Hybrid Underwater Robot System Based on ROS *

Abdou Yahouza M. Sani
School of Mechanical and Automotive Engineering
Shanghai University of Engineering Science
Shanghai China
mahamanesania@ymail.com

Tao He†
School of Mechanical and Automotive Engineering
Shanghai University of Engineering Science
Shanghai China
hetao@sues.edu.cn

Wenlong Zhao
School of Mechanical and Automotive Engineering
Shanghai University of Engineering Science
Shanghai China
zwl1664@163.com

TingTing Yao
School of Mechanical and Automotive Engineering
Shanghai University of Engineering Science
Shanghai China
303058446@qq.com

ABSTRACT

Underwater Robots play an important role in a number of shallow and deep-water missions. In recent years, Underwater Robots system design has been an active field of engineering researches.

This paper proposes new system design for underwater vehicles combining the best features of both ROV (remotely operated vehicle) and AUV (autonomous underwater vehicle) technologies. The system based on ROS platform and Ardupilot software is equipped with two types of navigation system setup (autonomous navigation and teleoperation navigation) capable of realizing precise motion control, navigating while performing SLAM (Simultaneous Localization and Mapping) path-planning and obstacles avoidance.

CCS CONCEPTS

• General and reference~Empirical studies • Computer systems organization~Robotic autonomy

KEYWORDS

ROV, AUV, ROS, SLAM, Pixhawk, Ardupilot, Underwater Robot

1 Introduction

Covering 70 percent of the Earth's surface, the oceans remain largely unexplored from a subsea perspective, making research and data collection challenging. Underwater robotics is by far the best option to explore the water bodies. The recent advances in underwater robotics are opening opportunities for the researchers and industries [1], creating the burden of needing two separate types of vehicles and essentially two separate crews for different exploration and inspection missions. For one thing, the remotely operated vehicles (ROVs) are tethered and require for the presence of human operators. Their extensive use is currently limited to a few applications such as inspection and evaluation of submersed structures, transportation and assembly of underwater equipment, ships rescue, geotechnical and environmental data gathering, dumps or toxic waste location, marine archeology because of very high operational costs, operator fatigue, safety issues [2] and they are presented with a significant barrier to both smaller operational radius and efficiency; for another thing, the need for autonomy in robots and vehicles however, became more and more a prevalent issue in many situations and environments worldwide. The recent advances in underwater robotic technologies has eventually led to fully autonomous vehicles [3]. The autonomous underwater vehicles (AUVs) operate autonomously, they can move independently, carrying out missions for hours, or in some instances, days. While being untethered enables AUVs to have a greater operational radius, they can only maneuver in certain directions and change depths. To date, there has been little effort to prove the ability of AUVs to survive under water for long periods of time.

However, the evolving demand for underwater exploration and inspection missions requires a multi-mission vehicle that offers users both automation and control. The solution that best addresses these needs is a hybrid AUV/ROV, an underwater vehicle that is changing the way underwater surveys are performed. The hybrid underwater vehicles utilizing both ROV and AUV technologies, they have not only a digital framework for autonomy, but also offer to users the ability to quickly take control. It is a multi-mission vehicle that is poised to be a game-changer for the underwater robotics industry.

In this paper, we propose a new hybrid AUV/ROV system design based on ROS and Ardupilot, with two types of navigation system setup (autonomous navigation and teleoperation navigation). The main Objective behind this system concept is to provide a test-bed to our University for studying advanced control and underwater navigation. The second goal of this project is to implement a prototype AUV/ROV hybrid system for inspection of submarine pipelines and power cables of oil platforms.

The paper firstly, presents the general concepts and related work on mobile robot autonomous navigation. Section 3 explains the hybrid robot system functionality and capabilities follow by

†Corresponding author: Tao He. Postal address: 333 Longteng Road, Songjiang District, Shanghai University of Engineering Science

RICAI '19, September 20–22, 2019, Shanghai, China

ACM ISBN 978-1-4503-7298-5/19/09…$15.00
https://doi.org/10.1145/3366194.3366264

navigation system setup. The section 4 explains the hybrid AUV/ROV system framework overview, the hardware and software systems design follow by the software-hardware systems interaction. And at the end, the paper presents the study conclusion and future perspectives.

2 General Concept of Mobile Robot Autonomous Navigation

Autonomous navigation technology integrated in a mobile robot, makes it intelligent machine. In order a mobile robot to achieve fully autonomous navigation, it must be capable of determining its own position and orientation regarding its surrounding environment. The essential problem is known as localization. The only means by which the vehicle can navigate successfully is by obtaining real-time data from its immediate surrounding environment. It attains the necessary information via its on-board sensors and systems.

