



## Research Article

# A Model-Driven Realization of AUV Controllers Based on the MDA/MBSE Approach

Ngo Van Hien ,<sup>1</sup> Van-Thuan Truong ,<sup>1</sup> and Ngoc-Tam Bui ,<sup>2,3</sup>

<sup>1</sup>School of Transportation Engineering, Hanoi University of Science and Technology, Hanoi 10000, Vietnam

<sup>2</sup>School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi 10000, Vietnam

<sup>3</sup>College of Systems Engineering and Science, Shibaura Institute of Technology, Tokyo 135-8548, Japan

Correspondence should be addressed to Ngo Van Hien; [hien.ngovan@hust.edu.vn](mailto:hien.ngovan@hust.edu.vn), Van-Thuan Truong; [thuan.truongvan@hust.edu.vn](mailto:thuan.truongvan@hust.edu.vn), and Ngoc-Tam Bui; [tambn@shibaura-it.ac.jp](mailto:tambn@shibaura-it.ac.jp)

Received 2 March 2020; Revised 2 October 2020; Accepted 10 October 2020; Published 26 October 2020

Academic Editor: Rocío de Oña

Copyright © 2020 Ngo Van Hien et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper introduces a model-driven control realization, which is based on the systems engineering concepts of the model-driven architecture (MDA)/model-based systems engineering (MBSE) approach combined with the real-time UML/SysML, extended/unscented Kalman filter (EKF/UKF) algorithms, and hybrid automata, in order to conveniently deploy controllers of autonomous underwater vehicles (AUVs). This model also creates a real-time communication pattern, which can permit the designed components to be customizable and reusable in new application developments of different AUV types. The paper brings out stepwise adapted AUV dynamics for control that are then combined with the specialization of MDA/MBSE features as follows: the computation independent model (CIM) is defined by the specification of the use-case model together with hybrid automata to gather the requirement analysis for control; the platform-independent model (PIM) is then designed by specializing the real-time UML/SysML's features including main control capsules that depict structures and behaviors of controllers in detail; and the detailed PIM is subsequently converted into the platform-specific model (PSM) by object-oriented platforms to rapidly implement the AUV controller. Based on this proposed model, a horizontal planar trajectory-tracking controller was deployed and tested that permits a miniature AUV possessing a torpedo shape to reach and follow the desired horizontal planar trajectory.

## 1. Introduction

Autonomous underwater vehicles (AUVs) are increasingly developed and used for the study of oceans 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, and self-operated underwater military means [1–6].

Up to date, many control applications of AUVs have been implemented by the traditional guidance, navigation, and control approach combined with soft-computing methods [7–14] to closely deal with their dynamic models for control. For example, Rodriguez et al. [15] introduced a design model of adaptive sliding mode control (SMC) for a micro-AUV to reject bounded perturbations, where the finite-time convergence using the control Lyapunov

function (CLF) was performed. This proposed control allowed achieving low-level tasks such as velocity control and regulation, as well as high-level tasks such as autonomous path-following using waypoints (WP). A robust adaptive integral SMC technique has also been developed by Yan et al. [16] for the trajectory tracking control of underactuated AUVs in the 5 degrees of freedom (DoF), where the vehicle dynamics were unknown and also subjected to environmental disturbances. The dual closed-loop control design in this study ensures that the vehicle's dynamics are all consistent with the reference "state-space" trajectory including the time evolutions of position and attitude, as well as velocities. A hierarchical robust nonlinear (HRN) controller was designed by Karkoub et al. [17] for the horizontal planar trajectory tracking of an AUV subject to uncertainties (e.g., current disturbances, unmodeled

dynamics, and parameter variations); in this study, the proposed HRN controller was used with the integral backstepping (IB) and SMC techniques enclosed with the control Lyapunov function (CLF). The robustness of the proposed HRN controller was then verified through injection of random uncertainties into the system model, and the closed-loop stability of the proposed individual subsystems was, respectively, guaranteed to have uniformly ultimately bounded performance. Furthermore, both extended Kalman filter (EKF) and unscented Kalman filter (UKF) algorithms have been experimentally evaluated offline by means of sea navigation tests performed by [18, 19] for estimating the dynamic behavior of the AUV and improving the performance of the navigation system for the vehicle.

Besides, the low development costs must be taken into account in the application construction. Thus, the reusability is also needed to be considered in the new development lifecycle of various AUV applications. According to the Object Management Group (OMG) [20], the Unified Modeling Language (UML) has been standardized for analyzing and designing system components visually in the software industry. In addition, the System Modeling Language (SysML) [21] is a UML profile for systems engineering that has also been standardized by the OMG. SysML is used for the analysis, design, verification, and validation of industrial systems in various domains. Nevertheless, both UML and SysML lack the constructs for modeling precisely timing and communication evolutions between the control objects for real-time and embedded systems, e.g., the AUV controller. Furthermore, OMG has also standardized the model-driven architecture (MDA) [22] whose main idea is to separate the specification of system operations from the details of the way that system uses the capabilities of its platform. MDA supports a framework for and enables tools to be provided for specifying a system independently of the platform that supports it; specifying platforms; choosing a particular platform for the system; and transforming the system specification into one for a particular platform. On the contrary, the model-based systems engineering (MBSE) has been formalized by INCOSE [23–25] to support the modeling of system requirements, design, analysis, verification, and validation of artifacts in the development lifecycle of complex systems. Thus, MDA defines a general system approach, which focuses on the use of rigorous visual modeling techniques throughout the whole of development lifecycle and provides a skeletal solution, whose details can be the MBSE specialization to systems engineering applications. The foundation of MDA actually enables the Unified Architecture Framework (UAF) [26], which can evolve to meet future needs as the leverage cross-industry and standards-based approaches (e.g., UML/SysML), to enhance tool and architecture data interoperability. This provides a means to develop an understanding of the complex relationships that exist between organizations, systems, and systems-of-systems and can permit the analysis of these systems to ensure that they meet the expectations of the user community. The applications of real-time UML/SysML and model-driven techniques for the implementation of industrial control systems can be found in [27–34].

The above considered points have inspired us to find a specialization of the MDA/MBSE features together with the real-time UML/SysML [21, 35–37], which can be used to systematically model the analysis, design, and implementation of artifacts for developing an AUV controller. In this study, the AUV dynamics for control are specified for the requirement inputs that are then implemented with the specialization of MDA/MBSE features composed of the computation independent model (CIM), platform-independent model (PIM), and platform-specific model (PSM). Here, the CIM consists of the use-case model defined closely with an implemented function block diagram, the supplemented EKF/UKF algorithms, and the hybrid automata (HA) features to depict in detail the analysis phase for an AUV controller; the PIM is designed from the defined CIM by specifying the real-time UML/SysML to capture the real-time control capsules with their ports, protocols, and their intercommunication evolutions in detail for AUV controllers. The designed PIM elements are then transformed into PSMs by open-source platforms such as Arduino [38] in order to rapidly implement the AUV controller. Finally, a horizontal planar trajectory-tracking controller of an application of a miniature AUV possessing a torpedo shape was completely deployed and tested. In this work, we followed the method of Hien et al. [39], presenting an object-oriented systems engineering viewpoint for controllers of micro-unmanned aerial vehicles. The main contributions of this paper can be summarized as follows:

- (1) The MDA/MBSE approach, in combination with the real-time UML/SysML, implemented functional block diagram, EKF/UKF algorithms, and hybrid automata, is specialized to intensively develop the AUV controller
- (2) The rule specializations permit the main designed control capsules to be customized and reused for various AUV types
- (3) A planar trajectory-tracking controller of a low-cost torpedo-shaped AUV is developed and tested

The paper is structured as follows: the AUV dynamics and control structure are introduced in the second section. The third section presents the details of MDA/MBSE-driven development to intensively realize AUV controllers, including the CIM, PIM, and PSM components. Following this proposed model, in the fourth section, it is applied to a case study of a miniature AUV possessing a torpedo shape. Conclusion and future work are reported in the final section.

## 2. AUV Dynamics and Control Structure

**2.1. Modeling the AUV Dynamics for Control.** The six motions of an underwater vehicle defined as surge, sway, heave, roll, pitch, and yaw [40] are shown in Table 1.

From the large field of guidance, navigation, and control of underwater vehicles [4, 31, 41–45], the 6-degree-of-freedom (DoF) models in the body frame considered in this study use the following representation:

TABLE 1: SNAME notations for underwater vehicles.

