strategy Ψ). Q can be called situations of the cooperative controller of UAV and AUS/MAUVs team; q_o is the initial situation.

- X presents the continuous state space of H_{HCM}, $X \subset \Re^n$, x_o is the initial value of this space, e.g. continuous components F_i of the HCM.

- Σ is a finite set of events, e.g. the external interacting events from the CCS and the internal event e triggered for W_i in the HCM.

- A is a set of transitions defined by (q, *Guard*, σ, *Jump*, q'). Here, $q \in Q$, $q' \in Q$; *Guard* is a subset of the state space in which the continuous state must be, so that the transition can be crossed; *Jump* represents the continuous state transformation during the change of situation; it is expressed by a state value function, whose result is affected like initial value of the continuous state in the new situation; $\sigma \in \Sigma$ presents the event being associated to the transition; this association does not imply to give an input or output direction to the event.

- *Inv* is an application for the interaction function φ_i of the HCM which associates a subset of the state space to each situation; it is called the *invariant* of the situation, in which the continuous state must remain, when the situation is q, the continuous state must verify $x \in inv(q)$.

- Φ is defined by using the global continuous model $\boldsymbol{F}$ of the HCM for each situation; the evolution of continuous state is occurred when the situation is activated.

To perform this evolution, we also introduced constraints as follows: $\sigma \in \Sigma$ are considered in term of inputs/outputs and internality/externality; X contains input/output signals. The realization hypotheses for the HA's evolution, which permit the invariant *Inv* and guard control *Guard* can generate internal events for this HCM, can be found in the author's report [3].

3.3. Implementation model of HCM for a UAV and AUS/MAUVs Team

From the authors' approach described in [3, 11], we developed the 5 main control capsules, which take part in H_{HCM} realization of a UAV and AUS/MAUVs team: the continuous part's capsule, discrete part's capsule, internal interface's capsule, external interface's capsule and Instantaneous Global Continuous Behavior (IGCB's capsule). Fig. 3 shows out the real-time communication pattern of these capsules by using the real-time UML language's convention.

Here, the discrete part's capsule contains a set of situations Q and of transitions A of H_{HCM}; The continuous part's capsule is related to continuous elements X; The IGCB's capsule contains the concrete global continuous model at time given just as Φ in H_{HCM}. In the evolution, the IGCB's capsule exchanges periodic signals with other capsules such as the discrete part's capsule, continuous part's capsule and external interface's capsule; The internal interface's capsule contains the invariant *Inv* and guard control *Guard* for generating internal events, so that the discrete part's capsule can make its own evolution by these events; The external interface's capsule is an intermediary, which receives or sends episodic events and periodic signals between the developed system and their interacted systems.

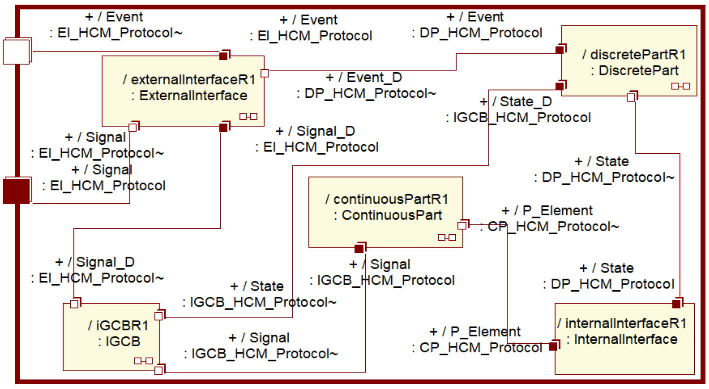

Fig. 3. Real-time communication pattern of H_{HCM} for a UAV and AUS/MAUVs team.

In this model, we use *OpenModelica* [5] tool to simulate the controller, because it is tightly based on object-oriented mechanisms and properties of *Modelica* language [6] such as the abstraction, encapsulation, modularity and heritance. Hence, we can convert the defined capsule elements into *OpenModelica* models in order to quickly simulate the functionalities and performance of this controller. This model transformation is performed by applying conversion rules, which can be seen in the authors' reports [11]. To deploy the realization model for H_{HCM} of a UAV and AUS/MAUVs team, we have to firstly update the real-time communication pattern with the control elements modified in the previous simulation model, e.g. the control law and its parameters, continuous elements, etc. Then, we convert this updated pattern into different Implementation Development Environments (IDE), which support object-oriented programming languages such as C++, Java and Ada in order to completely realize it in compatible industrial microcontrollers. This model conversion can be carried out by using object-oriented modeling software tools, which support the round-trip engineering such as *IBM Rational Rhapsody* [14].

