



HAL
open science

Analysis of coordination modes for multi-UUV based on Model Driven Architecture

Thierry Soriano, Hoang Anh Pham, van Hien Ngo

► **To cite this version:**

Thierry Soriano, Hoang Anh Pham, van Hien Ngo. Analysis of coordination modes for multi-UUV based on Model Driven Architecture. 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, Sep 2018, Tsu, France. pp.189-194, <10.1109/MECATRONICS.2018.8495893>. <hal-03253821>

HAL Id: hal-03253821

<https://hal.science/hal-03253821v1>

Submitted on 2 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Analysis of coordination modes for multi-UUV based on Model Driven Architecture

Thierry SORIANO

thierry.soriano@univ-tln.fr

Université de Toulon, SEATECH
Toulon, France

Hoang Anh PHAM

hoang-anh.pham@univ-tln.fr

Université de Toulon, COSMER
Toulon, France

Van Hien NGO

hien.ngovan@hust.edu.vn

Hanoi University of Science and Technology
Hanoi, Vietnam

Abstract—The aim of the present work is to analyze coordination modes for multiple unmanned underwater vehicles (UUV) which are an open-source and low-cost platform based on the Model Driven Architecture approach. The medium-term goal should be to transform an existing remote control system into an autonomous control system capable of multiple missions. Specially, we describe the general re-engineering process in the use-case of a coordination of two UUVs. The abstract model is studied and verified based on Matlab/Simulink. Furthermore, we propose a framework, which is able to support the development towards the real platform thanks to Robot Operating System (ROS) middleware and Gazebo simulator.

Index Terms—MDA approach, cooperative multi-robot systems, complex environment, unmanned underwater vehicles, ROS/Gazebo simulation

I. INTRODUCTION

In recent years, there has been a rapid increase in the distributed cooperative control for multiple robots which serve many military and civilian applications. Moreover, the robots platforms used for research may be commercial products or open-source. According to Oh et al [1], there are some approaches of a coordination control as following: position-based control, displacement-based control, distance-based control.

In many applications, a group of UUVs are intended to follow a predefined trajectory while maintaining a desired spatial pattern. Due to uncertainty in the UUVs dynamics for wave disturbances and communication constraints in the acoustic medium, it is difficult to control UUVs [2], [3]. To control a group of UUVs, it is possible to suppose a model of vehicles as single or double-integrator dynamics which are applied by a consensus algorithms [4], [5], [6]. Several recent surveys of cooperation control of multi UUVs are found in [1], [7] and [8]. However, starting from the number of existing solutions, it is necessary to have a methodology to rapidly, modularize the components of vehicles so that it can be reused in changing algorithms, changing constraints or changing specific vehicle platform.

Up to date, the Object Management Group (OMG) has standardized the Model-Driven Architecture (MDA) [9] which proposes a methodological framework and architecture for systems development that focuses first on the functionality and application behavior. The MDA process can be divided into two phases: the first phase builds a platform-independent model (PIM), which is a high-level technology independent

model, the previous model is then transformed into one or more platform specific models (PSMs); these models are lower level and describe the system in accordance with a given deployment technology. Finally, the source code is generated from each PSM. Beside, the Object-Oriented Methodology (OOM) is a common system development approach encouraging and facilitating re-use of software components. It has four mainly principles which are abstraction, encapsulation, modularity and hierarchy. The fundamental idea behind OOM is to decompose software, and it breaks the complex software into objects, and combines data and the functions that operate on the data into a single unit.

In this paper, we applied the MDA with OOM approach to BlueROV-1 [10] which is one of the unnamed underwater vehicles and is an open-source platform. From this platform, we firstly studied the components of PSM level to back it up in the PIM level, then at PIM we refine the study of the combined modes of operation such as teleoperation, maintenance and autonomous operation. The autonomous operation consists of two modes: formation control, collision avoidance. With this approach, we can quickly develop test-driven algorithms that does not depend on a particular platform, such as collision avoidance algorithms, high-level control algorithms. Then, at the PSM* level, we developed a tool to rapidly test algorithms for a specific UUV platform such as BlueROV-1. Advantages here in this case, Matlab/Simulink development algorithms can directly communicate with ROS/Gazebo simulator objects, saving development time and testing time. In addition, with the use of the Matlab/Simulink with ROS/Gazebo, it is possible to quickly apply these algorithms to specific robotics platforms. This paper contributes to the introduction of a complete approach from research to development of algorithms to a real-world system that is fast, clear, able to inherit and reuse components what has been developed previously.