2.1 Localization

The localization is the problem of determining the pose of a robot relative to a given map of the environment often called position estimation. Several techniques such as AMCL (Adaptive Monte Carlo localization) a probabilistic localization, were developed. However, in order to perform position estimation suitable for unstructured or Global Positioning System (GPS) denied environment such as underwater environment, the earliest method used is visual odometry technique, based on computer vision.

This paper focuses only vision-based pose estimation, visual odometry (VO) is the process of estimating a robot's 3D pose using only visual images [4]. Estimating a robot's 3D motion from sensor data typically consists of estimating its relative motion at each time step by aligning successive RGB-D frames of the Microsoft Kinect sensor. Visual odometry output is often fused with Inertial Measurement Units (IMU) data or Visual Inertial odometry (VIO), in an Extended Kalman Filter. The filter computes estimates of both the position and velocity, which are used by the PID controller for better control, navigation and motion planning. However, since the robot's localization must be with respect to its immediate surrounding environment, it is necessary to acquire the prior knowledge of the environment.

2.2 Mapping

The map of an environment is seen as the global pose or the environment representation where the robot's local position and the obstacles location are estimated. It is often used for navigation and it is one of the key problems in robot perception and is strongly tied to localization, since a robot needs to know its location over time to build a consistent map.

Recently, an octree-based 3D mapping approach is proposed by Hornung et al [5] where each node in an octree represents a cubic volume of space usually called a voxel. This volume is recursively subdivided into eight sub-volumes until a given minimum voxel size is reached. Hence, the minimum voxel size determines the resolution of the octree. Octomap is an open-source framework that performs 3D occupancy mapping based on octrees. Essentially, Octomap performs the probabilistic occupancy estimation at each observed voxel to represent not only the occupied space, but also the free and unknown areas.

2.2 Underwater SLAM

Since the localization and mapping are strongly tied together, one cannot exist without another, this seems to be a chicken-and-egg problem, which is often referred to SLAM problem. SLAM can be described as the problem, where a robot situated at an unknown location creates a map and localizes itself in that map. However, to perform SLAM in the underwater environment is not a simple task but not impossible. The underwater environment is observed by imaging sonar or other image sensors. Then the collected image data are extracted and the approximate coordinates of the features are estimated. Finally, the estimated feature location is matched with the observed map feature, which is called data association. In this process, the matching features are first estimated and fused, and then the estimated pose and feature position of the robot are corrected respectively. Mismatched features can be added to the map as new features for subsequent iterations or deleted directly from the map. The general framework of underwater water is seen in Figure 1.

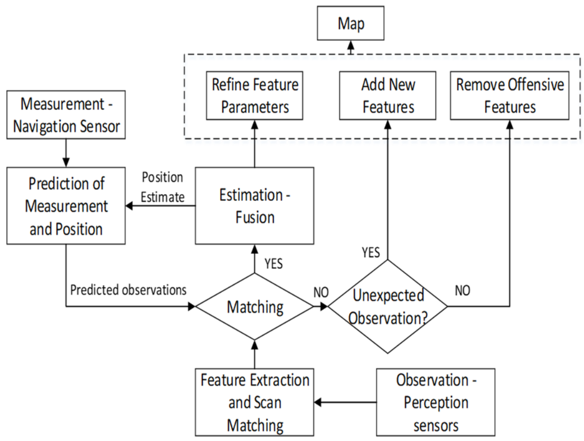

Figure 1: Framework of Underwater SLAM

Several SLAM methods were developed to be applied on mobile robot.

In this work we are only interested in Visual SLAM, which refers to the problem of vision-based simultaneous localization and mapping. Over the past few years many Visual SLAM algorithms were developed, such as RGBD SLAM [6] proposed by P. Henry, Stereo Parallel Tracking and Mapping Strategy (S-PTAM) coined by Pire et al. [7] and recently ORB-SLAM2 [8] of Mur-Artal et al. In this type of SLAM system, localization and mapping are performed using image information only. It has the main goal of estimating the camera trajectory while reconstructing the environment.

Since visual odometry (VO) can be used as a building block of SLAM, therefore, the Visual SLAM approach consists of combining visual odometry methods for finding the relative transformation between frames, and SLAM methods for global optimization and loop closure.

2.3 Path Planning

Even though the robot can acquire both localization and the environment map, the requirement of autonomous navigation is by far from being fulfilled. Therefore, to achieve fully autonomous navigation, the mobile robot must have the capability of planning a path between the robot position and a given goal coordinate in the map structure and still be able to perform obstacles avoidance.

Researchers have tried to solve the Path-planning problem by developing and implementing many algorithms. Burgard et al. [9] described a method for planning a 3D path by using the D* lite algorithm on autonomous quad-copter for indoor use. Hebecker T, Buchholz R, and Ortmeier F [10] applied the wavefront algorithm as the local path planning algorithm to calculate collision-free paths within the field of view of a UAV’s obstacle detection sensor.

3 System Functionality and Navigation Setup

3.1 Hybrid System Functionality

According to the project's goal, the AUV/ROV hybrid system function should achieve four main functions:

1. An underwater robot with four degrees-of-freedom actuation capability, with fully waterproof parts.
2. The system should be able to perform the Simultaneous Localization and Mapping (SLAM) and even use it to navigate autonomously.
3. The AUV/ROV hybrid has the capability to navigate from a point A to a set target point B. All reaching the target position, by avoiding obstacles on the way.
4. This robot is originally an AUV capable of autonomous navigation during mission but continuously running auto-system diagnostic, if any system failure is found it automatically alerts the user and shifts to ROV mode (RC-control).

3.2 Hybrid System Navigation Setup

To manage the AUV/ROV hybrid navigation system, we use ROS platform and PX4 autopilot to set the navigation behavior in two different status: Autonomous navigation and Teleoperation mode. In the autonomous mode, the hybrid robot must use the data coming from onboard RGB-D sensor, IMU and Pressure sensor to estimate its own position (local pose estimation) , build a dense 3D model of the environment from octrees (global localization) using Octomap and use the ORB-SLAM2 algorithm to perform V-SLAM (Simultaneous Localization and Mapping) and finally computes the 3DVFH+ algorithm for trajectories planning and obstacles avoidance through the environment.

In the teleoperation mode the robot navigation totally depends on the inputs from the remote controller (RC) and vision sensor.

4 Hybrid System Framework Hardware and Software

4.1 Hardware System

The Figure 2 shows the hardware overview of the new system proposed. On the surface we have a host computer running ROS and Ground Control Station for sending instructions to NVIDIA Jetson TX2 the lower computer (on-board main computer) through the umbilical cord cable.

The NVIDIA Jetson TX2 single board computer (SBC) also running ROS, is connected along with the Kinect sensor for SLAM and another monocular camera for streaming HD video purpose during manually navigation. The GPS sensor can be used to provide a global position to the vehicle on the water surface. In addition, during autonomous navigation, the companion computer providing external vision position estimates to Pixhawk, the main controller for better stability and motion control.

Finally, the system is equipped with a power system unit for energy management and supplying power to the Microsoft Kinect, companion computer, Pixhawk controller and the thrusters.

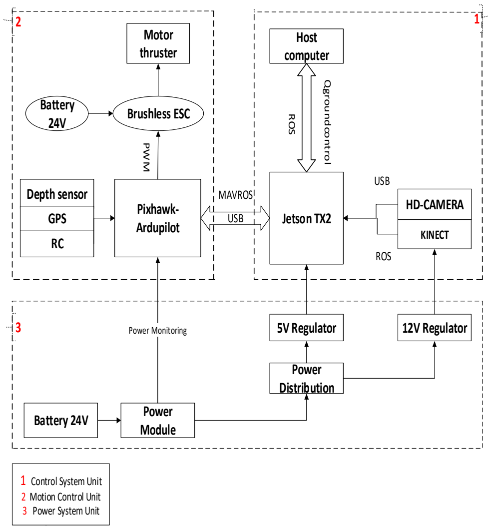

Figure 2: Figure Hardware System Overview

1.1.1 Control System Unit. The control unit is the most important part of the system. It acts as the brain of the AUV/ROV hybrid and is responsible for the autonomy of the vehicle. The main role of the control system is to assure the complete and absolute control of the vehicle. The term control here actually has a broad sense, including but not limited to (i) motion control: the low-level system control; (ii) mission control: the high-level behavioral control which is usually predefined and triggered by sensor measurement; (iii) power management. Our system control unit is carrying two major computers, the host computer on the surface and the companion computer on board connected via tether cable. The host computer serves as control station from which all the high-level commands are sent to the robot and received feedbacks. The companion computer on-board has the capability to process the commands received from the host computer and all the information and data coming from different subsystems on board. Finally, it exchanges the feedbacks of the robot states to the control station.

