WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Autonomous Under Water Robot

Aman Prasad Sinha[1], Arvind Kumar Sahu[2], Pradip Kumar Chandra[3], Triambak[4], Ms. Ranjeeta Krishan[5]

[1 2 3 4 5]Dept. Computer Science Engineering, Bhilai Institute of Technology

ABSTRACT –

Underwater robot prototyping and testing are sometimes time-consuming and expensive, and realistic simulation can significantly aid in validating numerous areas of the project. The described system is connected with ROS via the Gazebo dynamic simulator and the underwater image renderer UWSim in order to take advantage of currently available software. With contact mechanics, buoyancy, hydrodynamic damping, and low-level PID control, the entire system enables realistic portrayal of dynamic multi-robot simulation The choice of a competition setting that mirrors those in commercial applications will increase the applicability of the research work done. A new robotics program for autonomous underwater vehicles (AUVs) is something we want to start..

Index Terms or keywords –Autonomous Under Water Robot : Simulation Model, ROS, GAZEBO, UW-Simulation, Black Box Testing.

INTRODUCTION

Unmanned underwater vehicles, or AUVs, are technically autonomous underwater vehicles.
AUVs can be used for underwater surveying tasks like mapping submerged rocks, wrecks, and other objects that could pose a danger to commercial and recreational vessels trying to navigate.[1] More over two thirds of the world is covered by the oceans. Only 15% of the world's oceans have actually been explored. There has been little ongoing activity in mining or other areas of substantial economic and societal significance, and the majority of the exploitation of the accessible ocean resources has been related with offshore oil and gas production, offshore tourism, and fishing.[2] Robotic submarines, or autonomous underwater vehicles (AUVs), are a particularly difficult research topic that are appreciated for both their replace and expand capabilities since they may be used in risky areas without endangering human operators. An autonomous underwater vehicle, commonly referred to as an unmanned underwater vehicle, is an underwater vehicle that can populate itself. It is a three-dimensionally able robotic object that is propelled through the water by a population system, managed by an on-board computer. Underwater vehicles come in many varieties, primarily falling into the manned and unmanned systems categories. They can also be divided into a variety of categories, such as unmanned systems towed by ships.[3] The utilization of autonomous underwater vehicles rather than the traditional remotely operated vehicles and manned submersibles is a significant development in underwater robotics. Since it is never simple to deploy a crew on a surface vessel or operators in a submersible, risk and expense are greatly avoided. However, due to the climate and the nature of the vehicle, AUV testing is exceedingly challenging. Water tanks may be used for little experiments like low-level control and simple prototyping, but this already takes up space and money. Higher-level experiments, such as navigation, waypoint following, mapping of the seafloor, or sensor-based control, are intended to be conducted in open areas, which incurs large costs,
Simulators have been created to aid in the prototyping of AUV control and design for these reasons. They enable full real-time access to all data during an experiment, enabling the AUV to be significantly improved prior to the actual trials. In the year 2008, a study on AUV simulators was conducted. Since then, the ROS framework has become more widely utilized, and it has even obtained its own AUV simulator, known as UWSim. This simulator, which has been heavily utilised in the Trident project, uses OpenSceneGraph1 (OSG) and osgOcean2 to produce realistic pictures. osgOcean was created to produce realistic underwater photographs in OSG, an open source 3D graphics programme. a comparison between traditional rendering and rendering focused on the undersea environment. Multiple under water or surface vehicles can be present in the same simulateion, which enables carrying out experiments that would be very challenging in the real world. Embedded cameras, rudimentary sonar, and other AUV sensors are enabled.

LITERATURE REVIEW

ICube Laboratory, University de Strasbourg [5] The ROS topic that provides the strength and direction of the water flowing is subscribed to by the Gazebo simulator. Thus, it is possible to change the current as necessary. The direction of buoyancy is, of course, the opposite of the direction of gravity and vanishes as the vehicle rises above the surface. This plugin is managed by a Gazebo world file that is provided. The forces and torques that act on a body are inputs into Gazebo's dynamic simulation. Any form of effort may be added using plugins, allowing one to take a hydrodynamic model into consideration. The dynamic model of a robot connection and the approximations used in the implementation are described in this section.

David Nakath1, Mengkun She1, Furkan Elibol, Yashvi Mehta, Smrithi Agrawal, Sunil Ghane [6] UUV Simulator, WaterGAN, and UW IMG SIM are three cutting-edge techniques that employ in-air and depth pictures as the input to synthesise underwater images. Although our approach does not have this restriction, all of the assessed images are simulated at the size of 640*480 due to the WaterGAN picture size restriction.
The camera-light setups are first set up with two artificial spotlights that are 1 m distant from the camera on the left and right sides, both tilted forty-five degrees towards the centre of the picture, in order to create realistic deep sea photographs near to the images.