Degree of freedom	Motions	Force and moment	Linear and angular velocity	Position and Euler angles
1	Surge	X	U	x
2	Sway	Y	V	y
3	Heave	Z	W	z
4	Roll	K	P	$\phi$
5	Pitch	M	Q	$\theta$
6	Yaw	N	R	$\psi$

$$\begin{cases} \dot{\eta} = J(\eta), \\ M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau(v, u), \end{cases} \quad (1)$$

where  $\eta = [\eta_1^T, \eta_2^T]^T$  and  $v = [v_1^T, v_2^T]^T$  are vectors of position/Euler RPY angles and linear/angular velocities, respectively:  $\eta_1 = [x, y, z]^T$ ,  $\eta_2 = [\phi, \theta, \psi]^T$ ,  $v_1 = [u, v, w]^T$ , and  $v_2 = [p, q, r]^T$ ; the model matrices  $M$ ,  $C(v)$ , and  $D(v)$  denote inertia, Coriolis, and damping, respectively, while  $g(\eta)$  is a vector of generalized gravitational and buoyancy forces. Similarly,  $\tau(v, u)$  is the vector of resultant force and moment acting on the underwater vehicle, and  $u$  is the control input.

A model of state-space discreteness is needed to model the control evolution of the AUV that is used to estimate the motion states of the vehicle by using digital filters, e.g., the EKF and UKF; the control evolution of the system can then be written by the following equation:

$$\begin{cases} x_k = f_{k-1}(x_{k-1}, u_{k-1}) + w_{k-1}, \\ y_k = h_k(x_k) + v_k. \end{cases} \quad (2)$$

Here,  $x = \begin{bmatrix} \eta \\ v \end{bmatrix}$ ;  $x_k$  is the vector of state variables at the  $k^{\text{th}}$  instant of  $x$ ;  $u_k$  and  $y_k$  are, respectively, the inputs and outputs of the system;  $h_k$ ,  $w_k$ , and  $v_k$  are, respectively, the measurement function, additive process, and measurement noise.

According to AUV dynamic model (1) together with the following assumptions, the lateral and the vertical motions are ignored because we are only interested in developing the AUV possessing a torpedo shape in this study, the AUV dynamics can only be considered on the longitudinal direction; the timing evolution of control is then rewritten by the following equation:

$$\dot{x} = \begin{bmatrix} \dot{\eta} \\ \dot{v} \end{bmatrix} = \left( \begin{array}{c} \dot{\eta} = J(\eta_2)v \\ M^{-1}(\tau(v, u) - C(v)v - D(v)v - g(\eta)) \end{array} \right) + w. \quad (3)$$

**2.2. Specifying the Control Structure of an AUV.** There are three main systems within the physical structure of AUVs as follows: the guidance system is used for providing the desired path for the AUV to track; the navigation system is responsible for executing an estimation of the current state of the AUV; and the control system is used for calculating and applying the control forces and moments to conduct this AUV. These subsystems have themselves missions to be

accomplished yet must also have to cooperatively operate for allowing an AUV to fulfill its tasks even in the presence of unknown environmental disturbances. Figure 1 shows a general block definition diagram in SysML, which captures how these subsystems interact.

In addition, the problem of control design for AUVs is equally challenging because the controllers are tightly linked with the AUV dynamics. Hence, these control systems of AUVs must take into account evolution models combined with discrete events and continuous behaviors that can be called hybrid dynamic systems (HDS) [46, 47]. Besides, controlled systems do not always have the same behavior because they are associated with validity hypotheses to check at any moment; the security requirement forces to envisage events and behaviors different from nominal behaviors. Hence, the behaviors of such systems are complex and can be modeled by HA [48]. In this study, we are interested in implementing the horizontal planar trajectory-tracking controller of AUVs, so this hybrid dynamic model can be used to find out the control algorithms combined with a specific guidance law such as the line-of-sight (LOS) guidance implemented in [49–51].

### 3. MDA/MBSE-Driven Development for AUV Controllers

As previously stated, the CIM, PIM, and PSM components of MDA/MBSE combined with the real-time UML/SysML [21, 35–37] are specialized for analyzing and designing in detail control components of complex control systems, e.g., the AUV controller. Following the above described AUV dynamics and general control structures and real-time UML/SysML specifications, we specialize in detail the MDA/MBSE approach for deploying the AUV controller that consists of the 3 models as follows, CIM, PIM, and PSM, to separate the specification of the operation of the system from the details of the way that system uses the capabilities of its platform. All artifacts and activities in the CIM, PIM, and PSM for an AUV controller will be given in detail in the next sections.

**3.1. Defining the Computation Independent Model (CIM) for an AUV Controller.** Following the AUV dynamics and control structure adapted in the second section together with LOS guidance [49–51], we present here the main use-case model (Figure 2) of AUV controllers combined with an example of trajectory-tracking scenarios and local state machine of the “track a desired trajectory” use case, which

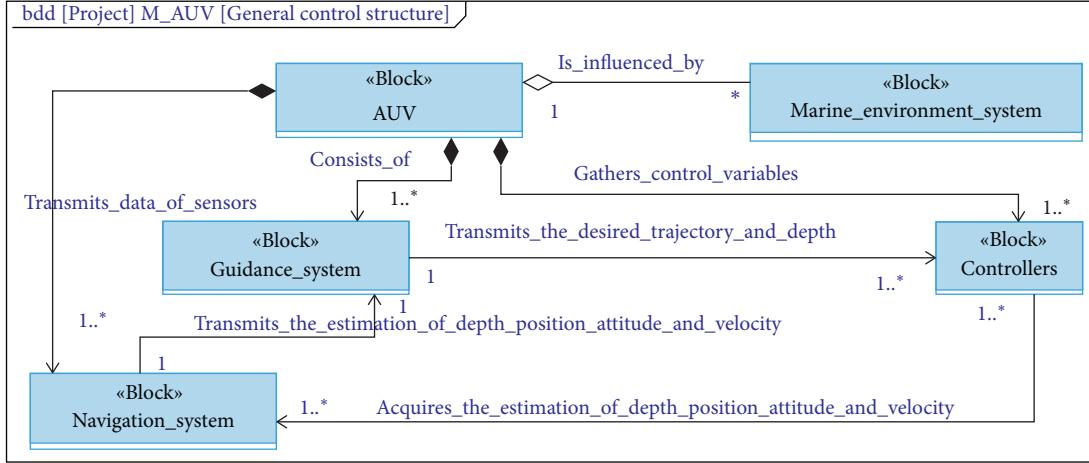


FIGURE 1: Block definition diagram of autonomy architecture for an AUV.

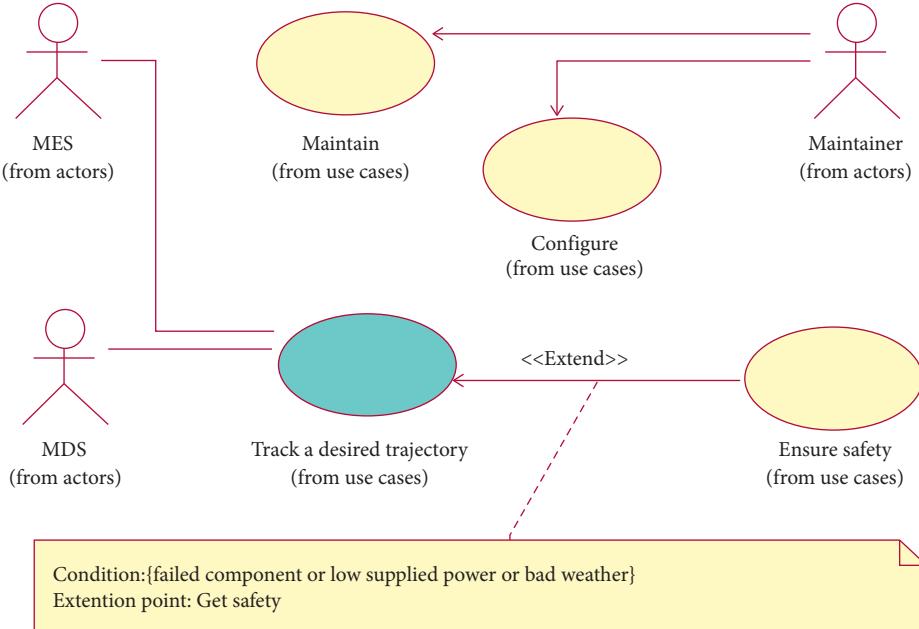


FIGURE 2: Use-case diagram for capturing the main control requirements of the AUV.

are, respectively, shown in Figures 3(a) and 3(b). In the use-case model, it is necessary to set up industrial maneuvering constraints, e.g., the maximum swing angles of rudders, velocity, and submersing depth of the AUV in order to closely ensure the system safety.

In Figure 2,

- (i) MDS is the measurement-cum-display system consisting of the guidance and navigation systems because both of them essentially act as a signal supplier for the controllers of the AUV
- (ii) MES is the marine environment system including disturbances such as the wind, waves, and ocean currents
- (iii) The “track a desired trajectory” use case is performed for tracking the desired trajectory for the vehicle to follow

- (iv) The “ensure safety” use case is realized for maintaining system safety when one of its component fails or supplied power is low or when the weather is bad
- (v) The “configure” use case permits users or maintainers to configure and update control parameters for starting up the controllers
- (vi) The “maintain” use case is performed for maintaining the whole system that includes activities, e.g., the error identifications and corrections of the whole of physical AUV or periodical maintenance

In addition, an implementing functional block diagram must be built for modeling continuous dynamics of the AUV controller because the real-time UML/SysML lacks the constructs for depicting internal continuous models for each state on the state machine diagram. Thus, we propose an

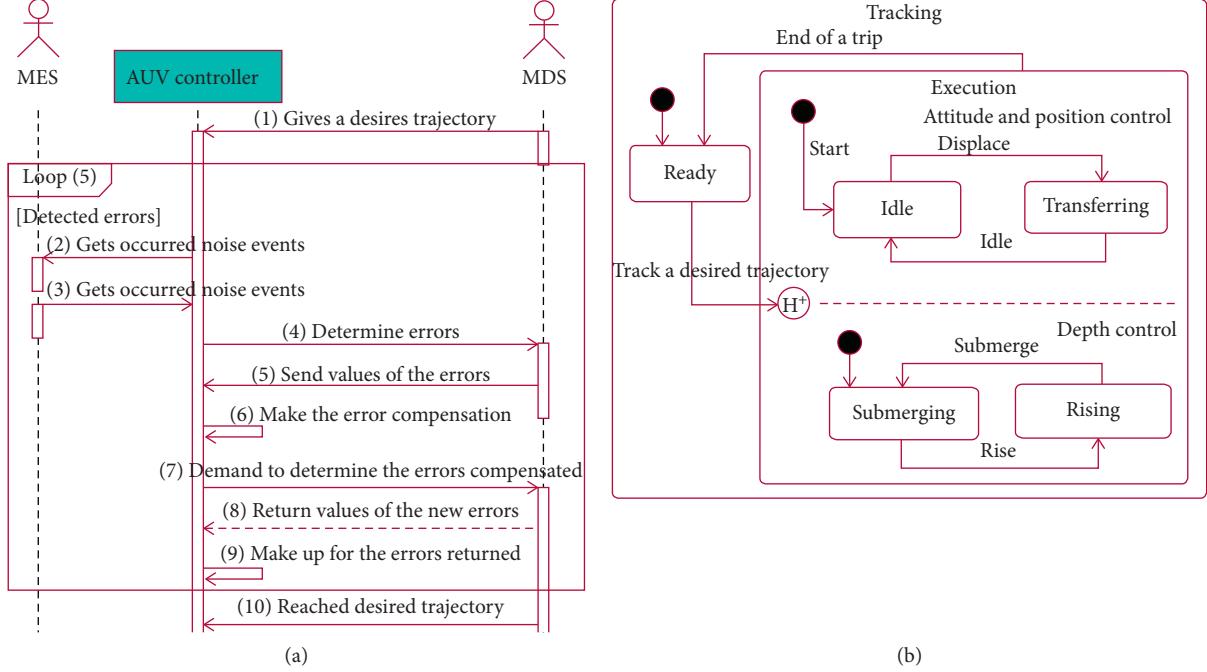


FIGURE 3: A desired trajectory-tracking scenario (a) and local state machine for performing the “track a desired trajectory” use case (b).

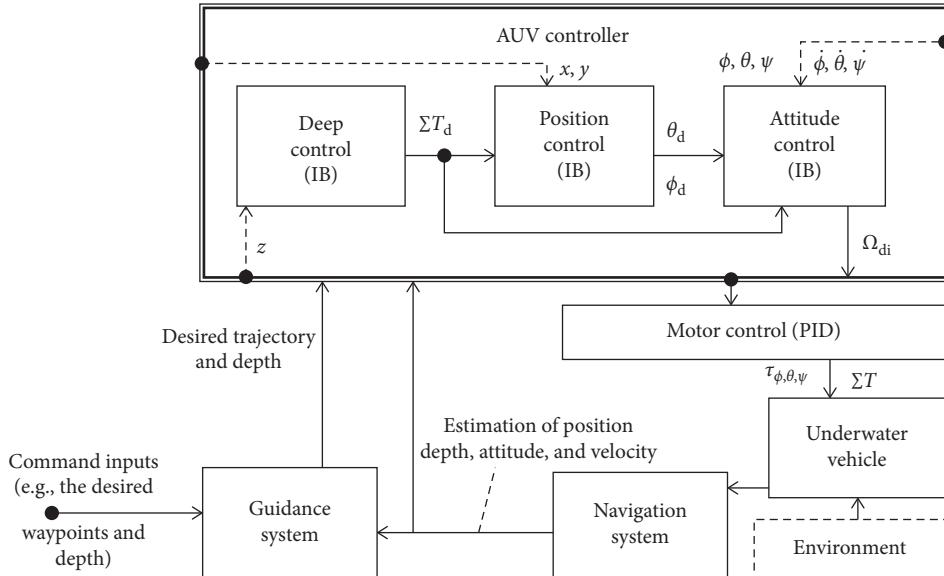


FIGURE 4: Proposal of a functional block diagram for implementing the continuous behaviors of the AUV controller.

implemented functional block diagram to gather the internal continuous behaviors of the AUV controller as shown in Figure 4.

In Figure 4,

- (i) Desired trajectory and depth are events, which are used for providing, respectively, the desired position and depth to the control blocks of positions and depths
- (ii)  $\Omega_{di}, i = \overline{1, n}$ , are desired rotational speeds, which are applied to the  $n$  actuators installed on the AUV, e.g.,

the blocks of motor control for the propellers and rudders

- (iii)  $\Sigma T$  and  $\tau_{\phi, \theta, \psi}$  are the overall output forces and moments acting on the actuators of the AUV

In this study, the integral backstepping (IB) methods of control enclosed with the CLF [42, 45, 52] are also applied to the control blocks of depth, position, and attitude for the AUV. The discrete state-space models in equations (2) and (3) are implemented in the navigation system block for estimating the position, depth, attitude, and velocity

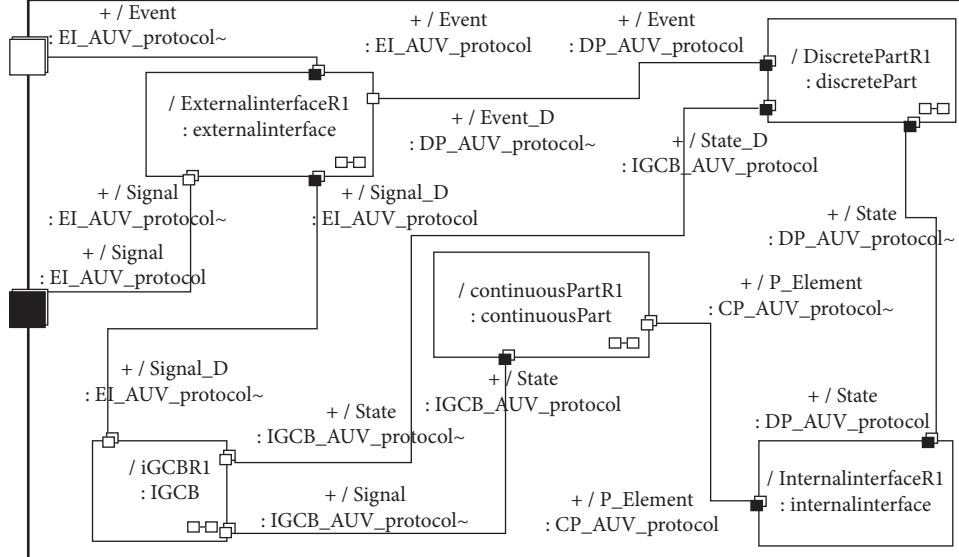


FIGURE 5: Capsules' collaboration of the real-time communication pattern for the AUV controller.

corresponding to the sensors installed on the AUV. The standard navigation filters are based on the EKF or UKF [19, 53, 54] algorithms. The application details of EKF/UKF algorithms for an AUV controller will be presented in the fourth section.

In the CIM, hybrid automata (HA) are specified to present the mathematical implementation model of the AUV controller consisting of terms such as the situations, continuous state variables, event, transition, global continuous behavior, and invariants. A HA of the AUV controller ( $\mathbf{H}_{\text{AUV}}$ ) is established by data as follows:

$$\mathbf{H}_{\text{AUV}} = (\mathbf{Q}, \mathbf{X}, \sum, \mathbf{A}, \mathbf{Inv}, \mathbf{F}, \mathbf{q}_0, \mathbf{x}_{c0}). \quad (4)$$

Here,  $\mathbf{Q}$  is a set of operation modes called the situations of  $\mathbf{H}_{\text{AUV}}$ , e.g., the modes of idle, transferring, submerging, and rising;  $\mathbf{q}_0 \in \mathbf{Q}$  is the initial situation;  $\mathbf{X}$  is the continuous state space of continuous elements of the AUV, e.g., the control blocks of motor, position, and depth;  $\mathbf{x}_{c0}$  is the initial value of this space;  $\sum$  is a set of external interacting events for triggering from the current situation to the reached situation, which is issued from the guidance and navigation systems and environmental disturbances for the AUV controller;  $\mathbf{A}$  is a set of transitions between the situations that are linked to the appropriate events  $\sigma \in \sum$ ;  $\mathbf{Inv}$  is called the invariant term of the situation for verifying when the situation is  $\mathbf{q}$ , the continuous state must then be  $\mathbf{x}_c \in \mathbf{inv}(\mathbf{q})$ ; and  $\mathbf{F}$  is the global continuous models issued from by 6-DoF dynamic model (1) and the implemented functional block diagram (Figure 4); the global evolution of continuous state is performed when a new situation is reached in the operation modes of the AUV. The detail of HA specialization and its realization hypotheses for the AUV controller can be seen in the authors' report [55].