1.1.2 Motion Control Unit. The Motion Control Unit is the second important and large subsystem in our new system approach. It is responsible for the motion and attitude control. Since controlling the high range motions of an underwater vehicle can be problematic, it is a must to determine the number of degrees of freedom and how to control each one of them, in order to keep the vehicle stable. According to the many published researches, the underwater robot can have at most six degrees of freedom (DOFs). However, our system capability is limited to controlling four degrees of freedom, which are independent motion variables known as 'surge', 'heave', 'pitch', and 'yaw'. The motion control unit of the AUV/ROV hybrid aims to regulate the variables to the desired values, focusing on inputs/outputs of the vehicle and the closed-loop properties based on the navigation stack. Most underwater robots adapt the PID (Proportion-Integral-Derivative) control technique. The PID control is universally applied to underwater robots due to its good robustness and easy implementation. Acceleration feedback technique which enables the inertia shaping to be incorporated into the conventional PID control design in an attempt to get better control stability. The motion control Unit of the new hybrid system is composed of the Pixhawk PX4 controller, and sensors. Pixhawk PX4 controller is high-quality and low-cost autopilot hardware, with a PID control algorithm implemented that aims not only to monitor attitude and altitude of the robot, but also to control the thrusters for propulsion purpose. It achieves this low-level control, with the aid of other external sensors such as depth sensor, GPS sensor (on the water surface only) and RGB-D Kinect.

1.1.3 Power System Unit. Most underwater robots in use today are powered by rechargeable batteries (lithium ion, lithium polymer, nickel metal hydride etc.), and are implemented with some form of Battery Management System. The power system unit aims at optimally distributing the onboard power, ensuring the batteries discharged evenly, and allowing onboard power to be disconnected from all circuits. The thrusters and electronic components are powered by two banks of 24V, 2600mAh. These batteries allow the vehicle to operate continuously at maximum specific period of time. They are housed in a waterproofed frame for easy removal from the hull. One battery is connected to the power module and power distribution board and another connected to the brushless ESC of the thrusters.

1.1.4 Propulsion Unit. There are couple of propulsion techniques for underwater robots. Some of them use brushed or brush-less electric motor, gearbox, Lip seal, and a propeller which may be surrounded by a nozzle. All of these parts are embedded in the AUV/ROV construction for propulsion purpose. The thrust produced by the motors is equal to the friction or drag of the vehicle.

$$Thrust = Drag = \frac{1}{2}\rho s^2 A c_D \quad (1)$$

Where $\boldsymbol{\rho}$ is the water density, $\boldsymbol{s}$ is the constant speed, A is the effective surface area and $\boldsymbol{c_D}$ is the drag coefficient Power consumption for the propulsion system increases dramatically as the speed of the vehicle increases because the thrust power is equal to the product of the thrust and the speed, meaning thrust power is a function of speed cubed.

$$Thrust\ Power = Thrust \times s = \frac{1}{2}\rho s^3 A c_D \quad (2)$$

Therefore, the best choice for thrusters are the brushless DC motors to achieve minimum friction and less power consumption. The propulsion system of our ROV/AUV hybrid is composed of four DC motors. That provide motions in 4DOF. The two horizontal motors control surge and yaw while the two vertical motors control heave and pitch. They are powered by an independent battery of 24 V.

4.2 Software System Architecture

The Simultaneous Localization and Mapping (SLAM), obstacles avoidance and path-planning all running in ROS. The low-level attitude and motion controllers run on the Pixhawk. The Figure 3 explains the implementation of the software system.

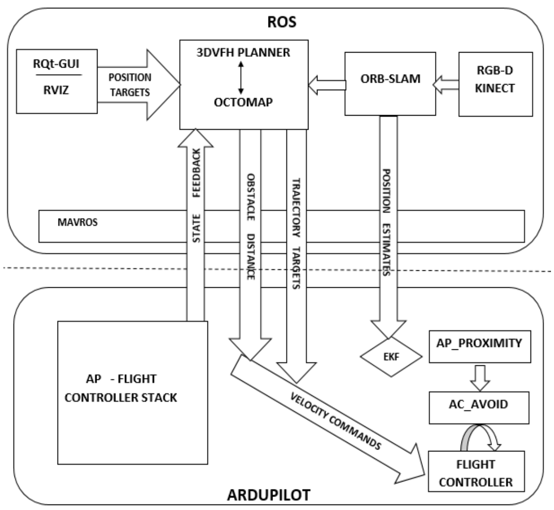

Figure 3: Software System Architecture

RVIZ/RQT-GUI (visualizer tools) sets the position targets. The local planner node generates 3D map Octomap acquired from the RGB-D KINECT points cloud and runs the whole planner algorithm to publish the calculated setpoints including obstacles position. Those are then sent to the autopilot over the Mavros node.