The paper has been divided into five sections. Section 2 begins by describing the model PSM of BlueROV-1 with its components. Section 3 discusses the overview of UUV dynamics and the model PIM with several modes. Section 4 proposes the simulation results with the framework of the transformation model from PIM to PSM. Finally, section 5 provides conclusions and future works.

II. MODEL PSM OF BLUEROV

A. Existing system - BlueROV-1

The BlueROV-1 is the open source and low-cost platform used. The figure 1 shows out the components of BlueROV-1 with the architecture control. The BlueROV frame and thrusters were included in a package from Blue Robotics. In addition to the BlueROV frame, a Raspberry Pi was used as an on-board computer (high-level control) and a HKPilot Mega 2.7 was used as an input/output unit (low-level control). The BlueROV's I/O consists of an HKPilot Mega 2.7 which is based on Ardupilot Mega. The HKPilot Mega 2.7 has the following on chip sensors: Magnetometer - HMC5883L, Barometer - MS5611-01BA, Inertial measurement unit (IMU) - MPU6000. An external pressure sensor MS5837-30BA which was encased in a watertight case by Blue Robotics was connected to the HKPilot Mega 2.7 by I2C. The HKPilot Mega 2.7 also controls the six ESCs.

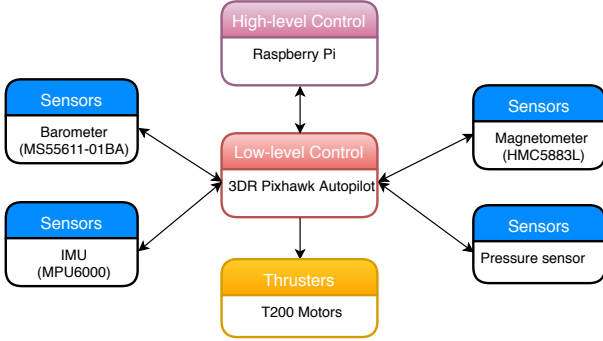


Figure 1. Physical structure of the existing system

B. General Model Driven Engineering and transformation of models

From the software point of view, the re-engineering strategy is translated into three phases on two levels of abstraction. The two level of model abstraction correspond to the traditional layers resulting from the approaches in a Model Driven Engineering (MDE). At the first level is the PSM, whose behavioral characteristics are profoundly dependent on the hardware and software platform and whose portability to another application is limited. The second level is the platform independent model (PIM) whose behavioral characteristics make it possible to extend it and bring it to new applications.

Meanwhile at the first phase, the extraction of a PIM model from the existing platform (in this case is BlueROV-1). The enrichment of this abstract and portable model with new extensions which are formation control, collision avoidance are added and researched in the second phase. Finally, the realization in a new platform PSM is implemented. The figure 2 shows out two levels and three phases in the general model driven engineering and transformation of models.

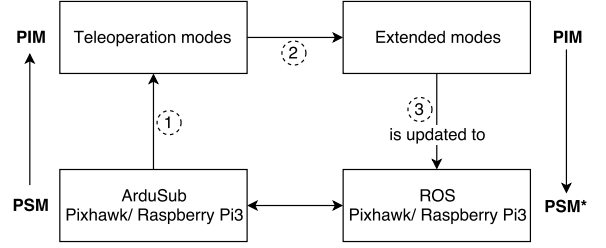


Figure 2. Model driven re-engineering of the unmanned underwater vehicles

III. MODEL PIM OF BLUEROV-1

A. Overview of UUV dynamics model

From the large field of guidance, navigation and control of underwater vehicles, the 6 degrees of freedom (DoF) dynamic model of AUVs in body frame [11], can be written in the following form:

$$\begin{cases} \dot{\eta} = J(\eta) \nu \\ M\dot{\nu} + C(\nu) + D(\nu)\nu + g(\eta) = \tau_{Act} + \tau_{Dist} \end{cases} \quad (1)$$