4. Application

Following the above described model, the simulation model was completely implemented for a cooperative controller of a quadrotor UAV coordinated with a pair of small-scale AUS combining with 03 AUVs (quadrotor UAV and AUS/03-AUVs) for performing the coordination scenarios described in Section 2.1. The physical configuration parameters of each vehicle can be found in the authors' reports [11, 15, 16]. The desired coordinated control behavior in this application is MAUV flocking like birds flying in loose formations, which is useful for underwater collaborative operation. There are three basic elements to maintain MAUV flocking: (i) Cohesion: attraction to distant neighbors up to a reachable distance, (ii) Separation: repulsion from neighbors within minimal distance, (iii) Alignment: velocity and average heading matching with neighbors.

All of artifacts of the design and implementation model have been produced by using the above proposed model for simulating completely the cooperative scenarios and control performance of this team. The simulation model was performed by using *OpenModelica* [5] software in this application. Fig. 4 illustrates the velocity transients in a MAUVs flock due to the velocities of two slave AUVs in convergence corresponding to the velocity of the master AUV at 1.5m/s received from the CSS by linking with the quadrotor UAV/AUS.

All of obtained simulation results permit us to theoretically evaluate the control performance of this system within the control criteria such as the admissible timing response, transition, static errors and run-time concurrency in the team, and to evidence a good reliability of this approach. From that point, we can decide to choose the designed control elements and their properties in order to accurately implement the realization model of the above application. This realization model is actually deployed in the laboratory of mechanical and robotic systems.

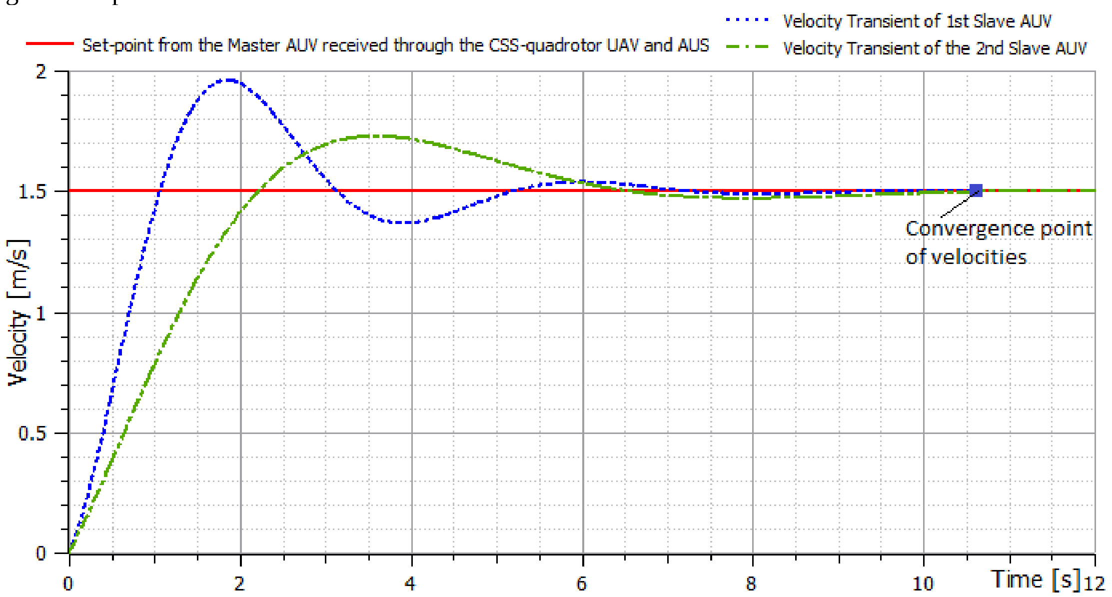

Fig. 4. Example of velocity convergence in a MAUVs flock.