### 3.2. Designing the Platform-Independent Model (PIM) for an AUV Controller.

From the above defined CIM, the 5 main

control capsules have been specialized to participate in the HA evolution of the AUV controller: 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). Figures 5 and 6 show the capsules' collaboration of the real-time communication pattern for the AUV controller by using the real-time UML collaboration and sequence diagrams.

Here, the discrete part's capsule consists of the situations  $\mathbf{Q}$  and transitions  $\mathbf{A}$  in HA of the AUV controller; the continuous part's capsule contains the continuous state-space  $\mathbf{X}$ ; the IGCB's capsule builds up concrete global continuous behaviors as  $f \in F$ ,  $f$  is derived from equations (1) and (2) and the implemented functional block diagram (Figure 4), and the EKF/UKF algorithm can be implemented in  $f$  for estimating the values of position, depth, attitude, and velocity of the AUV; the internal interface's capsule permits the  $\mathbf{Inv}$  tool to generate internal events in the HA evolution; and the external interface's capsule is an intermediary, which receives or sends episodic events and periodic signals between the AUV controller and its interacting systems such as MES and MDS.

The timing concurrency of evolutions for the above capsules' collaboration of the real-time communication pattern has been described in detail in the authors' reports [31, 55]. In addition, the reusability is very important to implement controllers for various AUV types because it allows reducing the cost of application deployments. The specializations, which permit the capsule collaboration of a developed AUV to be customizable and reusable in the new control application for various AUV types, can be summarized in Table 2.

### 3.3. Implementing the Platform-Specific Model (PSM) for an AUV Controller.

To carry out the AUV controller, the PIM is implemented to the PSM that is transformed from the above built PIM by using tools such as IBM Rational Rose

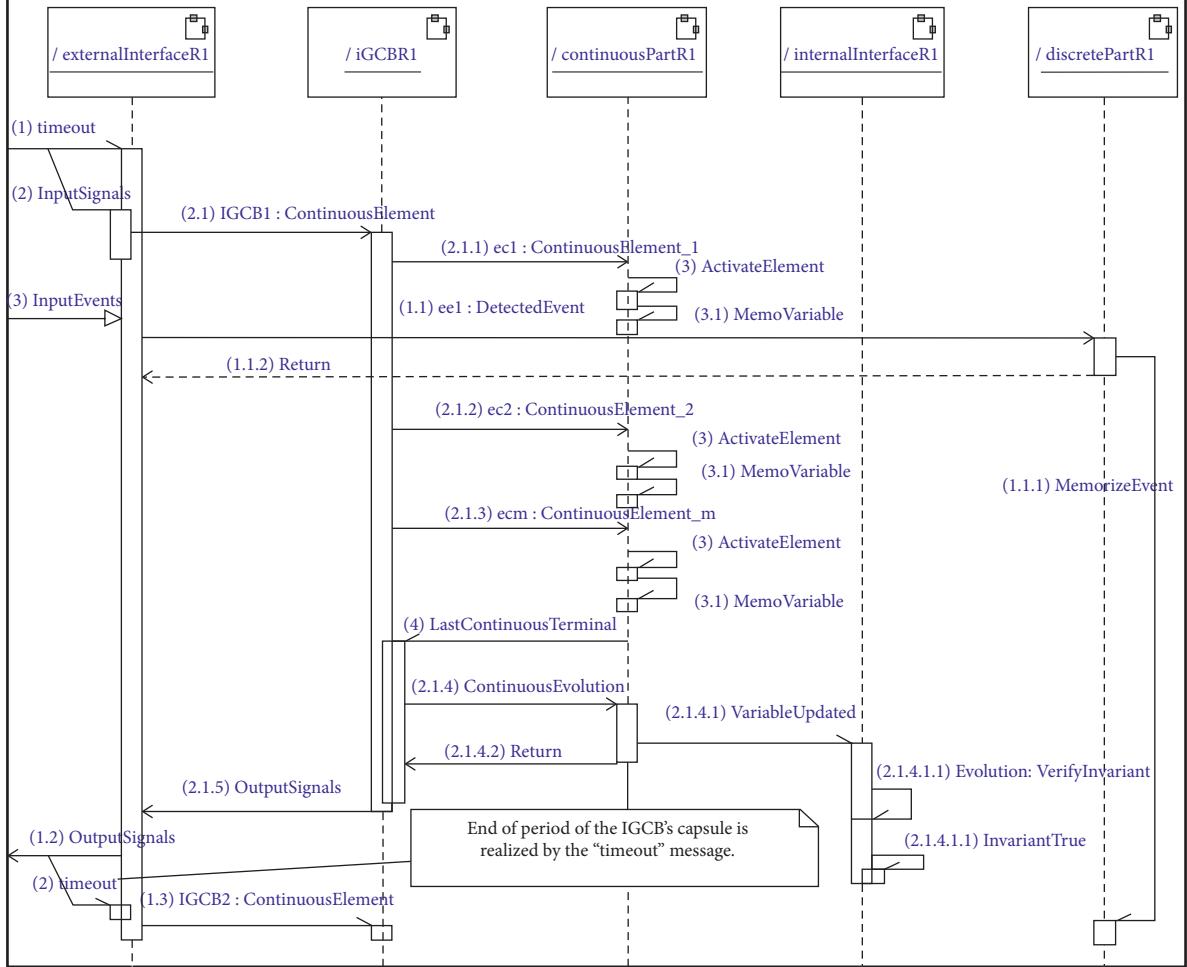


FIGURE 6: Sequence diagram of the real-time communication pattern for the AUV controller.

RealTime or IBM Rational Software Architect RealTime [56] or Papyrus for Real Time (Papyrus-RT) [57]. Here, IBM Rational's leading role in defining the real-time UML is widely acknowledged as it is the preeminence of the IBM Rational Rose RealTime or IBM Rational Software Architect RealTime products for supporting the real-time UML design of industrial software systems. It combines a rich modeling environment with a code-oriented tool set to make up a comprehensive practitioner desktop for creating solutions in a variety of architectural styles and targeted at specific runtime infrastructures. Besides, Papyrus-RT is an industrial-grade, complete open-source modeling environment for the development of complex, software-intensive, real-time, embedded, cyber-physical systems; it provides an implementation of the real-time UML together with editors, code generator for C++, and a supporting runtime system. Many other important lifecycle artifacts also benefit from these tools (e.g., requirement lists, test cases, and build scripts) to entirely cover development phases for the AUV controller.

Hence, the above designed PIM can be subsequently transformed into the PSM (i.e., the realization model) by using various object-oriented implementation development

environments (IDEs) in order to entirely deploy the AUV controller with compatible embedded microcontrollers. In fact, this transformation model is quickly performed through the round-trip engineering (i.e., the forward and reverse engineering) of intermediate programming codes, e.g., the C++ codes, which are issued from the design models depicted in IBM Rational Rose RealTime or IBM Rational Software Architect RealTime or Papyrus-RT tools.

Furthermore, the above defined HA can be automatically implemented by using the state pattern described in [58, 59]. The state pattern is in either of the following cases: (i) an object's behavior depends on its state, and it must change its behavior at runtime depending on that state. (ii) Operations have large, multipart conditional statements that depend on the object's state. This state is usually represented by one or more enumerated constants. Often, several operations will contain this same conditional structure. The state pattern puts each branch of the conditional in a separate class. This lets us treat the object's state as an object in its own right that can vary independently from other objects. Following this pattern, an implementation structure of HA is specified as shown in Figure 7 to improve the evolution of the AUV controller.

TABLE 2: Main control capsules of the PIM can be customized and reused in the new control application for various AUV types.

Designed control capsules	Generic artifacts	Specialization rules	Specialized artifacts
Discrete capsule	The discrete part's capsule is not changed in the overall design for the new AUV controller		None.
Continuous part	The ports and protocols of this capsule are not changed in the overall design for the new AUV controller		The continuous part's capsule is specialized by supplementing or cutting down continuous components ( $x_c \in X$ ) that depend on the physical actuators installed on the new AUV. The states and their behaviors, which correspond to the supplemented/cut down continuous elements, are supplemented/cut down in/from the state machine of this capsule. The behavior of the new set of continuous elements will be used to redefine the concrete IGCBs ( $f \in F$ ).
IGCB	The state machine, ports, and protocols of this capsule are not changed in the overall design for the new AUV controller		The specification of the IGCB's capsule captures new IGCB model, which is formed by restructuring the new set of continuous elements according to the implemented functional block diagram.
Internal interface	The state machine and ports of this capsule are not changed in the overall design for the new AUV controller		The specialization of the internal interface's capsule is performed by supplementing/cutting down in/from the new IGCB in the IGCB's capsule if necessary new Inv term that corresponds to new supplemented/cut down situations in/ from the discrete part's capsule of application.
External interface	The state machine, ports, and protocols of this capsule are not changed in the overall design for the new AUV controller		The external interface's capsule is specialized by supplementing or cutting down input or output events, which are issued from outside (i.e., supplementing/cutting down these events in/from its protocols)
The communication pattern shown in Figures 5 and 6 is not changed in the overall design for new controllers of various AUV types.			