Where: $\eta = (x, y, z, \phi, \theta, \psi)^T$ is the position (NED: North, East and Down) and orientation (Euler RPY: Roll, Pitch, Yaw angles); $\nu = [u, v, w, p, q, r]^T$ is the velocity and angular velocity; $J(\eta)$ is the Jacobian matrix typed 6×6 for η ; $M = M_{RB} + M_A$ is a matrix, which denotes the 6×6 system inertia matrix containing M_{RB} - the generalized constant inertia matrix, and M_A - the added mass inertia matrix; $C(\nu) = C_{RB}(\nu) + C_A(\nu)$ is the 6×6 Coriolis and centripetal forces matrix including added mass; linear and nonlinear hydrodynamics damping are contained within the 6×6 matrix $D(\nu) = D + D_n(\nu)$, D contains the linear damping terms, and $D_n(\nu)$ contains the nonlinear damping terms; $g(\eta)$ is the 6×1 vector of gravitational and buoyancy effects; τ_{Act} is the 6×1 control input forces and torques; and finally τ_{Dist} is the 6×1 vector external disturbances caused by wind, waves and ocean current.

It is assumed that the external disturbances can be neglected, $\tau_{Dist} = 0$. Furthermore, neglecting the motions in *heave*, *roll* and *pitch*, $w = p = q = 0$ and $z = \phi = \theta = 0$. From the equation (1), the simplified equations of motion for *surge*, *sway* and *heading* in the body-fixed reference frame are given as

$$\dot{u} = -\frac{d_u u}{m_u} + \frac{m_v r v}{m_u} + \frac{\tau_u}{m_u} \quad (2)$$

$$\dot{v} = -\frac{d_v v}{m_v} - \frac{m_u r u}{m_v} - \frac{\tau_v}{m_v} \quad (3)$$

$$\dot{r} = -\frac{d_r r}{m_r} + \frac{m_{uv} u v}{m_r} + \frac{\tau_r}{m_r} \quad (4)$$

Where $m_u = m - X_{\dot{u}}$, $m_v = m - Y_{\dot{v}}$, $m_r = I_z - N_{\dot{r}}$, and $m_{uv} = m_u - m_v$ capture the effect of mass and hydrodynamic added mass terms, and $d_u = -X_u - X_{u|u}|u|$, $d_v = -Y_v - Y_{v|v}|v|$, and $d_r = -N_r - N_{r|r}|r|$ capture the hydrodynamic damping effects.

B. Architecture: From a tele-operation circuit to an autonomous

According to Yokokohji et al [12], the different control circuit architectures supporting several combinations of tele-operated or autonomous modes can be distinguished. A classification which seems interesting to us is that proposed in [12] which replies on series, parallel or mixed serial/parallel circuits. Four modes and associated circuits are described in Figure 3.

- In manual control, the control system supported by the embedded cards to perform continuous transformations of signals and models without making any decision.
- In serial type, manual mode and autonomous are switched in serial. Either the robot is entirely manual that found the previous case or it operates autonomously.
- In combined case, control inputs both from manual control and from autonomous control are mixed. This case is used when the autonomous function modifies the commands by the operator or when the operator want to modify the motion by autonomous functions.
- In shared case, the given task is shared among the manual control and autonomous control.

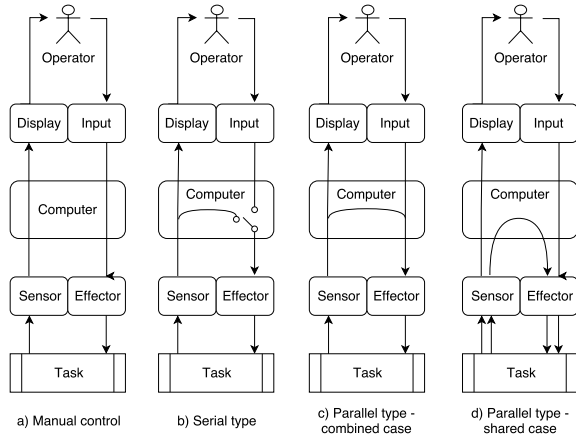


Figure 3. Four different configurations of modes and associated circuits

C. Equation dynamics of multi-UUV

In a study of the motion control system, Fossen [11] showed that the kinematic equations of motion of the vehicle in the inertial reference frame can be written as

$$\dot{x} = r_u \cos(\psi) - r_v \sin(\psi) \approx r_{u0} \cos(\psi) \quad (5)$$

$$\dot{y} = r_u \sin(\psi) + r_v \cos(\psi) \approx r_{u0} \sin(\psi) \quad (6)$$

$$\dot{\psi} = r_r \quad (7)$$

From the equation (5), (6), and (7), the kinematic equations in the inertial reference frame for the i^{th} UUV are

$$\dot{x}_i = r_{u_i} \cos(\psi_i) \quad (8)$$

$$\dot{y}_i = r_{u_i} \sin(\psi_i) \quad (9)$$

$$\dot{\psi}_i = r_{r_i} \quad (10)$$

Where $(x_i, y_i) \in \mathbb{R}^2$ is the Cartesian coordinates for the i^{th} vehicle, $\psi_i \in [0, 2\pi)$ is the orientation of the i^{th} vehicle with respect to the world frame and u_i, r_i are the linear and angular velocity inputs, respectively.

It is possible to feedback linearization for a fixed point off the center of the vehicle denoted as $(\xi_i = (\xi_{xi}, \xi_{yi})^T$, where

$$\xi_{xi} = x_i + d_i \cos(\psi_i) \quad (11)$$

$$\xi_{yi} = y_i + d_i \sin(\psi_i) \quad (12)$$

Letting

$$\begin{bmatrix} r_{u_i} \\ r_{r_i} \end{bmatrix} = \begin{bmatrix} \cos(\psi_i) & \sin(\psi_i) \\ -1/d_i \sin(\psi_i) & -1/d_i \cos(\psi_i) \end{bmatrix} \begin{bmatrix} \kappa_{xi} \\ \kappa_{yi} \end{bmatrix} \quad (13)$$

gives

$$\begin{bmatrix} \dot{\xi}_{xi} \\ \dot{\xi}_{yi} \end{bmatrix} = \begin{bmatrix} \kappa_{xi} \\ \kappa_{yi} \end{bmatrix}, \quad \text{for } i = 1, 2, \dots, n \quad (14)$$

Where $\kappa_i = [\kappa_{xi}, \kappa_{yi}]^T$ is the signal control input for i^{th} vehicle.

In this paper, we focus on the serial type, which includes the following modes: formation control and collision avoidance. Specific state transition steps are presented as figure 4.

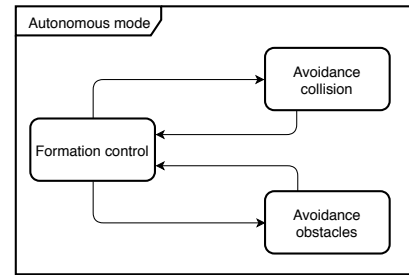


Figure 4. Flow chart of modes of vehicles

According to Wei Ren [5], let $G = [g_{ij}] \in \mathbb{R}^{n \times n}$ be the adjacency matrix. $\xi_i - \xi_j \rightarrow \delta_i - \delta_j$ if and only if the directed graph of G has a directed spanning tree. A consensus algorithm with a time-varying reference state is given as

$$\kappa_i^{fo} = - \sum_{j=1}^n g_{ij}(t) [(\xi_i - \delta_i) - (\xi_j - \delta_j)] \quad (15)$$

Where $\xi_i \in \mathbb{R}^m$ are the state of the i^{th} vehicle which are defined in the equations 11, 12; $\delta_i - \delta_j, \forall i \neq j$, denotes the desired separation between the information states; g_{ij} is 1 if information flows from vehicle j to vehicle i and 0 otherwise, $\forall i, j \in \{1, \dots, n\}$. For more details on the consensus algorithms that are presented in [4], [5], [13] and [14].

Remark. In this paper, the desired formation shape $\delta^* = [(\delta_1)^T \dots (\delta_n)^T]^T \in (\mathbb{R}^d)^n$ is fixed overtime.

When the vehicles are in autonomous mode, if the distance between vehicles $< R$, the collision avoidance will be activated. According to Mondal et al [6], when a vehicle is inside

the collision avoidance region of another vehicle, a strictly decreasing smooth potential function $V(\xi)$ can be introduced into action to avoid collision.