5. Conclusions

The paper has presented a hybrid control model to implement a UAV and AUS/MAUVs team for performing quickly missions in the wide range of actions in order to improve the efficiency of ocean exploration and survey. This model is based on the specialization of HA's features and real-time UML to intensively capture the analysis, design and implementation phases for the cooperative controller of a UAV and AUS/MAUVs team. This study contains the following main points: The coordinated structure and scenarios of a UAV and AUS/MAUVs team are adapted to gather control requirements and to combine them with the industrial HDS (IHDS); The HA's features are specialized to model the HCM (H_{HCM}) for a UAV and AUS/MAUVs team; The main control capsules are attached to a real-time communication pattern in order to perform the object-oriented design model in detail for H_{HCM} of this system; The detailed design model is converted into the implementation model with the open-source platform of *OpenModelica* based on *Modelica* language to quickly carry out the simulation model for the controller. Finally, a cooperative controller of a quadrotor UAV combined with a pair of small-scale AUS/03-AUVs was completely designed and simulated to illustrate a good reliability of the proposed control model. Furthermore, using the approach described in this paper, development engineers will be more capable of managing the system complexity through the visual modeling of artifacts and their transformations in the development lifecycle.

In the near future, the physical realization model of the above control application will be intensively deployed and tested out detailed experimental scenarios for ocean exploration.

References

[1] Henzinger TA, Kopke PW, Puri A, Varaiya P. What's Decidable about Hybrid Automata? Journal of Computer and System Sciences, Elsevier, ISSN 0022-0000,. 1998;57:94-119.

[2] Hien NV, Soriano T. Implementing hybrid automata for developing industrial control systems. Proceedings of 8th IEEE-ETFA, ISBN 0-7803-7241-7, Volume 2. Antibes-Juan les Pins, France: IEEE; 2001. p. 129-37.

[3] Hien NV, Soriano T. A Model Transformation Process to Realize Controllers of Ship Autopilot Systems by the Specialized MDA's Features with UML/SysML. Proceedings of IEEE Conference on MECATRONICS-REM 2012, ISBN 978-1-4673-4771-6. Paris, France: IEEE; 2012. p. 20-6.

[4] OMG. UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems: OMG Formal Version: http://www.omg.org/spec/MARTE/; 2011.

[5] OpenModelica. OpenModelica software, version 1.11. OpenModelica; 2017.

[6] Fritzson P. Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach, 2nd Edition. USA: John Wiley & Sons; 2015.

[7] McGillivary P. Coordinating AUV, ASV and UAS Capabilities for Maritime Domain Awareness. CG/DHS/CIMES Persistent Surveillance Workshop, Feb16th -17th, 2012. 2012.

[8] Beard RW, McLain TW. Small Unmanned Aircraft: Theory and Practice. New Jersey 08540: Princeton University Press; 2012.

[9] Robert GN, Sutton R. Further Advances in Unmanned Marine Vehicles. London: The Institution of Engineering and Technology; 2012.

[10] Hien NV. Une Méthode Industrielle de Conception de Commande par Automate Hybride Développée en Objets. France: Thèse de Doctorat, Univertsité de Marseille III; 2001.

[11] Soriano T, Hien NV, Tuan KM, Anh TV. An object-unified approach to develop controllers for autonomous underwater vehicles. Mechatronics, Elsevier, ISSN 0957-4158,. 2016;35:54–70.

[12] Sousa JB, Johansson KH, Silva J, Speranzon A. A verified hierarchical control architecture for co-ordinated multi-vehicle operations. International Journal of Adaptive Control and Signal Processing, ISSN 1099-1115,. 2012;21:159–88.

[13] Carloni LP, Passerone R, Pinto A, Sangiovanni VA. Languages and Tools for Hybrid Systems Design. Boston: Now Publishers Inc; 2006.

[14] IBM. IBM Rational Online Documentation and Training Kit. IBM; 2017.

[15] Diem PG, Hien NV, Khanh NP. An Object-Oriented Analysis and Design Model to Implement Controllers for Quadrotor UAVs by Specializing MDA's Features with Hybrid Automata and Real-Time UML. WSEAS Transactions on Systems, E-ISSN 2224-2678,. 2013;12:483-96.

[16] Hien NV, Thai LT, DiemPG et al. Research on, design and manufacture control systems with the integration of object-oriented technology (MDA & real-time UML) and navigation units (INS/GPS) for autonomous underwater vehicles, Final scientific report of protential research project, funded by the State, code: KC03.TN05/11-15. Hanoi, Vietnam: Hanoi University of Science and Technology; 2013.