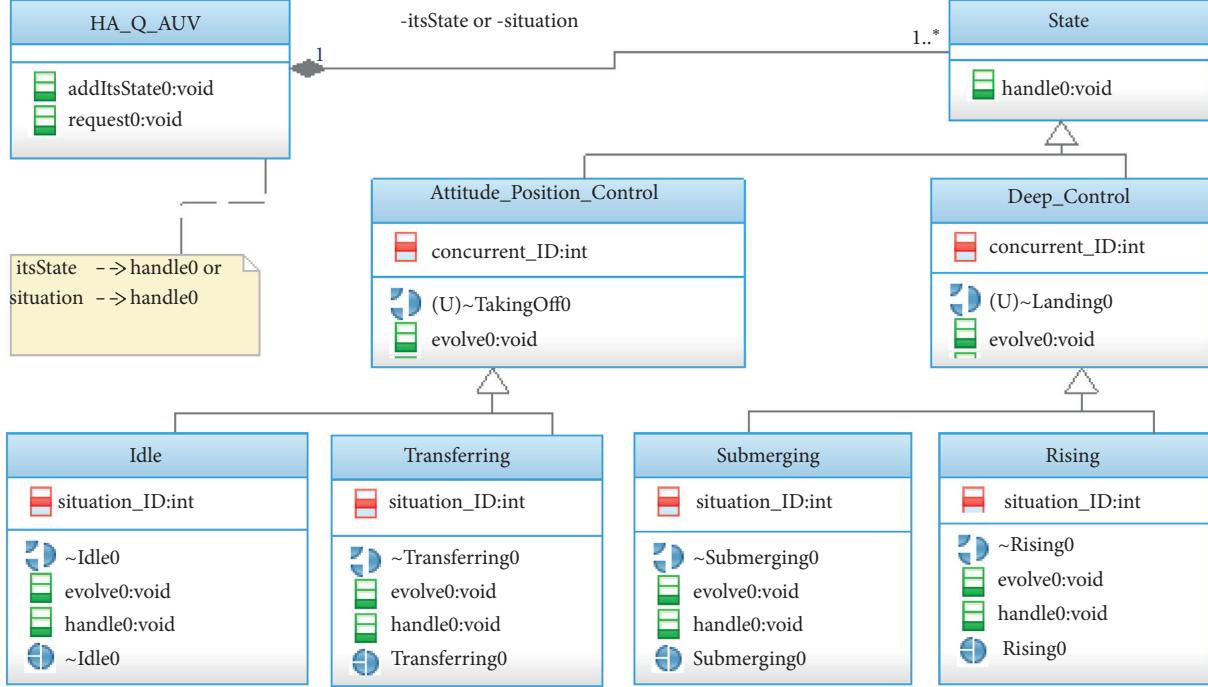


FIGURE 7: State pattern structure of HA for the AUV controller.

Here, software tools such as IBM Rational Rose RealTime or IBM Rational Software Architect RealTime or Papyrus-RT can support multiple transformations between design models and various object-oriented programming languages.

#### 4. An Application

Based on the above proposed hybrid control model, we deployed a horizontal planar trajectory-tracking controller, which permits a miniature AUV (M-AUV) possessing a

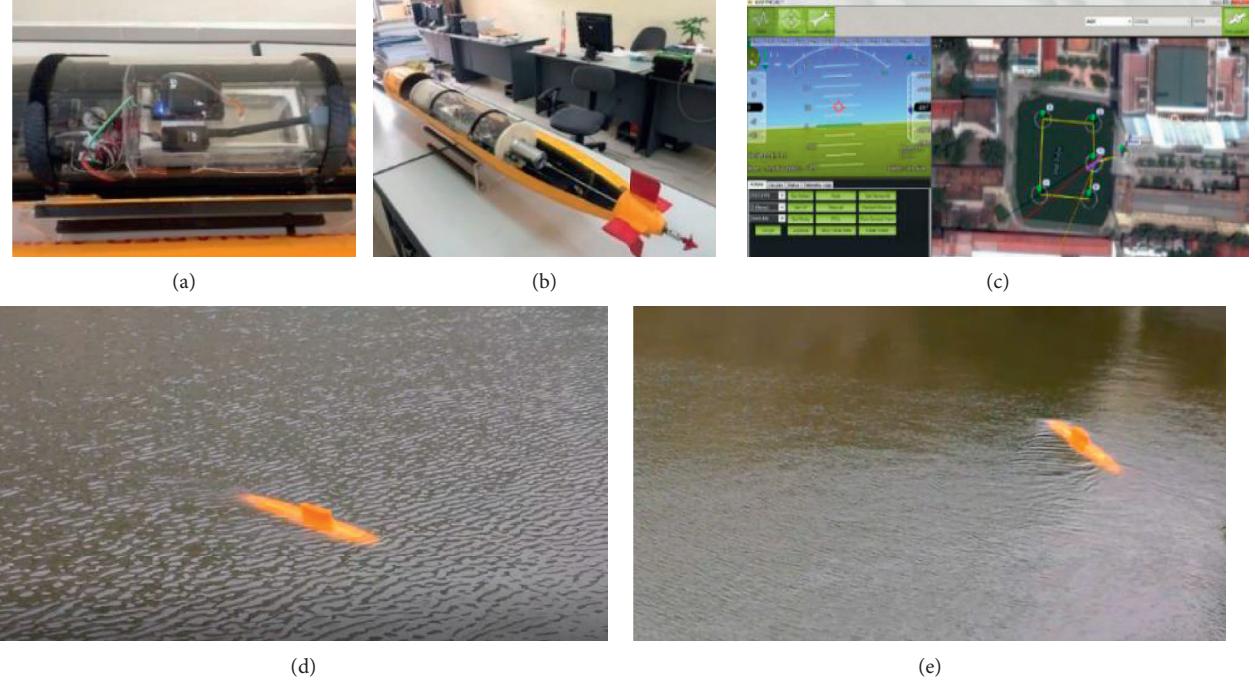


FIGURE 8: Installation and trial trips for the M-AUV controller.

torpedo shape with the small navigation bridge (Figure 8) to reach and follow a desired horizontal planar trajectory. The main operational parameters of the M-AUV are summarized in Table 3.

In the case study, the M-AUV dynamic model described in equation (1) is used with the following assumptions: the damping matrix is assumed as a diagonal matrix, which permits the coupling between dissipative effects to be neglected because the velocities of the M-AUV are not too high; the mass matrix is considered as diagonal that is derived from the body reference frame aligning with the vehicle principal axes of inertia for the M-AUV.

The state-space models described in equations (2) and (3) are implemented for estimating the position, depth, attitude, and velocity corresponding to the sensors installed on the M-AUV, such as the inertial measurement unit (IMU): MPU6000 with working frequency 100 Hz [60] and GPS: Ublox Neo 6M with working frequency 10 Hz [61]. These state estimations and predictions are implemented in the two cases by the navigation filters based on the EKF (Algorithm 1) and the UKF (Algorithm 2). In Algorithms 1 and 2,  $\hat{x}$  denotes an estimate;  $P$  is the state covariance;  $Q$  and  $R$  are, respectively, the covariance matrices of process and measurement noise, assumed as zero-mean stationary white noises with zero cross-correlation; the state is recursively estimated starting from the assumed initial conditions as follows:  $\hat{x}_{0/0} = x_0$  and  $P_{0/0} = 0_{12 \times 12}$ . The unscented transform (UT) is a deterministic sampling technique, which allows us to compute the mean and the covariance matrix of a random variable; it undergoes a generic nonlinear transformation by propagating a minimum set of its samples and exploiting the knowledge of the mean and the covariance of the starting variable.

TABLE 3: Main operational parameters of the M-AUV.

Parameters	Values
Size (L × H × W)	(1.54 × 0.27 × 0.25) m
Net dry weight	12.30 kg
Autonomous duration	20 min
2 × Li-Po battery	22.2 V; 20,000 mAh
Ultimate capacity	314 W
Maximum submerging/rising speed	0.65 m/s
Maximum horizontal transferring speed	1.82 m/s
Maximum diving depth	1.15 m
Maximum radius of action	390 m

In this case study, ATMEGA32-U2 and STM32-Cortex-M4 microcontrollers [38] have been used on the main board, and the control program is installed based on C++ language. An example of instantiation and use of the generic HA classes based on the above-specified state pattern (Figure 7) is implemented as Figure 9 for the idle/transferring state pair of the M-AUV controller.

The experimental scenarios of the M-AUV controller were based on various desired course angles and shape-based planar trajectories and mean transferring speeds that were performed on Ho Tien basin at Hanoi University of Science and Technology, Hanoi, Vietnam (Figure 8). Table 4 illustrates the main obtained test results for the course tracking of the M-AUV. Figures 10(a) and 10(b), respectively, indicate the horizontal planar trajectory-tracking controller with the EKF and UKF algorithms that permits the M-AUV to reach and follow the desired rectangle-shaped trajectory. Table 5 is the comparison of root mean square (RMS) and peak values for the path tracking error between the two cases using the EKF and UKF algorithms.

```

Function EKF algorithm
Step EKF predict
  Data:  $\hat{x}_{k-1|k-1}, P_{k-1|k-1}, f_{k-1}(\cdot)$ 
  Result:  $\hat{x}_{k|k-1}, P_{k|k-1}$ 
   $F_{k-1} = (\partial f_{k-1}/\partial x)(\hat{x}_{k-1|k-1} u_{k-1});$ 
   $\hat{x}_{k|k-1} = f_{k-1}(\hat{x}_{k-1|k-1});$ 
   $P_{k|k-1} = F_{k-1}, P_{k-1|k-1} F_{k-1}^T + Q_{k-1};$ 
end
Step EKF update
  Data:  $\hat{x}_{k|k-1}, P_{k|k-1}, h_k(\cdot)$ 
  Result:  $\hat{x}_{k|k}, P_{k|k}$ 
   $H_k = (\partial h_k/\partial x)|\hat{x}_{k|k-1};$ 
   $S_k = R_k + H_k P_{k|k-1} H_k^T;$ 
   $L_k = P_{k|k-1} H_k^T S_k^{-1};$ 
   $e_k = y_k - h_k(\hat{x}_{k|k-1});$ 
   $\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k e_k;$ 
   $P_{k|k} = P_{k|k-1} - L_k S_k L_k^T;$ 
end

```

ALGORITHM 1: Navigation filter based on EKF.

```

Function UKF algorithm
Step UKF predict
  Data:  $\hat{x}_{k-1|k-1}, P_{k-1|k-1}, f_{k-1}(\cdot)$ 
  Result:  $\hat{x}_{k|k-1}, P_{k|k-1}$ 
   $(\hat{x}_{k|k-1}, \bar{P}_{k|k-1}) = UT(\hat{x}_{k-1|k-1}, \bar{P}_{k-1|k-1}, f_{k-1}(\cdot));$ 
   $P_{k|k-1} = \bar{P}_{k|k-1} + Q_{k-1};$ 
end
Step UKF update
  Data:  $\hat{x}_{k|k-1}, P_{k|k-1}, h_k(\cdot)$ 
  Result:  $\hat{x}_{k|k}, P_{k|k}$ 
   $(y_{k|k-1}, \bar{S}_k, P_k^{xy}) = UT(\hat{x}_{k|k-1}, P_{k|k-1}, h_{k-1}(\cdot));$ 
   $S_k = R_k + \bar{S}_k;$ 
   $L_k = P_k^{xy} S_k^{-1};$ 
   $e_k = y_k - \hat{y}_{k|k-1};$ 
   $\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k e_k;$ 
   $P_{k|k} = P_{k|k-1} - L_k S_k L_k^T;$ 
end

```

ALGORITHM 2: Navigation filter based on UKF.

These values are carried out for the five paths generated by waypoints (WPs) as follows: WP0-WP1, WP1-WP2, WP2-WP3, WP3-WP4, and WP4-WP0. By comparing the RMS and peak values of the path tracking error, it is shown that the trajectory error is lower if the UKF algorithm is applied.

Based on the comparison between the above obtained experimental data and the experimental data of the existing results in [62] (the thesis has been supervised by the author) in which PID was only implemented as the control law for the same physical AUV model, the horizontal planar trajectory-tracking controller of this M-AUV was improved on the stabilized course duration and trajectory error. Following these obtained trial test results (Tables 4 and 5), we can find that the UKF could provide considered

improvement over the EKF in the control implementation for AUV controllers. In fact, the EKF has been widely accepted as a standard tool in the control and machine learning communities. In this work, we used the UKF, which addresses many of the approximation issues of the EKF. The UKF consistently performs equally to or better than the EKF at comparable levels of complexity. The performance benefits of UKF-based algorithms have been shown in a number of application domains, including state estimation, dual estimation, and parameter estimation. The UKF can facilitate the accurate implementation of the optimal recursive estimation equations, although equations specifying the UKF may appear more complicated than those for the EKF algorithm.

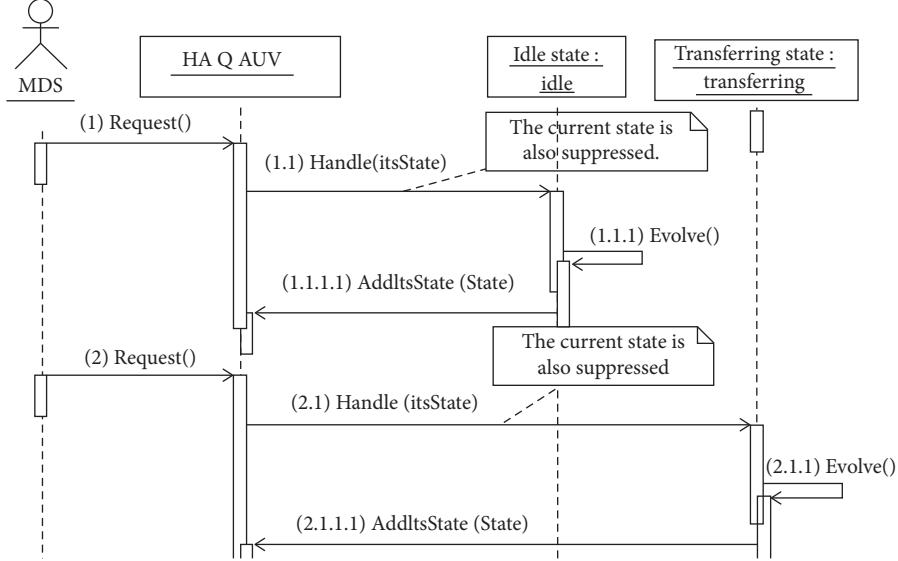


FIGURE 9: An example of instantiation and use of state pattern of HA for the M-AUV controller.

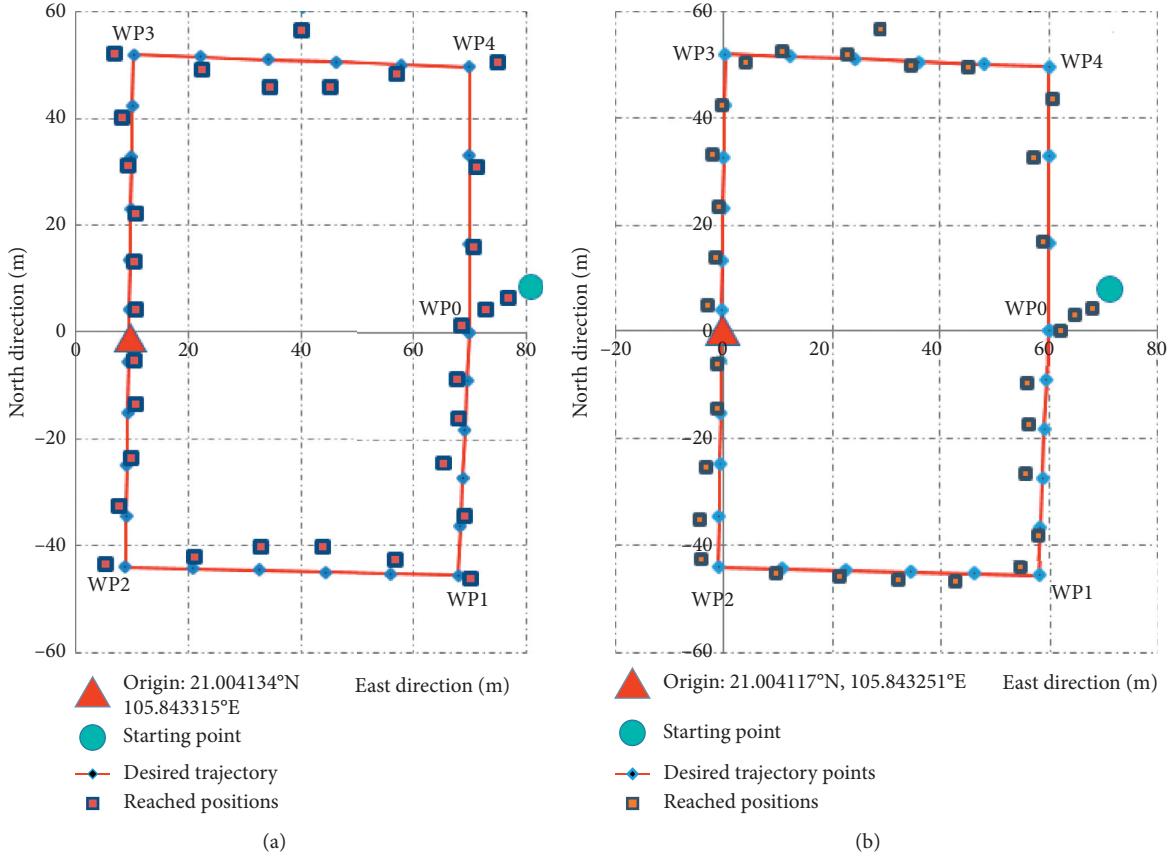


FIGURE 10: M-AUV reaches and follows the rectangle-shaped planar trajectory (a) with the EKF algorithm and (b) with the UKF algorithm.