Thomas Braunl, Adrian Boeing,The University of Western Australia [8] The simulation programme is made to cater to a wide range of users with various requirements, including the user interface's layout, abstraction levels, and the complexity of physical and sensor models. As a result, the creation of a simulation tool that is as expandable and versatile as feasible serves as the software's primary design objective. Calculating the positions, orientations, forces, torques, and velocities of all entities and joints in the simulation is the responsibility of the underlying low-level physical simulation library. The higher-level physics abstraction layer (PAL) is only necessary to simulate motors and sensors because the low-level physical simulation library handles the majority of the physical computations. Custom plug-ins can be added to the current library using the PAL, replacing or enhancing the current implementations of sensor and motor models.

RadimHercik , Radek Byrtus , Rene Jaros and Jiri Koziorek [9] In this paper Individual instructions, such as logic functions, move functions, docking functions, and others, were used to develop the programme for MiR mobile robots. Then, these tasks were classified as "missions" by grouping them together. The robot's mission was therefore a collection of instructions that specified what it should execute during a certain mission. Individual orders were carried out progressively during the operation, and if the robot had already completed all of them, the mission was declared complete. A web interface was used to launch tasks to the robot, and further missions may be added to the queue for the robot to accomplish. The mobile robot waited for the assignment to be issued if it had no missions waiting in the queue or performing any active missions.

PROPOSED METHODOLOGY

Through unique packages, the Gazebo simulator may establish a direct connection with the ROS. The interfaces required for simulating a Gazebo robot with ROS messages, services, and dynamic reconfiguration are provided by these packages. Identifying the project process is the first step in designing an AUV. The AUV's design is the primary focus of the first stage. The evolution of the mechanical structure is the first of two parts that may be used to explain the following stages. The intended and anticipated AUV are therefore drawn and animated using computer-aided tools such as Solid-Works.

World Plugin For Hydrodynamics

The Gazebo simulator subscribes to a ROS topic that gives the water current's intensity and direction while accounting for buoyancy direction and water velocity (vc). As a result, the current may be changed as needed. Naturally, buoyancy moves in the opposite direction from gravity and disappears when a vehicle rises above the water's surface. An included Gazebo world file is used to control this plugin.

Model plugin for thruster control

Thrusters are also defined in the URDF. The greatest effort is put out for each thruster, and the mapping between the thrust direction and the body frame is also done:
Thus, thruster saturation may be simulated and handled in the control law. For the time being, it is assumed that each thruster is attached to the same connection, referred to as the base link in the example. The model plugin reads this tag to provide low-level control. We can observe from the provided example that the Gazebo simulator will pay attention to joint efforts on the joint command subject as well as thruster forces on the body command topic. These attempts can be produced by a high-level force controller or, more frequently, by the simulator's built-in low-level controller. We now go into depth about how this controller was implemented.

Low-level PID control

In this section, we go through the AUV simulator's PID controller. Designing an AUV advanced force controller is particularly laborious due to the many uncertainties in the dynamic model. Typically, a low-level PID controller guarantees that the specified setpoints are obeyed while a high-level controller provides position or velocity setpoints for the body and the arm.

General design:-

The global PID controller, which is implemented as a group of parallel or cascaded controllers depending on the user's preference, is handled by a separate ROS node. The two configurations (cascaded and parallel) are described in depth. The intended location and speed of a joint value or a body in a linear or angular direction are represented by p and v, respectively. P and V stand for the estimate of location and velocity from sensors or an estimation technique, respectively. Both configurations have the identical velocity PID, but the position PID outputs the effort in parallel mode and the velocity set-point in cascaded mode. Real-time gain tweaking is possible with Dynamic Reconfigure5 and is stored in a configuration file. Position and velocity control are established individually for the body and the arm, and are switched between using a ROS service.

