

Mô hình điều khiển phương tiện thủy tự hành trên mặt nước bằng Automate lai và SysML

Cao Đức Sáng¹, Lê Trần Thắng¹, Ngô Văn Hiền²,
Ngô Văn Hệ^{2*}, Nguyễn Trọng Khuyên¹

¹Viện Tự động hóa Kỹ thuật quân sự, Viện Khoa học và Công nghệ quân sự, 89B Lý Nam Đế, Hoàn Kiếm, Hà Nội, Việt Nam;

²Đại học Bách khoa Hà Nội, 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội, Việt Nam.

*Email: he.ngovan@hust.edu.vn

Nhận bài: 12/01/2024; Hoàn thiện: 04/3/2024; Chấp nhận đăng: 14/3/2024; Xuất bản: 25/3/2024.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.CAPITI.2024.175-181>

TÓM TẮT

Các phương tiện thủy tự hành (AUVs/ASVs: Autonomous Underwater Vehicles/Autonomous Surface Vehicles) đang được khai thác một cách hiệu quả trong việc nghiên cứu, khảo sát và khai thác tài nguyên biển. Trong đó, việc thiết kế hệ thống điều khiển cho phương tiện thủy tự hành là rất quan trọng bởi vì nó được liên kết chặt chẽ với mô hình thủy động lực học của phương tiện hoạt động môi trường nước rất phức tạp. Bài báo này giới thiệu một mô hình hướng đối tượng lai trong thực thi hệ thống điều khiển cho phương tiện không người lái tự hành trên mặt nước (ASVs); mô hình này được phân tích, thiết kế và thực thi một cách hệ thống thông qua việc cụ thể hóa automate lai và ngôn ngữ mô hình hóa hệ thống (SysML: Systems Modeling Language). Dựa trên mô hình lai được đề xuất này, một bộ điều khiển bám theo hướng đi trên mặt ngang của ASV mô hình đã được triển khai và thử nghiệm với độ tin cậy tốt.

Từ khóa: Phương tiện thủy tự hành; Điều khiển phương tiện thủy tự hành; Automate lai; Ngôn ngữ mô hình hóa hệ thống; Công nghệ hướng đối tượng trong thời gian thực; Chuyển đổi mô hình.

1. GIỚI THIỆU

Hiện nay, phương tiện không người lái tự hành dưới nước (AUV: Autonomous Underwater Vehicles) nói chung và phương tiện thủy tự hành trên mặt nước (ASV: Autonomous Surface Vehicles) nói riêng, đang được sử dụng phổ biến bởi các nhà đầu tư quốc phòng và dân sự cho các tác vụ phức tạp ở môi trường nước. AUV/ASV có các đặc tính cơ bản về an toàn và chi phí hiệu quả hơn khi được so sánh với phương tiện thủy có người lái [1]. Trên thực tế đã có nhiều ứng dụng điều khiển ASV với các giải pháp tối ưu [2-4] cho việc kiểm soát mô hình động lực học của các AUV/ASVs này. Tuy nhiên, các mô hình điều khiển kèm theo các mô hình thủy động lực học tương ứng ở trên được phát triển theo hướng thủ tục và cơ sở dữ liệu tính toán trong kiến trúc tập trung; do đó chúng sẽ rất khó khăn trong việc được tùy biến và tái sử dụng các thành phần điều khiển đã được thiết kế để ứng dụng cho ASV khác nhau, nhằm giảm giá thành chi phí của thiết bị được chế tạo [5].

Với những phân tích trên đây, bài báo đã phát triển một mô hình đối tượng lai để phân tích, thiết kế, thực hiện và triển khai một cách thuận tiện hệ thống điều khiển cho ASV với ứng xử động lực học được mô hình hóa bởi automate lai (HA: Hybrid Automata) [6]. Trong cách tiếp cận này, các tính năng của công nghệ hệ thống hướng mô hình (MBSE: Model-Based Systems Engineering) [7], được cụ thể hóa nhằm tương thích với mô hình động lực học của ASV, bao gồm: Mô hình phân tích thực hiện bằng cách xác định mô hình trường hợp sử dụng với ngôn ngữ mô hình hóa hệ thống hợp nhất (UML/SysML: Unified Modeling Language/Systems Modeling Language) kết hợp với HA; Mô hình thiết kế được đưa ra thông qua các tác tạo trong thời gian thực với Real-Time UML/SysML [8, 9], và mô hình thực thi của nó được triển khai bởi các cơ chế hướng đối tượng dựa trên nền tảng mã nguồn mở, ví dụ: *OpenModelica* [10] và *Arduino* [11], nhằm đưa ra một qui trình hướng đối tượng cho việc phân tích, thiết kế và thi hành một cách có hệ thống và nhanh chóng

hệ thống điều khiển cho ASV. Cuối cùng, cách tiếp cận này đã được áp dụng để hoàn thiện phân tích, thiết kế và thực thi cho một ASV mô hình; nó cho phép phương tiện bám theo hướng đi trên mặt ngang mong muốn đặt trước.

2. THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN ASV

2.1. Mô hình động lực học trong điều khiển ASV

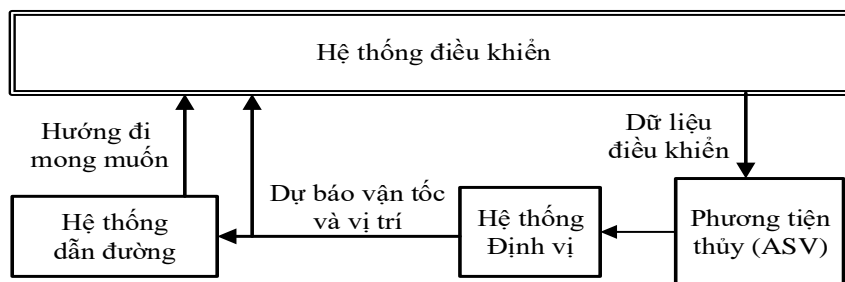
Một phương tiện thủy chuyển động theo 6 bậc tự do (DoF: *Degree of Freedom*) [12] được mô tả bởi hệ phương trình (1).

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

Trong đó: $v = (u, v, w, p, q, r)^T$ là vận tốc của phương tiện dưới nước trong hệ tọa độ liên kết với ASV; $\eta = (x, y, z, \varphi, \theta, \psi)^T$ biểu diễn vị trí và góc Euler; $M = M_{RB} + M_A$ là ma trận quán tính 6×6 đặc trưng cho khối lượng và mô men quán tính, nó bao gồm thành phần đại diện cho thực thể rắn di chuyển M_{RB} và bổ sung M_A ; $C(v) = C_{RB}(v) + C_A(v)$ là ma trận Coriolis và lực ly tâm 6×6 kèm theo khối lượng bổ sung; $D(v) = D + D_n(v)$ đặc trưng cho giảm chấn thủy động lực tuyến tính và phi tuyến được biểu diễn bởi ma trận dạng 6×6 ; $g(\eta)$ là véc tơ 6×1 đại diện cho tác động trọng lực; τ là véc tơ 6×1 đặc trưng cho lực phân phối điều khiển tới các cơ cấu chấp hành, như: cánh lái, chân vịt và ống đạo lưu (nếu có); ω là véc tơ 6×1 tham số đại diện cho của các ảnh hưởng nhiễu môi trường tác động, như là dòng hải lưu, gió và sóng.

2.2. Kiến trúc thực thi điều khiển cho ASV

Kiến trúc điều khiển để cho phương tiện dưới nước tự hành thường có ba hệ thống chính: Hệ thống dẫn đường được sử dụng nhằm đưa ra quỹ đạo cho phương tiện chuyển động bám theo; Hệ thống định vị có nhiệm vụ xác định các trạng thái hiện hành của phương tiện; Hệ thống điều khiển để thực thi tính toán và áp dụng theo mô hình động lực học (1) với các thông số theo mặt ngang $v_I = (u, v, r)^T$ và $\eta_I = (x, y, \psi)^T$ tương thích với mỗi trạng thái cụ thể của phương tiện. Hình 1 trình bày sơ đồ khối thể hiện tương tác giữa các hệ thống này.