TABLE 4: Experimental scenarios and obtained test data for the course tracking of the M-AUV.

Nº	Desired course angle (deg)	Mean transferring speed (m/s)	Duration for the stabilized course (s) (with the EKF)	Duration for the stabilized course (s) (with the UKF)
1	010	1.0	6.4	6.2
2	010	1.5	5.7	5.6
3	020	1.0	6.7	6.5
4	020	1.5	6.0	6.0
5	030	1.0	7.7	7.6
6	030	1.5	6.6	6.4

TABLE 5: Comparison of the path tracking error between the two cases using the EKF and UKF algorithms.

Paths generated by WPs	Using the EKF algorithm				Using the UKF algorithm			
	RMS position error (m)		Peak value (m)		RMS position error (m)		Peak value (m)	
	East direction	North direction	East direction	North direction	East direction	North direction	East direction	North direction
WP0-WP1	2.9296	3.1850	3.0982	3.5367	2.7036	3.0876	2.9207	3.4398
WP1-WP2	1.9599	4.1774	2.3279	5.3946	1.7465	2.1935	2.3279	2.8723
WP3-WP4	2.1686	2.7848	2.3864	3.1895	2.0754	2.6527	2.2675	2.0894
WP3-WP4	2.2597	3.8406	2.6073	5.0781	2.0532	2.0921	2.2793	2.7260
WP4-WP0	1.7650	1.5892	1.9934	1.7892	1.7270	1.5573	1.9874	1.7674

## 5. Conclusion and Future Work

The paper introduced a systems engineering-based implementation model to conveniently realize controllers for AUVs. This model is mainly based on the specializations of the MDA/MBSE approach combined with the real-time UML/SysML, EKF/UKF algorithms, and HA for analyzing, designing, and deploying closely AUV controllers. No single formalism or language of an engineering process can possibly capture all the knowledge and information needed to solve complex control systems such as the AUV controller. Hence, the AUV dynamics and control structure are adapted for the requirement analysis of control that are implemented by the specializations of MDA/MBSE features including the CIM, PIM, and PSM elements. In the CIM, the use-case model is clearly defined and is enclosed with the continuous behaviors, EKF/UKF algorithms, and HA in order to gather the control requirements for an AUV. The PIM is designed by specifying the real-time control capsules, ports, and protocols to obtain a real-time communication pattern for AUV controllers. This pattern can permit the designed PIM elements to be customizable and reusable in new application developments of various AUV types (Table 2). The PIM is then converted into the PSM through the round-trip engineering in order to entirely implement the AUV controller with compatible embedded microcontrollers. Following this proposed control model, a horizontal planar trajectory-tracking controller of a low-cost AUV was developed and tested out ATMEGA U2 and STM32-Cortex-M4 microcontrollers.

Using the above-specified MDA/MBSE approach, engineering developers will be more capable to manage the system complexity through the visual component models and their model transformations, e.g., the model transformations were performed through the round-trip engineering of the intermediate C++ codes in the case study.

However, the real-time UML/SysML versions lack the constructs for modeling internal continuous behaviors for each state on the state machine diagram, and implemented functional block diagram must then be defined in the CIM to model continuous behaviors for the AUV controller with events issued from outside.

Finally, we applied the above specialized model just in the low-cost AUV controller and intended to implement it in the new control applications for the autonomous coordinated AUV. Eventually, if we get positive feedbacks, we will investigate in the application strategy to extend it more effectively in order to create cooperative controllers for balancing the search and target response in the coordinated team of an autonomous unmanned ship and various AUVs for ocean exploration.

## Abbreviations

AUVs:	Autonomous underwater vehicles
CIM:	Computation independent model
CLF:	Control Lyapunov function
DoF:	Degrees of freedom
EKF:	Extended Kalman filter
GPS:	Global Positioning System
HA:	Hybrid automata
HDS:	Hybrid dynamic systems
HRN:	Hierarchical robust nonlinear
IB:	Integral backstepping
IDE:	Implementation development environment
IGCB:	Instantaneous global continuous behavior
IMU:	Inertial measurement unit
INCOSE:	International Council on Systems Engineering
LOS:	Line of sight
MBSE:	Model-based systems engineering
MDA:	Model-driven architecture
MDS:	Measurement-cum-display system

MES:	Marine environment system
M-AUV:	Miniature AUV
OMG:	Object Management Group
PID:	Proportional-integral-derivative
PIM:	Platform-independent model
PSM:	Platform-specific model
RPY:	Roll, pitch, and yaw
SMC:	Sliding mode control
SNAME:	Society of Naval Architects and Marine Engineers
UKF:	Unscented Kalman filter
UML:	Unified Modeling Language
WP:	Waypoint.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Centennial SIT Action for the 100<sup>th</sup> anniversary of Shibaura Institute of Technology to enter the top ten Asian Institute of Technology.

## References

- [1] R. B. Wynn, V. A. I. Huvenne, T. P. L. Bas et al., "Autonomous underwater vehicles (AUVs): their past, present and future contributions to the advancement of marine geoscience," *Marine Geology-International Journal of Marine Geology, Geochemistry and Geophysics*, vol. 352, pp. 451–468, 2014.
- [2] D. Ribas, P. Ridao, C. Melchorri, G. Palli, J. J. Fernández, and P. J. Sanz, "I-AUV mechatronics integration for the trident FP7 project," *IEEE/ASME Transactions on Mechatronics*, ISSN, vol. 20, pp. 2583–2592, 2015.
- [3] L. Brignone, E. Raugel, J. Opderbecke, V. Rigaud, R. Piasco, and S. Ragot, "First sea trials of HROV the new hybrid vehicle developed by IFREMER," in *Proceedings of OCEANS 2015 - Genova*, pp. 1–7, Genova, Italy, May 2015.
- [4] B. Allotta, R. Conti, R. Costanzi et al., "A low cost autonomous underwater vehicle for patrolling and monitoring," *Journal of Engineering for the Maritime Environment*, vol. 231, no. 3, pp. 740–749, 2016.
- [5] S. Sivčev, J. Coleman, E. Omerdić, G. Dooly, and D. Toal, "Underwater manipulators: a review," *Ocean Engineering*, vol. 163, pp. 431–450, 2018.
- [6] Y. H. Lin, S. Y. Chen, and C. H. Tsou, "Development of an image processing module for autonomous underwater vehicles through integration of visual recognition with stereoscopic image reconstruction," *Journal of Marine Science and Engineering*, MDPI, vol. 7, no. 4, p. 42, 2019.
- [7] H. R. Karimi, "A sliding mode approach to Hoo synchronization of master-slave time-delay systems with Markovian jumping parameters and nonlinear uncertainties," *Journal of the Franklin Institute*, vol. 349, no. 4, pp. 1480–1496, 2012.
- [8] T. H. Bryne, R. H. Rogne, T. I. Fossen, and T. A. Johansen, "A virtual vertical reference concept for aided inertial navigation at the sea surface," *Control Engineering Practice*, vol. 70, pp. 1–14, 2018.
- [9] K. Y. Pettersen and T. I. Fossen, "Guidance of Autonomous Underwater Vehicles," in *Encyclopedia of Robotics*, M. A. Ang, O. Khatib, and B. Sicilano, Eds., Springer, New York, USA, 2018.
- [10] Y. Sun, C. Zhang, G. Zhang, H. Xu, and X. Ran, "Three-dimensional path tracking control of autonomous underwater vehicle based on deep reinforcement learning," *Journal of Marine Science and Engineering*, MDPI, vol. 7, no. 12, p. 22, 2019.
- [11] J. Fan, Y. Li, Y. Liao et al., "Second path planning for unmanned surface vehicle considering the constraint of motion performance," *Journal of Marine Science and Engineering*, MDPI, vol. 7, no. 4, p. 19, 2019.
- [12] D. Qi, J. Feng, and J. Yang, "Longitudinal motion control of AUV based on fuzzy sliding mode method," *Journal of Control Science and Engineering*, vol. 2016, Article ID 7428361, 7 pages, 2016.
- [13] M. M. Naddaf-Sh, H. Myler, and H. Zargarzadeh, "Design and implementation of an Assistive real-time red lionfish detection system for AUV/ROVs," *Complexity*, vol. 2018, Article ID 5298294, 10 pages, 2018.
- [14] X. Wang and S. Liang, "Maneuverability analysis of a novel portable modular AUV," *Mathematical Problems in Engineering*, vol. 2019, Article ID 163193, 17 pages, 2019.
- [15] J. Rodriguez, H. Castañeda, and J. L. Gordillo, "Design of an adaptive sliding mode control for a micro-AUV subject to water currents and parametric uncertainties," *Journal of Marine Science and Engineering*, MDPI, vol. 7, no. 12, p. 23, 2019.
- [16] Z. Yan, M. Wang, and J. Xu, "Robust adaptive sliding mode control of underactuated autonomous underwater vehicles with uncertain dynamics," *Ocean Engineering*, vol. 173, pp. 802–809, 2019.
- [17] M. Karkoub, H. M. Wu, and C. L. Hwang, "Nonlinear trajectory-tracking control of an autonomous underwater vehicle," *Ocean Engineering*, vol. 145, pp. 188–198, 2017.
- [18] B. Allotta, L. Chisci, R. Costanzi et al., "An unscented Kalman filter based navigation algorithm for autonomous underwater vehicles," *Mechatronics: The Science of Intelligent Machines*, vol. 39, pp. 185–195, 2016.
- [19] B. Allotta, A. Caitib, R. Costanzi et al., "A new AUV navigation system exploiting unscented Kalman filter," *Ocean Engineering*, vol. 113, pp. 121–132, 2016.
- [20] OMG, "Documents associated with unified modeling language™ (UML® Version 2.5): OMG formal/15-03-01," 2015, <http://www.omg.org/spec/UML/2.5/>.
- [21] OMG, "SysML Specifications Version 1.5: OMG formal/17-05-01," 2017, <https://www.omg.org/spec/SysML/1.5/>.
- [22] OMG, "Model driven architecture (MDA): guide revision 2.0 of MDA Guide Version 1.0.1 (12th June 2003). OMG Document Ormsc/2014-06-01," 2014.
- [23] INCOSE, *Systems Engineering Vision 2020*, Version 2.03, INCOSE, San Diego, CA 92111-2222, USA, 2007.
- [24] INCOSE, *Systems Engineering Vision 2025*, INCOSE, San Diego, CA 92111-2222, USA, 2014.
- [25] INCOSE, "Model-based systems engineering (MBSE)," 2019, <https://www.incose.org/>.
- [26] OMG, "Unified Architecture framework, Version 1.1: formal/19-11-07," 2020.
- [27] L. T. W. Agner, I. W. Soares, P. C. Stadzisz, and J. M. Simão, "A Brazilian survey on UML and model-driven practices for embedded software development," *Systems and Software*, vol. 86, no. 4, pp. 997–1005, 2013.