Anti-windup and body thrusters mapping

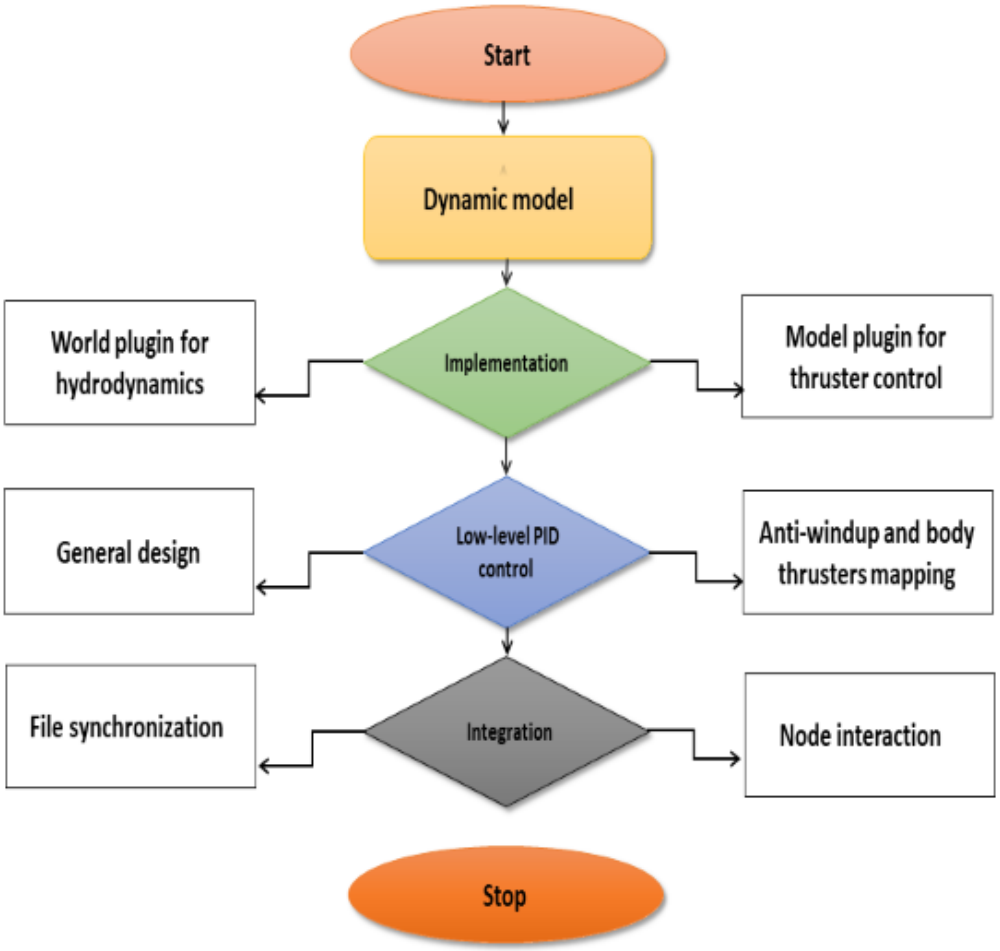
Each PID's anti-windup inner loops employ the highest speeds and efforts specified in the URDF. In order to determine the real controllable directions and the maximum effort associated with them, the maximum thruster efforts are specifically mapped to the body directions. For uncontrollable directions (usually roll, but occasionally also pitch and sway), no PID are instanced. While the AUV's velocity setpoint is specified in the vehicle frame, its location setpoint is given in the world frame. The various PID outputs show what is intended to be applied to the AUV body.

Integration

This section investigates the interconnections between the PID controller, the UWSim visualisation, and other potential higher-level controllers and the I-AUV simulation in Gazebo. As it is important to synchronise the Gazebo and UWSim description files, the file integration is first addressed. We then reveal and describe a node-based graph example.

File synchronization

Both Gazebo and UWSim use different methods to define the global landscape, extra moving objects, and the robot model. The URDF used to describe the robots is often simpler than the Gazebo ones since UWSim is merely a kinematic simulator as opposed to the latter, which contains all the inertial and collision-related data.. In UWSim, on the other hand, the scene file used to represent the whole world setup comprises all information about the simulation under consideration, however when using Gazebo, the same data is divided into different files. The synchronisation script that assists in creating lower-information files from higher ones is briefly discussed in this section. The UWSim launch file, which includes the UWSim scene file, which explains the whole setup, serves as the starting point. This makes it possible to retrieve the list of robots' URDF files, along with information about their beginning positions and the terrain's 3D mesh. The URDF files that are utilised by Gazebo may already exist or they may need to be created from xacro files that already exist. The UWSim scene file contains all beginning locations, making it possible to create a Gazebo model spawner and guarantee that the same items will be present in the same location in both UWSim and Gazebo. Finally, the file path conventions used by UWSim and Gazebo currently differ. Because of this, while creating or replacing missing files, the paths are modified instantly. With this kind of synchronisation, an experiment may be run initially solely in UWSim with kinematic control for ease of use and debugging, and subsequently with the dynamic simulator and low-level PID loops to have a realistic behaviour.