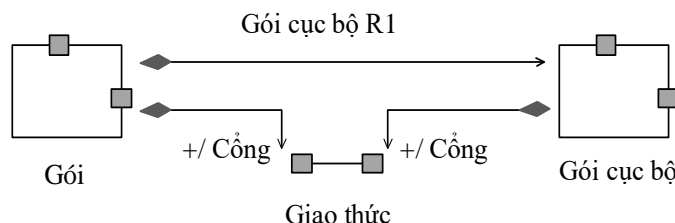


Hình 1. Sơ đồ khối tổng quát dẫn đường, điều khiển và định vị cho ASV.

Trong bài báo này, việc điều khiển bám theo hướng đi mong muốn trên mặt nước cho ASV được lấy làm mục tiêu chính. Từ phương trình động lực học tổng quát của phương tiện dưới nước (1), ta có thể triển khai được mô hình động lực học cho ASV chuyển động trên mặt phẳng nằm ngang với các thông số $v_I = (u, v, r)^T$ và $\eta_I = (x, y, \psi)^T$. Xuất phát từ mô hình động lực học điều khiển (1) và các đặc điểm của hệ thống động lực lai (HDS: *Hybrid Dynamic System*) công nghiệp đã được mô tả trong [6] và sơ đồ khối trên đây, các mô hình phân tích, thiết kế và thực thi hệ thống điều khiển ASV được xem như là một HDS công nghiệp và có ứng xử điều khiển được mô hình hóa bởi automate lai (HA).

Để lựa chọn ngôn ngữ mô hình hóa trực quan theo các nguyên tắc cơ bản trong công nghệ hướng đối tượng, như: trừu tượng hóa, mô đun hóa, đóng gói và phân cấp, nhằm thực hiện toàn bộ các pha phân tích tới thiết kế, mô phỏng và thực thi, bài báo đã chọn Real-Time UML/SysML;

phiên bản này bao gồm tất cả các ký hiệu mô hình hóa của UML/SysML; ngoài ra, nó còn đưa ra các ký hiệu mô hình hóa: gói (*Capsule*), cổng (*Port*) và giao thức (*Protocol*) nhằm mô hình hóa chi tiết tương tác giữa các đối tượng của các hệ thống điều khiển trong công nghiệp nói chung và ASV nói riêng. Hình 2 minh họa về gói, gói cục bộ (*Nested Capsule*), cổng và giao thức trong Real-Time UML/SysML.



Hình 2. Minh họa về gói, cổng và giao thức trong mô hình hóa trực quan với Real-Time UML/SysML.

2.3. Thiết kế hệ thống

Khác với phương pháp truyền thống với hệ thống điều khiển được xây dựng thiết kế theo hướng thủ tục. Phương pháp đề xuất đưa ra một qui trình hướng đối tượng cho việc phân tích, thiết kế, thi hành một cách có hệ thống và nhanh chóng bộ điều khiển cho ASV. Nó bắt đầu từ mô hình phân tích điều khiển hệ thống cho đến bước đưa ra mã thực thi.

2.3.1. Mô hình phân tích điều khiển ASV

Trong giai đoạn phân tích với UML/SysML cho hệ thống điều khiển ASV, mô hình trường hợp sử dụng đưa ra một tập hợp các tác nhân (*Actors*) và các trường hợp sử dụng (*Use cases*) với các mối quan hệ dạng truyền đạt, mở rộng, bao hàm và kế thừa giữa chúng. Tác nhân là một sự mô tả một tập hợp nhất quán các vai trò mà một người sử dụng hoặc hệ thống tham gia khi nó tương tác với ASV. Trường hợp sử dụng thể hiện một tập hợp các hàm chức năng hoặc các ứng xử động được cung cấp bởi ASV cho các tác nhân với các mối quan hệ: bao hàm (*Include*) và mở rộng (*Extend*). Các bước nhận dạng mô hình trường hợp sử dụng của ASV được thực hiện như sau:

- Xác định các trường hợp sử dụng và các tác nhân liên kết của ASV. Sau đó, chúng ta đặc tả các trường hợp sử dụng với các quan hệ bao hàm và mở rộng.

- Mô tả các ứng xử của hệ thống bởi các sơ đồ diễn tiến cho mỗi trường hợp sử dụng đã được xác định. Những sơ đồ diễn tiến này được tìm thấy từ sự kết nối giữa ASV và các hệ thống tương tác khác, như là hệ thống dẫn đường/định vị. Trong bước này chúng ta cần đưa ra điều kiện ràng buộc cho ASV công nghiệp nhằm đảm bảo sự hoạt động chính xác và an toàn của hệ thống, như là vận tốc tối đa khi di chuyển, thời gian hoạt động tối đa trên một chu trình vận hành, góc quay và vận tốc quay cực đại của bánh lái hoặc cánh lái,...

- Đưa ra các máy trạng thái cục bộ của hệ thống phát triển tương ứng với sơ đồ diễn tiến đã xác định. Các máy trạng thái này được suy ra từ ứng xử của ASV được phát triển trong trường hợp sử dụng đã xác định với các sự kiện hoặc thông điệp từ các tác nhân.

- Phân loại các trường hợp sử dụng đã xác định thành hai nhóm: các trường hợp sử dụng được hướng theo thuật toán điều khiển cụ thể; các trường hợp sử dụng được dựa theo thông tin truyền thống chuyên dụng.

Tiếp theo, chúng ta xét các trường hợp sử dụng hướng theo thuật toán điều khiển cụ thể của ASV; Sơ đồ máy trạng thái cục bộ của các trường hợp này được sử dụng để xác định các đặc trưng trong HA của hệ thống được mô tả bởi phương trình (2).

$$H_{ASV} = (Q, \Sigma, A, X, F, Inv, q_o, x_o) \quad (2)$$

Ở đây, Q là tổ hợp các vị trí mô tả các chế độ hoạt động của hệ thống; q_o là vị trí (trạng thái)

ban đầu; Σ là tập hợp hữu hạn các sự kiện tác động; A là tập hợp các dịch chuyển giữa các vị trí, $\sigma \in \Sigma$ là tổ hợp các sự kiện cho phép dịch chuyển vị trí; X là không gian trạng thái liên tục của Automate, $X \in R^n$; x_0 là giá trị ban đầu của trạng thái liên tục; F là hàm điều khiển liên tục tổng thể được triển khai từ trình động lực học tổng quát của phương tiện dưới nước (1) cho ASV chuyển động trên mặt phẳng nằm ngang với các tập tham số (u, v, r) và (x, y, ψ) ; tiến trình của hàm điều khiển liên tục tổng thể F được tạo ra ngay tức thời khi vị trí tương ứng $q' \in Q$ của nó được kích hoạt; Inv là đại lượng bất biến, dùng để theo dõi trạng thái liên tục phải được duy trì, cụ thể vị trí là q' thì trạng thái liên tục cần phải được xác thực theo $x \in Inv(q')$. Các qui tắc xây dựng HA của ASV được mô tả như sau:

- Mỗi trạng thái tương ứng với một vị trí của HA có kết hợp theo phép *hoán vị* với tập các trạng thái chuyển động của ASV;

- Tập hợp các điều kiện ràng buộc công nghiệp hoặc điều kiện sử dụng của các thành phần trong sơ đồ khối chức năng mở rộng tương ứng với mỗi đại lượng bất biến, được kết hợp với vị trí hiện thời trong HA. Bên cạnh đó, các điều kiện ràng buộc của các thành phần này có thể được tạo ra nhằm xác định không gian con của một dòng liên tục hiện thời mà trong đó trạng thái liên tục phải được gọi nhớ. Trạng thái không gian con này mô tả đại lượng bất biến kết hợp với một vị trí kèm theo dòng liên tục cụ thể $f_i \in F$;

- Mỗi dịch chuyển trạng trong máy trạng thái tương ứng với một dịch chuyển trong HA; *Guard* là tập hợp con đại lượng bất biến của vị trí đến; *Jump* mô tả giá trị phép biến đổi trạng thái liên tục trong suốt quá trình thay đổi vị trí, nó đưa ra một cách chung nhất dưới dạng của một hàm giá trị trạng thái, trước khi truyền dữ liệu kết quả của chúng được coi như giá trị ban đầu của trạng thái liên tục trong vị trí mới.

Trên đây, bài báo đã đưa ra cách tiếp cận hướng đối tượng nhằm xây dựng mô hình phân tích của ASV dựa trên phương pháp trường hợp sử dụng (*Use-Cases Method*) trong UML/SysML và cụ thể hóa các đặc trưng của HA. Đặc tả chi tiết và minh họa cách tiếp cận này có thể được xem trong [14].

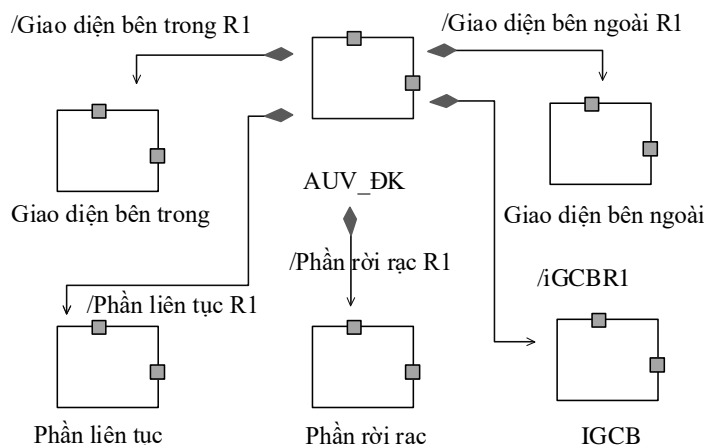
2.3.2. Mô hình thiết kế điều khiển ASV

Sự chuyển đổi từ mô hình phân tích tới môi trường thực thi phải được bổ sung thêm các mô hình cấu trúc và ứng xử chi tiết để thực thi hệ thống điều khiển ASV. Để xây dựng cấu trúc tổng quát mô hình thiết kế hệ thống điều khiển ASV bằng Real-Time UML/SysML dựa trên HA, nghiên cứu đã đưa ra năm khối điều khiển chính, bao gồm: khối thể hiện các thành phần rời rạc (*Discrete Part Block*) bao gồm tập hợp vị trí Q và tập hợp dịch chuyển trạng thái A của HA; Khối thể hiện các thành phần liên tục (*Continuous Part Block*) chứa các lớp thực thể để lưu trữ và xử lý các thông tin hoạt động biến đổi của các phần tử liên tục thuộc X trong HA; Ứng xử liên tục toàn cục tức thời (IGCB: *Instantaneous Global Continuous Behavior Block*) bao gồm các ứng xử liên tục toàn cục của ASV được thực thi như F trong HA; Giao diện bên ngoài (*External Interface Block*) là thành phần trung gian giữa hệ thống điều khiển của ASV với các hệ thống tương tác bên ngoài, như: hệ thống dẫn đường, hệ thống định vị và nhiều môi trường; Giao diện bên trong (*Internal Interface Block*) đưa ra sự kiện bên trong của ASV dựa theo đại lượng bất biến Inv . Mâu kết nối giữa các khối điều khiển chính mô hình hóa bởi các khối, giao thức và khối thông qua trong sơ đồ lớp trong Real-Time UML/SysML được trình bày như trên hình 3.

- Khối phần rời rạc bao gồm tập hợp vị trí Q và tập hợp dịch chuyển trạng thái A của HA đối với ASV, được kết hợp với máy trạng thái cục bộ; Mỗi vị trí tạo nên sự liên kết với từng IGCB cụ thể. Khối này cũng chứa một máy trạng thái để thực hiện tiến trình riêng của nó với các khối khác, như: Khối giao diện bên trong và IGCB và để xử lý các sự kiện nội bộ mặc định.

- Khối phần liên tục chứa các lớp thực thể để lưu trữ và xử lý các thông tin hoạt động biến đổi của các phần tử liên tục trong ASV. Trong khối này, có thể xây dựng một lớp thực thể trừu tượng để có thể thực thi tính kế thừa nhằm tránh nhân đôi thông tin và cũng có thể tìm thấy các thuộc tính chung giữa các lớp thực thể để tái sử dụng chúng.

- Khối IGCB bao gồm các ứng xử liên tục toàn cục của ASV được phát triển tại thời điểm cho trước như $f_i \in F$ trong HA. Khối IGCB có một máy trạng thái để tạo nên tiến trình riêng với các khối phân rời rạc, khối phân liên tục và khối giao diện bên ngoài. Trong tiến trình này, khối IGCB thay đổi các tín hiệu theo chu kỳ.



Hình 3. Mẫu liên kết giữa các khối điều khiển chính bởi Real-Time UML/SysML cho ASV.

- Khối giao diện bên trong đưa ra sự kiện bên trong của ASV dựa theo đại lượng bất biến *Inv*; từ đó, khối phân rời rạc có thể xử lý các sự kiện đã được đưa ra. Cộng tác này là thành phần trung gian giữa cộng tác phân liên tục và cộng tác phân rời rạc. Nó kiểm tra các đại lượng bất biến *Inv* kết hợp với dịch chuyển *A* của HA và đưa ra sự kiện bên trong cần thiết cho phép thực hiện tiến trình thời gian thực trong nội bộ HA cho ASV.

- Khối giao diện bên ngoài là thành phần trung gian giữa hệ thống điều khiển của ASV với các hệ thống tương tác bên ngoài, như: hệ thống dẫn đường, hệ thống định vị và nhiều môi trường. Nó nhận/gửi các sự kiện và tín hiệu chu kỳ giữa từ/đến các hệ thống tương tác. Ngoài ra, nó có thể được sử dụng trong hiển thị kết quả điều khiển, thông số cài đặt cấu hình hệ thống nhanh chóng và tối ưu.

2.3.3. Mô hình thực thi điều khiển ASV

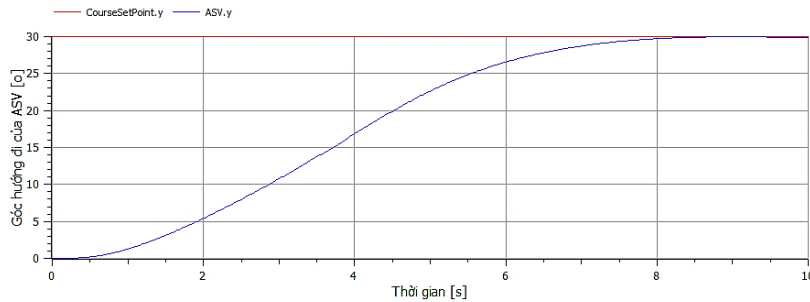
Trong nghiên cứu này, chuyển đổi mô hình được tiến hành như sau: trước hết tạo ra các mô hình phân tích và thiết kế hệ thống bằng công cụ mô hình hóa với Real-Time UML/SysML (ví dụ: *IBM Engineering Systems Design Rhapsody* [13]) theo cách tiếp cận đã được mô tả trong mục 3.1 và 3.2. Sau đó, tiến hành chuyển đổi để tạo ra mô hình phát triển mới trong khung nhìn của ngôn ngữ mô phỏng hướng đối tượng (ví dụ: *Modelica* trong môi trường phần mềm *OpenModelica* [10]). *OpenModelica-Modelica* là nền tảng mã nguồn mở và chuyên biệt hóa trong mô phỏng các hệ thống động lực lai công nghiệp với sự hỗ trợ của các thư viện hướng đối tượng liên lĩnh vực, như: cơ-điện tử, kỹ thuật thủy khí và tự động hóa. Mã nguồn của chương trình mô phỏng với *OpenModelica* có thể tùy biến và tái sử dụng trong pha triển khai hệ thống điều khiển với các bộ vi điều khiển có hỗ trợ cho nền tảng hướng đối tượng trên thực tế.

3. THỬ NGHIỆM VÀ KẾT QUẢ

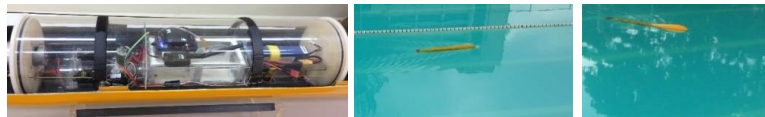
Để minh họa tính khả thi của phương pháp thiết kế hệ thống điều khiển của ASV đề xuất trên đây, bài báo đã áp dụng cho phương tiện không người lái tự hành bám theo hướng đi mong muốn trên mặt nước là ASV mô hình, với các thông số vật lý cơ bản như sau: (Chiều dài × Chiều cao × Chiều rộng) = (1,50 × 0,20 × 0,20) m; Vận tốc tối đa theo mặt phẳng ngang $V = 1,5$ m/s; Thời gian hoạt động tối đa $T = 20$ phút/1 chu trình vận hành.

