Simulation of Hybrid Autonomous Underwater Vehicle based on ROS and Gazebo

1st Thuyen–An Ngo
Ho Chi Minh City University of Technology, VNUHCM
Vietnam
thuyen.ngo5371@hcmut.edu.vn

2nd Ngoc-Huy Tran(✉)
Ho Chi Minh City University of Technology, VNUHCM
Vietnam
tnhuy@hcmut.edu.vn

3rd Hung Nguyen
HUTECH Institute of Engineering, Ho Chi Minh City University of Technology (HUTECH)
Vietnam
n.hung@hutech.edu.vn

4th Thien-Phuc Tran
Ho Chi Minh City University of Technology, VNUHCM
Vietnam
ttphuc.rectie@hcmut.edu.vn

Abstract - Simulation is a crucial task in algorithm testing for autonomous vehicles. Furthermore, simulation also helps to observe the possible errors precedent to the empirical experiment, thus optimizing the time and effort of researchers. This paper will illustrate the simulation method of Hybrid Autonomous Underwater Vehicle(AUV), a new AUV platform utilizing Robotic Operating System (ROS) and Gazebo. Gazebo will simulate 3D behaviors and movements of AUV based on calculations regarding the effect of hydrodynamic and hydrostatic force, the objective relation of the actuator as well as the sensor simulation, in which ROS is used for control algorithm development.

Keywords—Hybrid AUV, Gazebo, ROS

I. INTRODUCTION

AUV is becoming increasingly popular and plays a vital role in the life of people. There are multiple applica-tions of AUV such as sub-bottom survey[1], data collection, underwater biological mapping[2], in the military field, AUV could also be used for search and detection of mines/bombs[3], or becomes an explosives carrier for underwater warfare. Thus, the research and making AUV are considerably necessary in recent times.

The making of AUV and algorithms research should be repetitive for ensuring stable and sustainable control. A reliable methodology is analysing physical characteristics of AUV in a simulation environment along with an ever-running data collection and evaluation program. However, this method has a major drawback is that it is time and computing resources consuming, hence a robust computer is needed for its operation. In recent years, an answer for that has been proposed by simulation programs such as MATLAB/Simulink[1], OpenSim[2], UWSim[3], Webots[4] . In this paper, the dynamics of AUV will be calculated and indicated using Gazebo, a open-source simulation program with a large supporting community and the ability to enable 3D and multi Physics Engine simulation. The combination with ROS allows researchers to embed from fundamental algorithms such as Proportional Integral Derivative (PID) Controller, to advance ones like Sliding Mode Controller (SMC)[4], Backstepping Controller (BC)[5] or enforcing navigation algorithms such as Line Of Sight (LOS), Pure Pursuit (PP)[6] , etc. Additionally, ROS is also used as a middle-ware (between hardware and software) which can easily translate simulation results to real AUV without costing additional time for translating and debugging. Overall, the AUV simulation based on ROS and Gazebo could not only utilize modern algorithms but also perform debugging in the testing.

This article will have the following structure: **Section II -** system architecture, **Section III -** AUV Gazebo model and control algorithm, **Section IV -** simulation result and discussion, **Section V -** conclusion and future work.

II. SYSTEM ARCHITECTURE

The system consists of 3 main components as follows: Gazebo, ROS and Graphical user interface (GUI).

The model will be described and configured by Unified Robot Description Formart(UDRF) and Semantic Robot Desciption Fomart (SRDF). These files would determine the size, weight, moment of inertia, center of mass, etc and Gazebo will calculate and indicate it in defined environments. Gazebo will create inputs and outputs in connection with ROS. ROS will read and write the states created by Gazebo to indicate them on GUI, while receiving task from GUI and transmit back to Gazebo making a closed cycle.In this simulation system, the version of softwares used are: Gazebo 9.0, ROS Melodic distro, GUI using VIAM-AUV-GC V0.0.1 coded in Qt 5.13.