Implementation

The project implementation is divided in mainly three parts:-
•Creating the vehicle for the simulation and setting up the environment.
•Setting up the sensors for the sensor fusion algorithms.
•Creation of the autonomous path planning algorithm.

Step 1:- Vehicle modeling and simulation setup:-
• Vehicle setup:-
Actuators and vehicles are modeled separately in the robot description and should also have separate mesh file.
• Simulation setup:-
For the environment simulation setup we run a Gazebo world with Bullet physics environment.

Step 2:-Setting up of Sensors and Sensor Fusion :-
• Doppler velocity Log (DVL):-
An acoustic sensor called a Doppler Velocity Log (DVL) calculates velocity in relation to the ocean floor..

• Inertial Measurement Unit(IMU):-
A body's specific force and angular rate are measured and reported by an inertial measurement unit (IMU), an electrical instrument..

•Ultra-short baseline acoustic positioning system(USBL):-
Underwater acoustic location is accomplished using USBL (ultra-short baseline)..

Underwater Camera:-
The underwater camera is not directly used in the sensor fusion algorithm, rather it is used in the detection of underwater reefs and wreckages.

•Sensor-Fusion algorithm: -
To fuse the sensors together, Extended Kalman Filter was used which is a part of the ROS robot-localization package.

Step 3:-Path planning algorithm :-
• Path Segments:-
It is used to identify an SR path in an SR-MPLS network.
• Line Segments:-
Simply connecting two locatioin space, the line segment enables the calculation of all points along the line..
• Helical segment:-
It is possible to parametrize the helical curve segment to provide waypoints that, when interpolated, form a helical route..

EXPECTED OUTCOME

An applied project, we have taken lot of physical constraints in mind. Hence this can be implemented as a robot in the real world very easily.
The robot can be a priority list upon which it should hover, as of now it treats all underwater objects with same priority which makes it a bit slow since it has to give the location of unwanted objects.

CONCLUSION

A dynamic simulator for intervention autonomous under water Robot. Both Gazebo's dynamic simulation capabilities and UWSim's realistic underwater rendering were usable through the ROS framework.. The considered modeling is focused on the main effects due to hydrodynamic forces, that are drag and buoyancy. It is feasible to make improvements by accounting for the additional inertia and Coriolis, but it is challenging to exactly measure these effects.. That is why all uncertainties are here considered in the damping coefficients d(v). The buoyancy and damping properties of each robot connection and other moving objects, like the black box shown in the example, may be defined using additional URDF tags.. The The experiment was aimed to demonstrate that genuine challenges like thruster or joint effort limits, collisions, and uncertainty on the setpoint following can be reproduced using dynamic simulation.. Additional URDF tags may be used to specify the buoyancy and damping characteristics of each robot link as well as those of other moving objects, such as the black box in the example.. The proposed simulation framework can thus be used to validate new algorithms for sensor-based control or state estimation.

REFERENCE

1. EichhornM.; Ament, C.; Jacobi, M.; Pfuetzenreuter, T.; Karimanzira, D.; Bley, K.; Boer, M.; Wehde, H.modularauv system with integrated real time water quality analysis. Sensors 2018, 18, 1837.
2. Pino Diez, R.; Priore Moreno, P.; Puente Garcia, F.J.; Gomez Gomez Quesada, M.I.; Garcia Fernandez, N.; RosilloCamblor, R.; Ponte Blanco, B.organizational engineering in industry 4.0. Book of abstracts. 2019.
3. BnbnbnbGafurov, Salimzhan A., and Evgeniy V. Klochkov. "Autonomous Unmanned Underwater Vehicles Development Tendencies." Procedia Engineering (2015): 106.
4. Casalino, G., Zereik, E., Simetti, E., Torelli, S., Sperinde, A., Turetta, A.: Agility for underwater floating manipulation: Task & subsystem priority based control strat- egy. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. pp. 1772–1779. Vilamoura, Portugal (September 2012).
5. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., von Stryk, O.: Comprehen- sive simulation of quadrotor uavs using ros and gazebo. In: Simulation, Modeling, and Programming for Autonomous Robots, pp. 400–411. Springer (2012).
6. Pietrzik, S., and B. Chandrasekaran. "Setting up and Using ROS- Kinetic and Gazebo for Educational Robotic Projects and Learning." In Journal of Physics: Conference Series, vol. 1207, no. 1, p. 012019. IOP Publishing, 2019.
7. Guirguis, Silvana, Mark Gergis, Catherine M. Elias, Omar M. Shehata, and Slim Abdennadher. "ROS-based Model Predictive Trajectory Tracking Control Architecture using LiDAR-Based Mapping and Hybrid A* Planning." In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2750-2756. IEEE, 2019.