The function $g(x)$ which helps to vary the potential function smoothly, is defined as

$$g(x) = \begin{cases} k_1 & \text{if } x \in (0, d_{in}), \\ k_2(x) & \text{if } x \in [d_{in}, D_{out}], \\ 0 & \text{if } x \notin (0, D_{out}). \end{cases} \quad (16)$$

Where $k_1 = 1$, $x = \|\xi_i - \xi_j\|$ (x is the distance between vehicles) and

$$k_2(x) = \frac{1}{2} \left[1 + \cos \left(\pi \frac{x - d_{in}}{D_{out} - d_{in}} \right) \right]$$

To generate a gradient based the collision avoidance region, a smooth collective potential function is defined as

$$V(\xi) = \frac{1}{2} \sum_i \sum_{j \in N_i} \psi_c(\|\xi_i - \xi_j\|) \quad (17)$$

Where

$$\psi_c(x) = \int_{D_{out}}^x \phi(s) ds$$

With

$$\phi(x) = \begin{cases} \frac{-x}{\sigma_{ij} + x^2} & \text{if } x \in (0, d_{in}), \\ \frac{-x}{2(\sigma_{ij} + x^2)} [1 + \cos(\pi h)] & \text{if } x \in [d_{in}, D_{out}], \\ 0 & \text{if } x \notin (0, D_{out}). \end{cases}$$

$$h = \frac{x - d_{in}}{D_{out} - d_{in}}$$

According to Mondal et al [6], finally, the collision avoidance term κ_i^{ca} can be written as

$$\kappa_i^{ca} = -\nabla_{p_i} V(\xi) = \sum_{j \in N_i} \phi(\|\xi_j - \xi_i\|) \frac{\xi_j - \xi_i}{\|\xi_j - \xi_i\|} \quad (18)$$

IV. FRAMEWORK OF A MODEL FROM PIM TO PSM

To apply the MDE method, we have introduced a framework to study as shown in Figure 5. The framework consists of two parts: PIM and PSM. For the PIM layer, the algorithms which do not depend on a particular model, such as collision avoidance algorithms, collision avoidance algorithms, and high-level control algorithms are developed on Matlab/Simulink to quickly verify the performance of the algorithms. The UUV operation modes which are developed and controlled based on Stateflow as shown in Figure 6 are including: teleoperation, autonomous and maintenance mode. It is noteworthy that within this paper, we did not investigate the depth control issues but rather focus on the problem of the position and attitude of unmanned underwater vehicles. To transfer the control signals κ_i^{fo} and κ_i^{ca} of the equation (15) and (18)

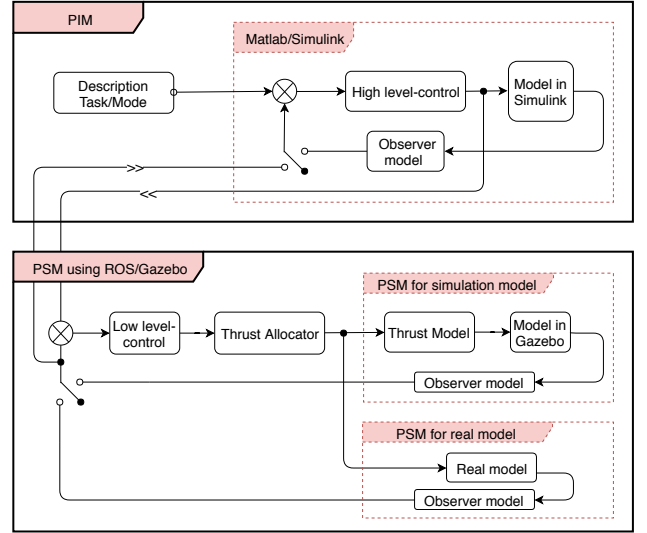


Figure 5. Flow information between PIM and PSM

from the PIM model to the PSM model, according to Chen et al [15] the following consensus control is designed

$$\tau_i = -R_i^T \left(\left(\kappa_i^{fo} + \kappa_i^{ca} \right) + \text{sgn}(R_i \nu_i) \left\| \kappa_i^{fo} + \kappa_i^{ca} \right\| \right) - R_i^T R_i \nu_i + g_i \quad (19)$$

Where $\tau_i \in R^3$ is the control force vector for vehicle i , R_i is the transformation matrix from the body-fixed reference frame to the inertial reference frame and $g_i = [(W_i - B_i) s_{\theta_i}, -(W_i - B_i) c_{\theta_i} s_{\phi_i}, -(W_i - B_i) c_{\theta_i} c_{\phi_i}]^T$, where W_i and B_i represent the gravitational and buoyancy forces, respectively. $\nu = [u, v, w, p, q, r]^T$ is defined in the equation 1.

V. SIMULATION OF RESULTS

A. Model of PIM in Matlab/Simulink

Before transferring the algorithm (15) and (18) to a low-level model, we made the model of the simulation in Matlab/Simulink to verify the theoretical results and demonstrate the effectiveness of formation control and collision avoidance of multiple platform-independent models. The desired formation configuration δ_i is predefined by orthogonal coordinates as $(\delta_{x1}, \delta_{y1}) = (6, 6)$, $(\delta_{x2}, \delta_{y2}) = (0, 0)$ and the initial states of the vehicles are $(\xi_{x1}, \xi_{y1}) = (0, 0)$, $(\xi_{x2}, \xi_{y2}) = (6, 0)$. The parameters of the simulation are chosen as follows: $\alpha_{ij} = 1$, $k_{ij} = 1$. The outer region $D_{out} = 2m$, the inner region $d_{in} = 1m$. In figure 9 illustrate the distances among the vehicles without the collision which gradually converges to the value $8.484m$. This value guarantees the achievement of the desired formation and without collision between the vehicles.

B. Model of PSM in ROS/Gazebo

After verifying the effectiveness of the independent-model algorithms in Matlab/Simulink, we transfer the algorithms to a specific physical model. In this case, it is the platform BlueROV-1 with the parameters such as linear damping,

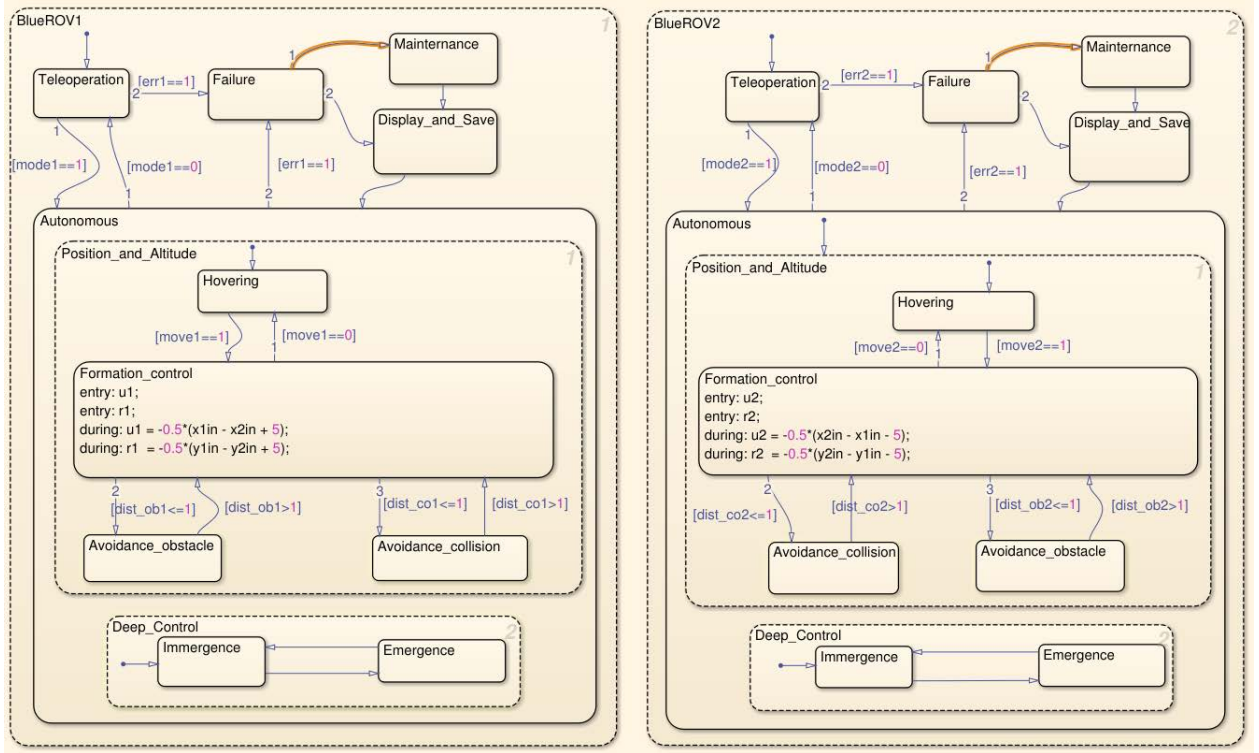


Figure 6. Stateflow of a cooperation of two low-cost UUVs on Matlab/Simulink

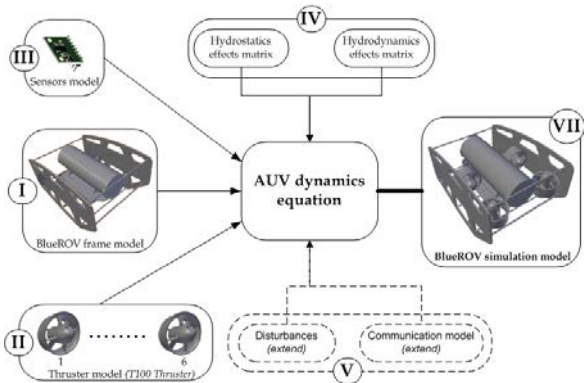


Figure 7. Simulation model of PSM in ROS/Gazebo

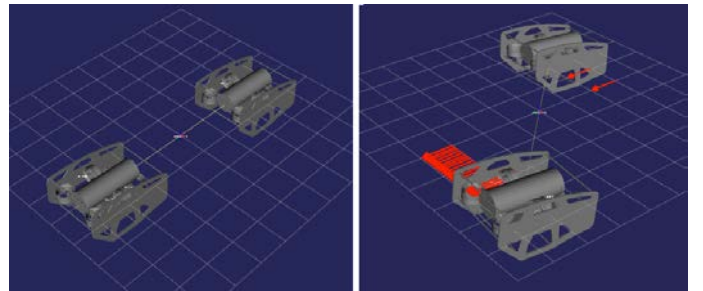


Figure 8. The initial position (left) and the formation position (right) of two BlueROVs in RViz

quadratic damping, inertia and increased inertia, mass added and 3D geometric are determined in [17], [18]. Beside, according to Manhães [16], we have developed the model of simulation for BlueROV-1 in ROS/Gazebo which simulates dynamics models using Simulation Description Format (SDF) or Unified Robot Description Format (URDF) is shown as the figure 7 and the figure 8. It can be observed from the figure 10 the control forces of two UUVs gradually converge on the value 0. Although affected by momentum inertia on specific platforms, the distance among vehicles is also reached and oscillates around the value $8.48m$ (as shown in figure 11). It is clear that this value is as close as the simulation results of section V-A.

VI. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have studied the analysis of coordination modes for multi-UUV based on Model Driven Architecture approach. Specially, we made the use-case of formation control and collision avoidance for multi-BlueROV-1. We then transformed from PIM level to PSM level as figure 2. Research extending the experiments to multi-BlueROV platform will also be conducted where the issues of time delay, effects of the external disturbances caused by waves and ocean current. In addition, the depth control issues will also be incorporated into the scenarios. We will also extend the application of the framework to various robot platforms as well as the change of control algorithms and the effects of underwater communication constraints.

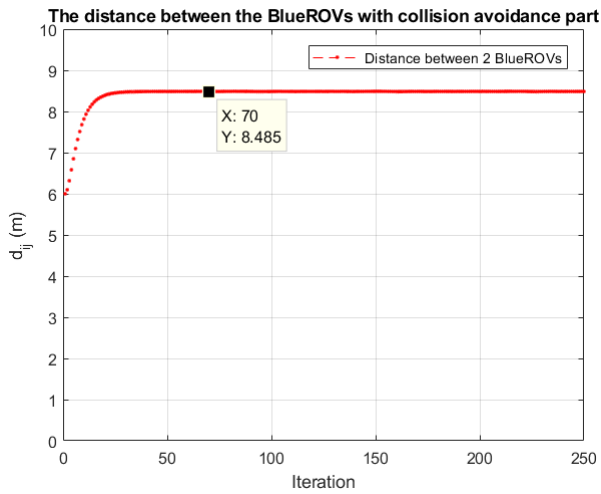


Figure 9. The distance among vehicles without the collision avoidance on the PIM layer

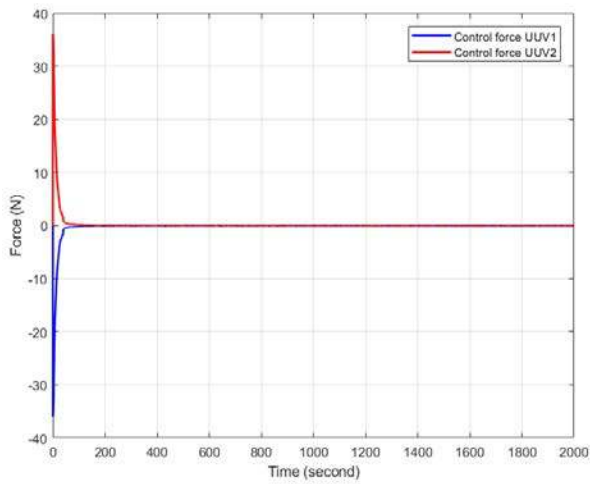


Figure 10. The control force of two UUVs on the PSM layer

ACKNOWLEDGMENT

This work was achieved thanks to specific funding from the Ministry of National Education France, Université de Toulon and QUARZ, Laboratory SUPMECA, Paris.

REFERENCES

- [1] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, mar 2015.
- [2] Y. Wang and W. Yan, "Path parameters consensus based formation control of multiple autonomous underwater vehicles in the presence of ocean currents," in *2012 17th International Conference on Methods Models in Automation Robotics (MMAR)*, Aug 2012, pp. 427–432.
- [3] Z. Peng, D. Wang, Z. Chen, X. Hu, and W. Lan, "Adaptive dynamic surface control for formations of autonomous surface vehicles with uncertain dynamics," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 513–520, March 2013.
- [4] R. B. Wei Ren, *Distributed Consensus in Multi-vehicle Cooperative Control Theory and Applications*. Springer-Verlag London, 2008. [Online]. Available: <http://www.springer.com/gp/book/9781848000148>

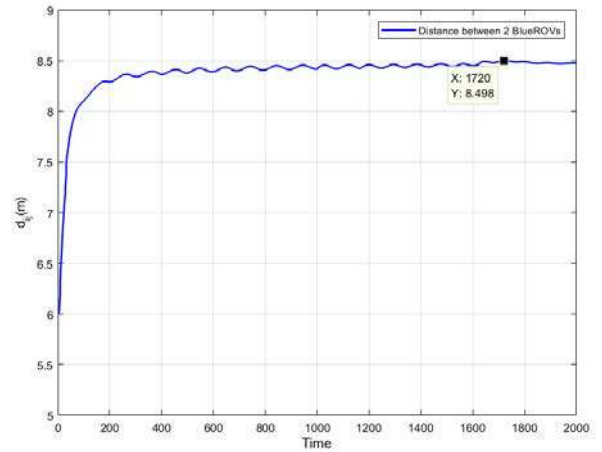


Figure 11. The distance among vehicles without the collision avoidance on the PSM layer

- [5] W. Ren, "Multi-vehicle consensus with a time-varying reference state," *Systems and Control Letters*, vol. 56, no. 7, pp. 474 – 483, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167691107000114>
- [6] A. Mondal, L. Behera, S. R. Sahoo, and A. Shukla, "A novel multi-agent formation control law with collision avoidance," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 558–568, 2017.
- [7] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *2005 American Control Conference ThA04.2 June 8-10, 2005. Portland, OR, USA, 2005*.
- [8] X. Ge, Q.-L. Han, D. Ding, X.-M. Zhang, and B. Ning, "A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems," *Neurocomputing*, oct 2017.
- [9] OMG, "Mda guide version 2.0," 2014. [Online]. Available: <http://www.omg.org/mda/>
- [10] BlueRobotics, "Bluerov," 2016. [Online]. Available: <https://www.bluerobotics.com/>
- [11] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley, 2011.
- [12] Y. Yokokohji, A. Ogawa, H. Hasunuma, and T. Yoshikawa, "Operation modes for cooperating with autonomous functions in intelligent teleoperation systems," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, May 1993, pp. 510–515 vol.3.
- [13] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 324 – 333, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889007001108>
- [14] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, June 2017.
- [15] S. Chen and D. W. Ho, "Consensus control for multiple auvs under imperfect information caused by communication faults," *Information Sciences*, vol. 370-371, pp. 565 – 577, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025516302869>
- [16] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *OCEANS 2016 MTS/IEEE Monterey*, Sept 2016, pp. 1–8.
- [17] O. Alexander and Eidsvik, "Identification of hydrodynamic parameters for remotely operated vehicles," Ph.D. dissertation, Norwegian University of Science and Technology, 2015.
- [18] A. Aili and E. Ekelund, "Model-based design and development and control and of and an underwater and vehicle," Master's thesis, Master of Science Thesis in Automatic Control Department of Electrical Engineering, Linköping University, 2016.