The RGB-D KINECT also provides RGB-D frames to feed ORB-SLAM2. The local position estimates computed by ORB-SLAM2 is fused with IMU data and the barometer data in the AP (Ardupilot)'s Extended Kalman Filter (EKF) to increase control stability and pose estimation accuracy. The Ardupilot (AP_stack) sends the current state to the local planner which returns the velocity commands to flight controller. The Mavros stack acts as the fabric for translating between ROS and MAVLink. The ROS path handler node handles communications between the global planner node and Mavros.

4.2 System Software Hardware Interaction

The software system is the combination of different platform system of ROS (robot operating system), QGround Control (QGC), and Ardupilot. The Figure 4 describes how the software interacts with the hardware. The QGround Control installed serves as the user interface for calibrating the sensors and manually operating the AUV/ROV hybrid.

The ROS platform provides the means of communication between different parts of the system (USER, ROS, Ardupilot and Sensors). ROS also provides tools such as hardware abstraction, device drivers, libraries, visualizers, message passing, package management, and more. The usage of ROS allows us to add any number of data sources and software libraries. Since we use ROS, we can take the advantages of various open-source stacks in our system such as Mavros stack, Octomap, OpenCV library, 3DVFH+ algorithm and ORB-SLAM2.

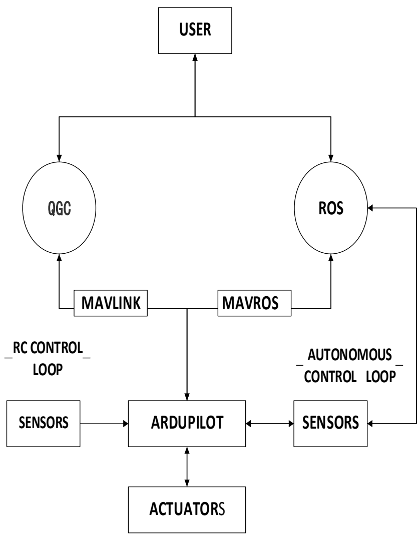

Figure 4: System Software-Hardware Interaction

5 Conclusion

The ROV/AUV hybrid system with dual navigation system setups implemented does not only provide the satisfaction and capability of exploring the depths of the world's bodies of water, but also the technological advancements like cloud computing and SLAM promote greater interactivity with the surrounding environment during missions both below and above the water. However, the system could suffer some time delay due to the fact that the companion computer communicates with the Pixhawk controller via ROS nodes. In addition, with many sensor features, the computation of data is very huge and the data association is very difficult, which makes the system highly cost-effective. In the future work, the implementation of the hybrid system on the real underwater robot will follow.

ACKNOWLEDGMENTS

This project is funded by National Key Research and Development Program of China (Grant No. 2018YFB1307900), Key Research Program of Shanghai Science and Technology Commission (Grant No. 16030501200), National Natural Science Foundation of China (Grant No. 51805314) and Shanghai University of Engineering and Science (Grant No. E3-0501-18-01002). The Robot Functional Materials Preparation Laboratory in Shanghai University of Engineering Science is also gratefully acknowledged.

REFERENCES

[1] Budiyano, A (2009). Advances in unmanned underwater vehicles technologies: Modeling, control and guidance perspectives. Indian Journal of Marine Sciences, 38(3), 282-295.

[2] S. M. Zanoli, G. Conte (2003). Remotely operated vehicle depth control, Control Engineering Practice, 11(4), 453-459.

[3] T. Hyakudome (2011). Design of Autonomous Underwater Vehicle, International Journal of Advanced Robotic Systems, 8(1), 131-139.

[4] Bellavia F, Fanfani M, Colombo C (2017). Selective visual odometry for accurate AUV localization[J]. Autonomous Robots, 41(1), 133-143.

[5] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard (2013). OctoMap: An Efficient Probabilistic 3D Mapping Framework based on Octrees. Autonomous Robots, 34(3),189–206.

[6] Henry P, Krainin M, Herbst E, et al (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments[J]. International Journal of Robotics Research, 31(5), 647-663.

[7] Pire, Taihú, Fischer T, Castro, Gastón, et al (2017). S-PTAM: Stereo Parallel Tracking and Mapping[J]. Robotics and Autonomous Systems, (93), 27-42.

[8] Raúl Mur-Artal and Juan D Tardós (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, 33(5), 1255-1262.

[9] S. G. G. G. W. Burgard (2012). A fully autonomous indoor quadrotor. IEEE Transaction on Robotics, 28(1), 90-100.

[10] Hebecker T, Buchholz R, Ortmeier F (2015). Model-Based Local Path Planning for UAVs. [J]. Journal of Intelligent & Robotic Systems, 78(1), 127-142.