UNIVERSIDADE DA CORUÑA

n

AUTONOMOUS AND MISSION BASE NAVIGATION FOR BLUE-ROV2 USING OPENCV

FRAN BELLAS

FÉLIX ORJALES

SARA GUERREIRO

SEMESTER PROJECT

BY

AKINMULEWO DANIEL

NWAFOR OGECHUKWU

SUSTAINABLE SHIP AND SHIPPING 4.0

HIGHER POLYTECHNIC UNIVERSITY COLLEGE

UNIVERSITY OF A CORUÑA

1.0. INTRODUCTION

Autonomous robotic systems are becoming more important in many industries. In the context of a marine area, these tasks cover water exploration to underwater mining and inspection, borderline guarding to search and rescue victim in water area.

The Blue Robotics BlueROV2 is an underwater drone made by Blue Robotics. The BlueROV2 features six powerful T200 thrusters and Basic ESCs, enabling it to travel at up to 1.5 meters per second. This also allows the ROV to gain high mobility in all directions. The BlueROV2 has the best thrust-to-weight ratio in its class to perform demanding tasks. It is ideal for operations in shallow to moderate waters, with a standard 100m depth rating and up to 300m tether lengths available. Various accessories are available, such as a gripper, sonar, and additional thrusters (up to 8 thrusters can be mounted at a time).

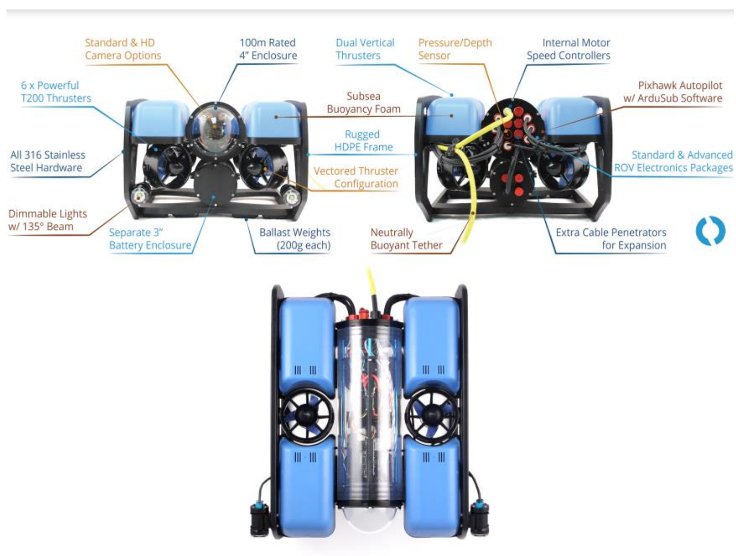

Figure 1: BlueROV2 Light Configuration

2.0. STRATEGIES FOR MISSION BASE NAVIGATION:

The BlueROV2 is equipped with a live camera feed and DVL sensor. The main idea for the mission-based navigation is to autonomously navigate the BlueROV2 by integrating a color detection algorithm that identify the color of an object using OpenCV in python. The coordinate of the identified object is used as input for the control system. The control system consists of a PID speed controller and ardupilot controller to navigate the BlueROV close to the identified object while performing the task.

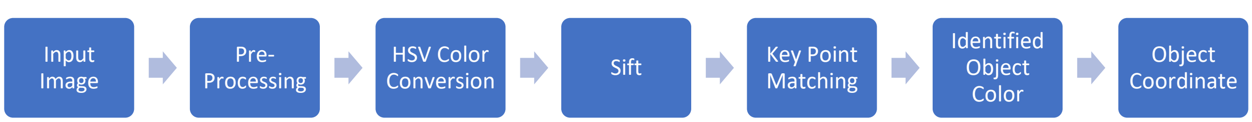

Figure 2: Object Detection Flowchart using OpenCV

For autonomous navigation, the following procedures are undertaken:

- **Search Operation and Path Planning**:
  The first task performed by the BlueROV2 is to conduct an initial scan of the environment using the camera. This is done to identify the color of objects within the surrounding. If objects are not detected, the BlueROV2 throttle backward and readjust the camera position by tilting camera position for a wider vision. After the coordinate of the object is identified, the planned trajectory is achieved by minimizing the error between the lateral position and the position of the identified object. This is done by computing the error from its current orientation to the coordinate of the identified object color and navigating using a proportional control based on the error.
  When a particular distance close to the object is achieved, the BlueROV2 throttles downward by maintaining a particular depth relative to the object center.
- **Navigation:**
  Navigation of the BlueROV2 is achieved by implementing a proportional, integrated speed controller based on the error orientation and the distance of the current to the coordinate of the object. The gain for the proportional and integral term are selected or tunned to minimize the errors.
- **Performing Task:**
  After maintaining a particular depth and distance from the object, the BlueROV2 re-aligns its center to the center of the object. After the alignment process, the BlueROV2 approaches the object and aims to grab the object by actuating the gripper. If the object target is missed, the BlueROV2 reverse back and tries to grab the object (In this case the lever arm). This operation is continuously performed until the object is grabbed. After grabbing the object, the BlueROV2 moves backward to pull, moves forward to push or roll to rotate the lever. The three different tasks are performed until an indicator light is displayed to show task completion for each lever.