[1] https://www.mathworks.com/
[2] https://simtk.org/
[3] http://www.irs.uji.es/uwsim/
[4] https://cyberbotics.com/

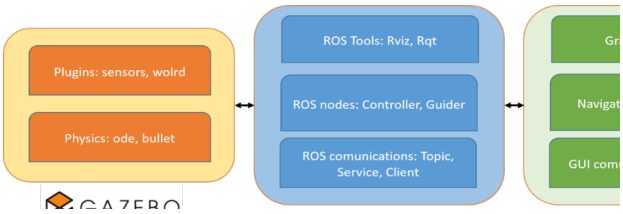

Fig. 1 System architecture

The AUV simulation process as follows:

1) Building AUV model on Gazebo
2) Building necessary plugins to accurately describe the forces and moment impacting the AUV
3) Building AUV steering strategy based on ROS.

The ROS will be compiled by a popular program-ing language such as C++ or Python. ROS interfaces betw-een files and with Gazebo through nodes in which it can publish or subscribe the topics. Helping ROS to understand file architecture as well as interface with Gazebo in the simulation process is not an easy task. There are at least 3 required parts which are ***description***, ***gazebo_simulation*** and ***control***. In description, it is necessary to have information about simulated objects like sizes, colour, material, weight, moment of inertia. The main components are:

- meshes: this folder contains mesh files which are the parts and devices creating AUV in the form of COLLADA.
- launch: this folder contain one or many file *.launch for AUV test operation when the plugins are absent.
- urdf: this folder contains file *. xacro (XML Macro file) to define size, impact, material, colour and refer to the corresponding mesh files in the Meshes of each component.

gazebo_simulation contains information about the world in which AUV would be spawned, include the following components:

- worlds: this folder consists of *.world files containing information about simulation environment
- launch: this folder contains *.launch files for simulation program operation, while performing a series of tasks to call the programs in a logical process: call world, spawn models, config controller, connect to GUI, ect.
- include: this folder contains necessary header files for plugins operation process.
- src: this folder contains necessary plugins files for simulation.

control contains information about the control strategy in with the simulation program could be run, including the following components:

- include: this folder contains necessary header files for the controller.
- src: this folder contains controller files.

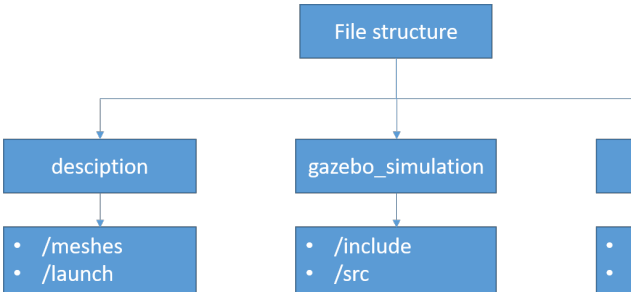

Fig. 2 The folder tree described

III. AUV GAZEBO MODEL AND CONTROL ALGORITHM

The AUV used is AUV2000, an AUV type researched and developed by VIAMLAB. HCMUT, with the following fundamental parameters:

Table 1. AUV Parameters

Para-meters	Value	Units	Description
m	98.000	Kg	Mass
l	2.5900	m	Length
d	0.2500	m	Hull diameter
I_{xx}	1.3575	$Kg.m^2$	Moments of Inertia about x axis
I_{yy}	28.0980	$Kg.m^2$	Moments of Inertia About y axis
I_{zz}	28.6999	$Kg.m^2$	Moments of Inertia about z axis
W_{span}	1.5460	m	Wingspan
CG	$(0,0,-0.008)$	m	Center of Mass
CB	$(0,0,0)$	m	Center of Buoyancy

A. AUV description package

AUV200 is simulated in Gazebo with the following modules:

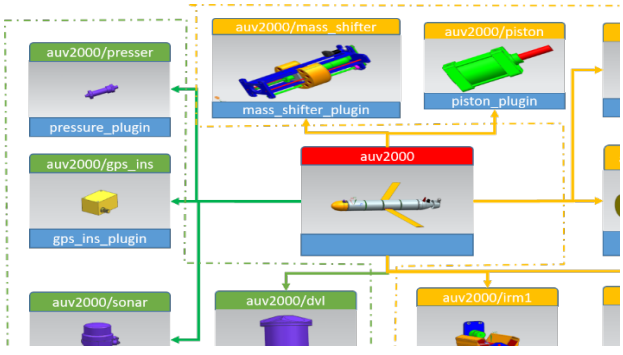

Fig. 3 Component connecting diagram

Gazebo environment includes information about models, physics engine, plugins and sensors. Specifically. ,models contain the links connecting the components together with appropriate joints. At first, AUV was designed in CAD, this CAD model was then imported into Gazebo in COLLADA format (with *,dae extension), all will be merged into the mentioned meshes folder. For each part of the AUV, if it is a sensor, there would be a ROS plugin sensor to simulate the operation and function of that sensor. For the actuator internal and external:

- Internal: Internal rolling mass(IRM) in the AUV to cope with the roll phenomena; mass shifter to control AUV's pitch angle; piston changes the buoyancy of the AUV for Glider mode.
- External: Thruster is the main propulsion system of AUV; Rudder for heading control.

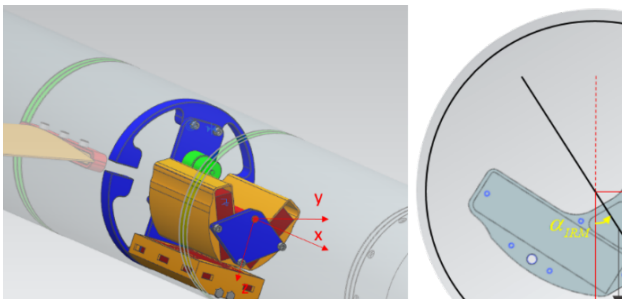

Fig. 4 Internal Anti-rollinh mechanism of AUV2000

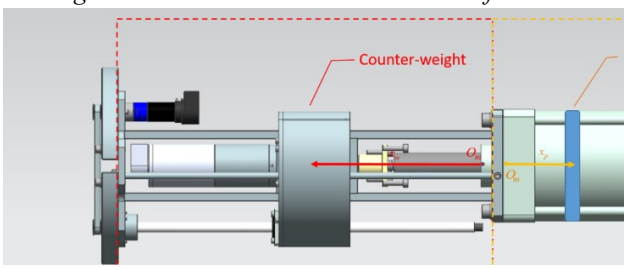

Fig. 5 Counter-weight structure and ballast system

The internal actuator would be calculated by Gazebo based on Physics Engine, we only need to change the position of IRM, mass shifter, piston through ROS topic without adding any additional force or moment. The other external actuator will be considered in upcoming AUV Dynamics section.

B. AUV Dynamics

For the kinematics of rigid bodies, Gazebo itself provides AUV the simulation through what is defined in the description such as mass, moment of inertia, impact. Thus, we can easily get the parameters about speed, acceleration to calculate external force and moment in [7], [8] and through *AddRelativeForce (const ignition::math::Vector3d &_force)* and *AddRelative-Torque (const ignition::math::Vector3d &_torque)* available in Link Class Reference of Gazebo.

Remark 1: The default frame in Gazebo is ENU(East-North-Up), hence the values of acceleration, speed need to be converted to NED(North-East-Down) to match the functions shown in [7], [8].

Remark 2: The adding of external force through *AddRelativeForce/Torque* could lead to system instability if the Added mass, damping parameters are too large. A simple solution for this problem is to add a low-pass filter when values of acceleration, speed are read to ensure that they would not change too quickly resulting in the instability of the system.

C. Control Algorithm

The control algorithm for AUV2000 would be written in package control as shown in section II Due to being a Hybrid AUV, AUV200 could flexibly switch between Glider and AUV modes. In AUV mode, when the propulsion is in operation, the torque of the motor will make the AUV rolls, damped oscillation would occur for the AUV roll angle and be set at non-zero value. To keep it simple, a PID controller would be used to keep the roll angle stable. Similarly, when controlling roll angle with pitch and yaw, PID will be preferred due to its simplicity.

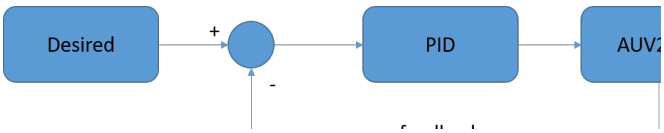

Fig. 6 PID controllers used

To control the depth for AUV, as shown in[7], the depth control strategies in 2 modes Glider and AUV would use the mentioned PID controller to respond.

Depth control:

Glider Mode: In this, AUV2000 will use continuous diving and floating inertia to keep moving forward. Assuming that the AUV will operate at a defined depth z_1 and z_2 as above. An error parameter z_e is defined to control the current depth of the AUV as well as knowing whether its operational status is diving or floating, additionally, δ is the distance to compensate for the inertia when the AUV changes from diving to floating or vice versa. We could apply the following strategy:

- When the Hybrid AUV dives, the cylinder will suck up the water to make $B < W$ and control so that the pitch angle is -30^0 downward ($\theta_d = -30^0$) (Downward phase)
- When the Hybrid AUV floats, the cylinder will release the water to make $B < W$ and control so that the pitch angle is 30^0 upward ($\theta_d = 30^0$) (Upward phase)

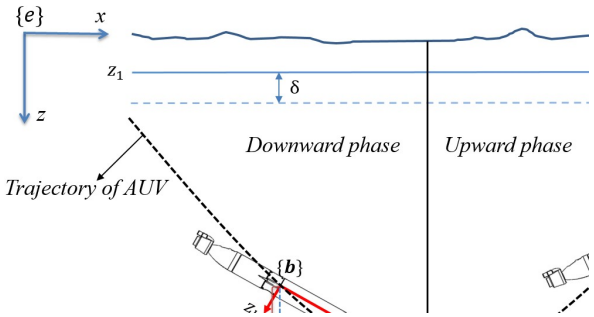

Fig. 7 Glider control strategy

AUV mode: Common applications of AUVs operating in propulsion mode often require the vehicle to travel at fixed depths (such as collecting environmental data, scanning the seabed map, minesweeping ,...), so in this section we will build a control strategy that can drive the AUV to a given fixed depth.

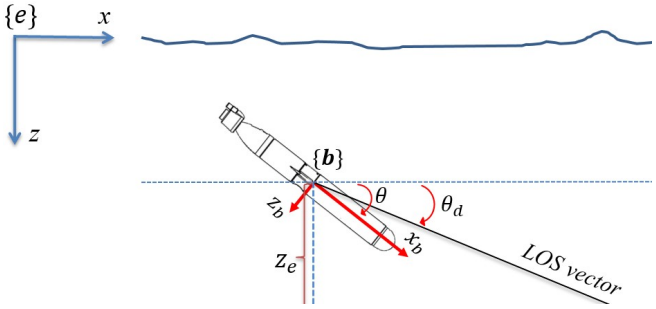

Fig. 8 LOS vector definition

The main task is to adjust the piston so that $B = W$ .Suppose that we want the AUV to reach and maintain its position at a certain depth z_d ,we will define a LOS vector to calculate θ_d angle, and steer the AUV to the desired depth with pitch controller. We could easily calculate from the picture that:

$$\theta_d = \arctan\left(\frac{z_e}{\Delta}\right) \qquad (1)$$

With the desired pitch angle calculated, the problem of aut pitching could be solved by simply using a PID controller. In that, Δ is a custom positive coefficient and $z_e = z - z_d$.

IV. SIMULATION RESULT AND DISCUSSION

This section presents the simulation results of AUV2000 on Gazebo environment with the world of auv_underwater_world.world created by[9]. The parameters of the controllers are given in the following table:

Table 2. Controller parameters

Controller	parameters
Roll	$K_P = 1.75, K_I = 1.25, K$
Pitch	$K_P = 1.25, K_I = 0.1, K$
Yaw	$K_P = 5, K_I = 0, K_D$
Surge(AUV Mode)	$K_P = 8, K_I = 6, K_D$ =
AUV Depth	$\Delta = 7.$

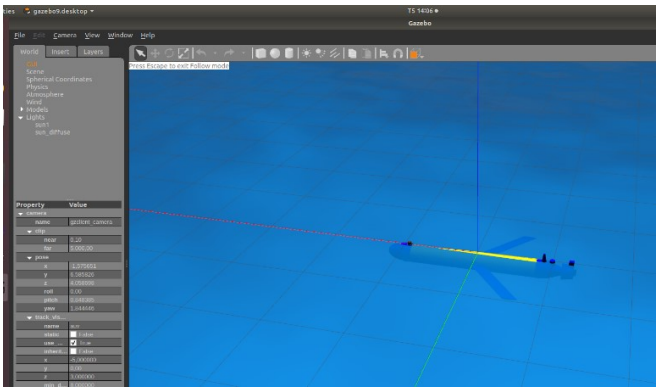

Fig. 9 AUV2000 spawned in the Gazebo enviroment

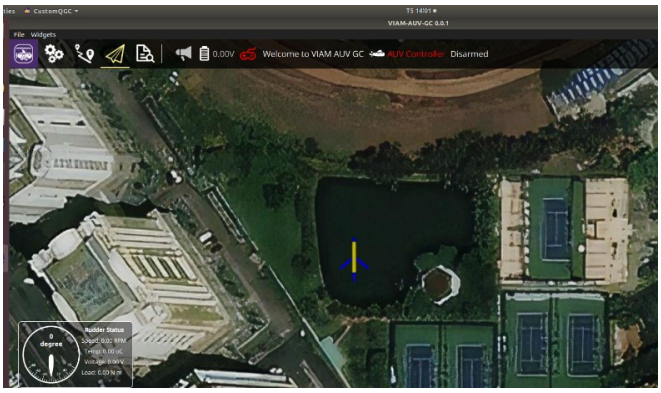

Fig. 10 AUV2000 simulated on Gazebo in connection with VIAM-AUV-GC-0.0.1 GUI

A. Glider mode depth control

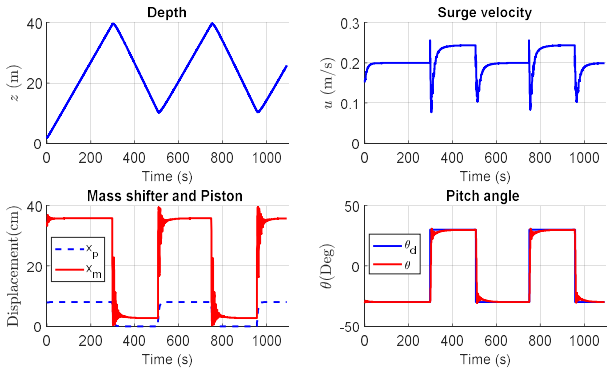

Fig. 11 Depth response in Glider mode

In the Glider mode, the required depth of survey is 10m and 40m which the AUV2000 fully meets. Pitch angle PID controller responds well to pitch angle when floating as well as diving. However, if looking closely at the response of the

Mass Shifter, it fluctuates when the setting is changed. This can be explained that this actuator has a certain delay, and is reasonable with reality. Moreover, with Glider's continuous floating/diving strategy, the forward velocity is quite low, namely 0.2 m/s when diving and 0.25 m/s when floating. This difference in floating speed is because the AUV2000 is designed when the piston is in the middle position, the buoyant force will be higher than gravity, in accordance with the safety requirements of the AUV.

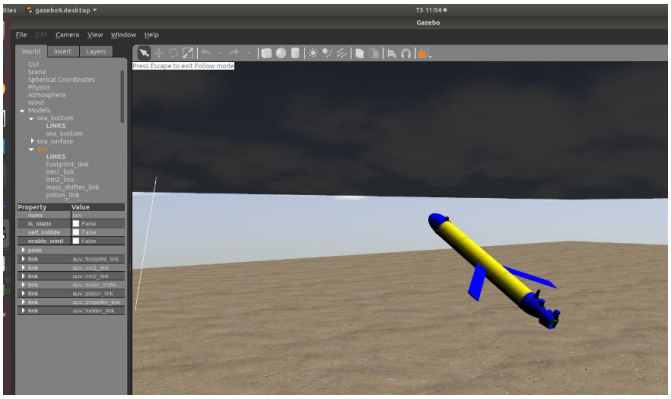

Fig. 12 In Glider mode, the AUV2000 is the process of floating

B. AUV mode depth control

In AUV mode using propulsion device, the displacement of the piston is 4 cm at this time so that $W \approx l$, and the speed is also controlled at 1 m/s, with the LOS method according to the depth, the AUV2000 can fully meet many different set values with few overshoot and the error goes to zero. In this mode the dive time is about 30s with 15m, infers a speed of about 0.5m/s, and floats about 50s with 30m, infers a speed of about 0.6 m/s. In Figure 13, we also see that, when the set Pitch angle changes continuously according to the LOS vector, the PID Pitch controller also gives a pretty good response.