- [28] M. Rashid, M. W. Anwar, and A. M. Khan, "Toward the tools selection in model based system engineering for embedded systems—a systematic literature review," *Journal of Systems and Software*, vol. 106, pp. 150–163, 2015.
- [29] L. O. Freire, L. M. Oliveira, R. T. S. Vale et al., "Development of an AUV control architecture based on systems engineering concepts," *Ocean Engineering*, vol. 151, pp. 157–169, 2018.
- [30] F. Herrera, H. Posadas, P. Peñil et al., "The complex methodology for UML/MARTE Modeling and design space exploration of embedded systems," *Journal of Systems Architecture*, vol. 60, no. 1, pp. 55–78, 2014.
- [31] N. V. Hien, N. V. He, and P. G. Diem, "A model-driven implementation to realize controllers for Autonomous Underwater Vehicles," *Applied Ocean Research*, vol. 78, pp. 307–319, 2018.
- [32] T. Soriano, N. V. Hien, K. M. Tuan, and T. V. Anh, "An object-unified approach to develop controllers for autonomous underwater vehicles," *Mechatronics: The Science of Intelligent Machines*, vol. 35, pp. 54–70, 2016.
- [33] OMG, "MDA success stories," 2019.
- [34] M. W. Anwar, M. Rashid, F. Azam, M. Kashif, and B. Butt, "A model-driven framework for design and verification of embedded systems through SystemVerilog," *Design Automation for Embedded Systems*, vol. 23, pp. 179–223, 2019.
- [35] OMG, "UML profile for MARTE: modeling and analysis of real-time embedded systems. OMG Formal Version," 2011.
- [36] B. P. Douglass, *Real-Time UML Workshop for Embedded Systems*, Elsevier, Oxford, UK, 2nd edition, 2014.
- [37] B. Selic and S. Gerard, *Modeling and Analysis of Real-Time and Embedded Systems with UML and MARTE*, Elsevier, Oxford, UK, 2014.
- [38] Arduino, "Open-source electronics prototyping platform for hardware and software," 2019.
- [39] N. V. Hien, V. T. Truong, and N. T. Bui, "An object-oriented systems engineering point of view to develop controllers of quadrotor unmanned aerial vehicles," *International Journal of Aerospace Engineering*, vol. 2020, Article ID 8862864, 17 pages, 2020.
- [40] SNAME, *Nomenclature for Treating the Motion of a Submerged Body through a Fluid, Technical and Research Bulletin No. 1–5*, SNAME (The Society of Naval Architects and Marine Engineers), New York, NY, USA, 1950.
- [41] B. Allotta, R. Costanzi, L. Pugi, and A. Ridolfi, "Identification of the main hydrodynamic parameters of Typhoon AUV from a reduced experimental dataset," *Ocean Engineering*, vol. 147, pp. 77–88, 2018.
- [42] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, John Wiley & Sons, London, UK, 2011.
- [43] H. Xu, G. C. Zhang, Y. S. Sun, S. Pang, X. R. Ran, and X. B. Wang, "Design and experiment of a plateau data-gathering AUV," *Journal of Marine Science and Engineering, MDPI*, vol. 7, no. 10, p. 20, 2019.
- [44] J. H. Li, M. G. Kim, H. Kang, M. J. Lee, and G. R. Cho, "UUV simulation modeling and its control method simulation and experimental studies," *Journal of Marine Science and Engineering, MDPI*, vol. 7, no. 4, p. 18, 2019.
- [45] B. Lantos and L. Márton, *Nonlinear Control of Vehicles and Robots*, Springer, London, UK, 2011.
- [46] L. P. Carloni, R. Passerone, A. Pinto, and V. A. Sangiovanni, *Languages and Tools for Hybrid Systems Design*, Now Publishers Inc, Boston, MA, USA, 2006.
- [47] P. A. Fishwick, *Handbook of Dynamic System Modeling*, Taylor & Francis Group, Boston, MA, USA, 2007.
- [48] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *Journal of Computer and System Sciences*, vol. 57, no. 1, pp. 94–124, 1998.
- [49] A. M. Lekkas and T. I. Fossen, "Integral LOS path Following for curved paths based on a monotone cubic hermite spline parametrization," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2287–2301, 2014.
- [50] K. Shojaei and M. Dolatshahi, "Line-of-sight target tracking control of underactuated autonomous underwater vehicles," *Ocean Engineering*, vol. 133, pp. 244–252, 2017.
- [51] Z. Zheng and Y. Zou, "Adaptive integral LOS path following for an unmanned airship with uncertainties based on robust RBFNN backstepping," *ISA Transactions*, vol. 65, pp. 210–219, 2016.
- [52] W. Li, W. Wu, J. Wang, and M. Wu, "A novel backtracking navigation scheme for Autonomous Underwater Vehicles," *Measurement*, vol. 47, pp. 496–504, 2014.
- [53] E. A. Wan and R. V. D. Merwe, "The Unscented Kalman Filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed., pp. 221–280, Wiley, New York, NY, USA, 2001.
- [54] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation-Theory Algorithms and Software*, John Wiley & Sons, New York, NY, USA, 2001.
- [55] N. V. Hien, V. T. Truong, N. V. He et al., *Research, 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 Report of Research Project, Funded by the State, Code: KC03.TN05/11-15*, Hanoi University of Science and Technology, Hanoi, Vietnam, 2013.
- [56] IBM, "IBM rational online documentation and training kit," 2019, <https://estore.onthehub.com>.
- [57] Papyrus, "Eclipse papyrus for real-time ("Papyrus-RT")," 2019, <https://www.polarsys.org/papyrus-ic/products>.
- [58] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Oxford, UK, 1995.
- [59] B. P. Douglass, *Design Patterns for Embedded Systems in C—an Embedded Software Engineering Toolkit*, Elsevier, Oxford, UK, 1st edition, 2011.
- [60] InvenSense, "Sensor system on chip," 2019, <http://www.invensense.com/>.
- [61] u-blox, "Gobal leader in wireless communications and positioning semiconductors and modules for the industrial, automotive and consumer markets," 2019, <https://www.u-blox.com>.
- [62] N. Dong, *Analysis of hydrodynamics and control system design of Object-Oriented technology for autonomous underwater vehicles*, Ph.D. thesis, Hanoi University of Science and Technology, Hanoi, Vietnam, 2015.