Mô phỏng giả lập các tình huống bám theo hướng đi mong muốn khác nhau của ASV đã được

thực hiện thông qua công cụ mô phỏng *OpenModelica* [10]. Hình 4 minh họa một trong các kết quả mô phỏng khả năng ổn định hướng của ASV với vận tốc trung bình di chuyển ngang là 1,0 m/s và hướng đi mong muốn đặt trước là 30°. Hình 5 giới thiệu một vài hình ảnh tích hợp và thử nghiệm mô hình triển khai hệ thống điều khiển ASV bám theo hướng đi mong muốn. Trong ứng dụng này, bài báo đã sử dụng nền tảng *Arduino* và bộ vi điều khiển *MCU-STM32-Cortex M4* [11] để nhanh chóng triển khai các mô hình thực thi của hệ thống điều khiển ASV. Dự án trên *Arduino* có thể đứng riêng lẻ hoặc có thể giao tiếp với phần mềm chạy trên máy tính. Môi trường thực thi trên nền tảng *Arduino* cho phép dễ dàng tạo ra các thư viện mã chương trình điều khiển bổ sung.



Hình 4. Quá độ ổn định hướng của ASV mô hình trong trường hợp: vận tốc trung bình di chuyển trên mặt ngang $V = 1,0$ m/s và hướng đi mong muốn đặt trước 30°.



Hình 5. Minh họa tích hợp hệ thống và thử nghiệm bộ điều khiển cho phép ASV bám theo hướng đi mong muốn đặt trước.

Bảng 1. Khoảng thời gian ổn định các hướng đi mong muốn đặt trước cho ASV mô hình.

STT	Vận tốc trung bình di chuyển trên mặt ngang (m/s)	Hướng đi mong muốn đặt trước (°)	Khoảng thời gian ổn định hướng đi thực (s)
1	0,5	10	5,2
2	1,0	10	7,2
3	0,5	20	7,1
4	1,0	20	7,6
5	0,5	30	9,5
6	1,0	30	8,8

Tùy theo vào vận tốc di chuyển trung bình của ASV, thời gian ổn định hướng của ASV mô hình từ 5,2 s đến 9,5 s và thời gian trễ từ 0,45 s đến 1,60 s, theo các hướng đi mong muốn là 10°, 20° và 30° tương ứng với vận tốc di chuyển trung bình là 0,5 m/s và 1,0 m/s (bảng 1). Kết quả thử nghiệm về tính ổn định hướng đi có sai lệch từ 0,2 s đến 0,5 s so với kết quả mô phỏng trên máy tính bởi *OpenModelica*, cho thấy kết quả thử nghiệm về tính ổn định hướng đi có độ tin cậy có thể chấp nhận được theo [4].

4. KẾT LUẬN

Bài báo đã trình bày mô hình phân tích, thiết kế và thực thi phần điều khiển cho phương tiện thủy tự hành trên mặt nước (ASV). Cách tiếp cận này dựa trên *automate* lại và ngôn ngữ mô hình hóa hệ thống hợp nhất trong thời gian thực (Real-Time UML/SysML), nhằm triển khai một cách có hệ thống bộ điều khiển của ASV. Phần thử nghiệm ứng dụng phương pháp đề xuất cho ASV mô hình tự bám theo hướng đi mong muốn được đặt trước và được triển khai trên bộ vi điều khiển *MCU-STM32-Cortex M4*; Tính ổn định hướng đi của phương tiện được đánh giá có độ tin cậy có thể chấp nhận

được. Trong nghiên cứu tiếp theo, cách tiếp cận này sẽ được mở rộng nhằm mô hình hóa, thiết kế và triển khai hệ thống điều khiển cho đa phương tiện thủy tự hành hoạt động theo đội hình.

Lời cảm ơn: Nghiên cứu này được hỗ trợ bởi Đề tài Khoa học, cấp Bộ Giáo dục và Đào tạo, mã số B2023-BKA-13, tại Trường Cơ khí, Đại học Bách Khoa Hà Nội.

TÀI LIỆU THAM KHẢO

- [1]. AUVAC. "Autonomous Undersea Vehicles Applications Center", (2023). Available: <https://auvac.org/>
- [2]. Z. Liu, Q. Liu, W. Xu, L. Wang, Z. Zhou, "Robot learning towards smart robotic manufacturing: A review", ID 102360, Robotics and Computer-Integrated Manufacturing, Elsevier, ISSN 0736-5845, 77, 21 pages, (2022). <https://doi.org/10.1016/j.rcim.2022.102360>
- [3]. S. Kang, J. Yu, J. Zhang, Q. Jin, "Development of Multibody Marine Robots: A Review", IEEE Access, ISSN 2169-3536, 8, 21178-21195, (2020). <https://doi.org/10.1109/ACCESS.2020.2969517>
- [4]. T.I. Fossen, "Handbook of Marine Craft Hydrodynamics and Motion Control", 2nd Edition, John Wiley & Sons, United Kingdom, (2021).
- [5]. T. Soriano, V. Gies, H.A. Pham, N.V. Hien, "Mechatronics Iterative Design for Robots Multi-agent Integration", in: F. Chaari, M. Haddar, Y.W. Kwon, F. Gherardini, V. Ivanov (Eds.) Mechatronics 4.0, ISBN 978-3-030-46728-9, Springer, Switzerland, (2020). https://doi.org/10.1007/978-3-030-46729-6_6
- [6]. T. Soriano, N.V. Hien, K.M. Tuan, T.V. Anh, "An object-unified approach to develop controllers for autonomous underwater vehicles", Mechatronics: The Science of Intelligent Machines, Elsevier, ISSN 0957-4158, 35, 54-70, (2016). <https://doi.org/10.1016/j.mechatronics.2015.12.011>
- [7]. INCOSE, "Systems Engineering Vision 2025", INCOSE, San Diego, CA 92111-2222, USA, (2014).
- [8]. OMG, "SysML Specifications", Version 1.7 beta: OMG formal/ptc/22-08-02, OMG, (2022) <https://www.omg.org/spec/SysML/>
- [9]. OMG, "UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems", Version 1.2. OMG ptc/18-07-05, OMG, (2019). <http://www.omg.org/spec/MARTE/>.
- [10]. OpenModelica. "OpenModelica software", version 1.21.0. (2023). <https://www.openmodelica.org/>
- [11]. Arduino. "Open-source electronics prototyping platform for hardware and software". Available: <http://www.arduino.cc/>, (2023).
- [12]. SNAME, "Nomenclature for Treating the Motion of a Submerged Body through a Fluid", SNAME, New York 18, N. Y., USA, (1950).
- [13]. IBM. "IBM Rational's Methodology, Software, Online Documentation and Training Kits." Available: <https://www.ibm.com/academic/home>, (2023).
- [14]. N.V. Hien, N.V. He, V.T. Truong, P.G. Diem, "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", Hanoi University of Science and Technology, (2013).

ABSTRACT

A hybrid automata and SysML-based model for controllers of autonomous surface vehicles

Autonomous Underwater Vehicles/Autonomous Surface Vehicles (AUVs/ASVs) are increasingly being used in research, monitor and exploitation of marine resources. In particular, the design of control system for AUVs/ASVs is very important because it is closely connected to the hydrodynamic model of operating vehicles in the complex water environment. This paper presents a hybrid object-oriented control model, which is based on the specializations of hybrid automata and Systems Modeling Language (SysML), in order to systematically analyze, design and implement controllers of autonomous unmanned surface vehicles (ASVs). Based on this proposed hybrid model, a horizontal planar course-tracking controller of an ASV model was implemented and tested with good reliability.

Keywords: Autonomous underwater vehicles/autonomous surface vehicles (AUVs/ASVs); AUV/ASV control; Hybrid automata; Systems modeling language (SysML); Object-oriented technology; Model transformation.