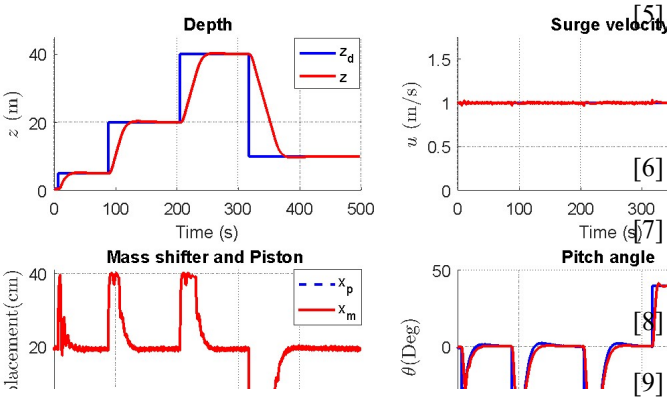

Fig. 13 Depth response in AUV mode

V. CONCLUSION AND FUTURE WORK

This paper describes how to simulate a Hybrid AUV on Gazebo - ROS Framework. Moreover, this simulation can be combined with VIAM-AUV-GC 0.0.1 GUI to easily verify the operation of AUV in the Gazebo environment in particular and multi-software integration in general. Both operating modes of this type of AUV, Glider and using propulsion, are presented on the control strategy as well as using that control strategy with the PID controller. All simulation results are visualized on Gazebo and numerical results.

Future work is to build a complete control package to perform a complete simulation of a task that AUVs must have: trajectory planning - trajectory tracking - bottom scanning data collection.

ACKNOWLEDGEMENT

This research is funded by Ho Chi Minh City University of Technology (HCMUT), VNU-HCM under grant number B2018-20b-01. We acknowledge the support of time and facilities from HCMUT, VNU-HCM and Laboratory of Advance Design and Manufacturing Processes, HCMUT for this study.

REFERENCES

[1] S. Marini *et al.*, “Enduruns: An integrated and flexible approach for seabed survey through autonomous mobile vehicles,” *J. Mar. Sci. Eng.*, vol. 8, no. 9, 2020, doi: 10.3390/JMSE8090633.

[2] N. Barrett, J. Seiler, T. Anderson, S. Williams, S. Nichol, and S. Nicole Hill, “Autonomous underwater vehicle (AUV) for mapping marine biodiversity in coastal and shelf waters: Implications for marine management,” *Ocean. IEEE Sydney, Ocean. 2010*, 2010, doi: 10.1109/OCEANSSYD.2010.5603860.

[3] P. E. Hagen, “Automated mine detection , classification and identification with AUV,” no. December, 2017.

[4] L. Rodrigues, P. Tavares, and M. Prado, “Sliding mode control of an AUV in the diving and steering planes,” *Ocean. Conf. Rec.*, vol. 2, pp. 576–583, 1996, doi: 10.1109/oceans.1996.568291.

[5] W. Chen, Y. Wei, and J. Zeng, “Back-stepping control of underactuated AUV’s depth based on nonlinear disturbance observer,” *Chinese Control Conf. CCC*, vol. 2015-Septe, pp. 6061–6065, 2015, doi: 10.1109/ChiCC.2015.7260587.

[6] A. M. Lekkas, *Guidance and Path-Planning Systems for Autonomous Vehicles*, no. April. 2014.

[7] C. T. Hải, T. N. Huy, T. T. Phương, and H. M. Diễn, “Xây dựng bộ điều khiển các chế độ lặn khác nhau cho VIAM-AUV2000,” vol. 3, pp. 114–128.

[8] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011.

[9] M. M. M. Manhaes, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, “UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation BT - 2016 OCEANS MTS/IEEE Monterey, OCE 2016, September 19, 2016 - September 23, 2016,” *Ieee*, 2016, [Online]. Available:

http://www.openscenegraph.org/%0Ahttp://dx.doi.org/
10.1109/OCEANS.2016.7761080.