3.0. LIMITATIONS, CURRENT CHALLENGES AND FUTURE IMPROVEMENTS:

During the testing and program implementation, some of the challenges encountered are listed below:

- It was not possible to determine the current depth of the BlueROV2 using the pressure sensor. The challenge to maintain a particular depth consumes more than 80% of the total time.
- Some features of the Autopilot (Ardupilot) controls are not working properly. Especially the case of the ALT_HOLD command that performs in a totally different form.
- Due to the unstable environment, the coordinate of the color identified (object detected) fluctuates repeatedly and affect the overall performance of the control system in terms of the calibrated thresholds.

For future development, the following areas will be considered:

- During the program implementation, color Inference, collision avoidance and path re-planning is not incorporated due to the limited time involved in the project.
- Path optimization is also not considered during program implementation. A better option is to implement an AI based and Machine learning route optimization technique that reduces time, smooth navigation and energy consumption.

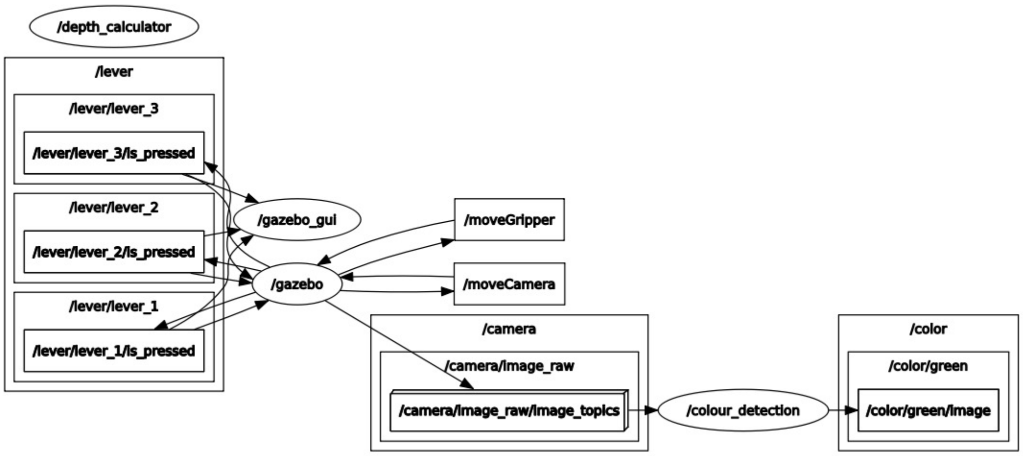

Figure 3: ROS Data Flow and Connection Model

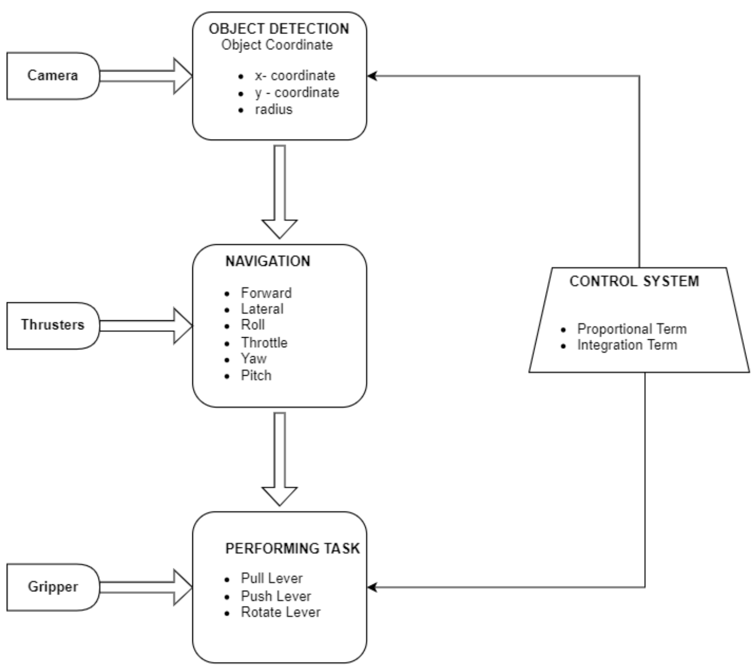

Figure 4: Control Architecture

4.0. RESULTS: