

Senior Design Final Report

Robosub

Version 1.0 - 05/13/2022

Team Members:

Carl Solli
Robin Romero
Christian Castillo
Leslie Araujo
Ashkan Aledavoud
Alan Chan
Edwin Tran
Bryan Sanchez
Jeanie Jeon
Daniel Valadez

Faculty Advisor:

Richard Cross

Table of Contents

1. Introduction:	3
1.1. Background:	3
1.2. Design Principles:	3
1.3. Design Benefits:	3
1.4. Achievements:	3
2. Related Technologies:	5
2.1. Existing Solutions:	5
2.2. Reused Products:	5
3. System architecture:	6
3.1. Overview:	6
3.1.1 Mission Planning:	6
3.1.2 Guidance Navigation Control:	6
3.1.3 Computer Vision:	6
3.1.4 Sensing and Actuation:	6
3.1.5 User Interface	7
3.2. Data Flow:	7
3.3. Implementation:	7
4. Conclusions:	10
4.1. Results:	10
4.2. Future:	10
5. References:	12

1. Introduction:

1.1. Background:

The RoboSub competition is an annual event held by RoboNation with the purpose of learning, cultivating, and enhancing the skills required to navigate an AUV through an underwater obstacle course. Participants will engage in the challenges of utilizing technology within ocean systems. The process of transforming a design prototype to a machine that operates in the real world is invaluable to both engineers and computer scientists.

An AUV is a fully autonomous underwater vehicle and is the machine that will implement the teams' work over the past year. California State University, Los Angeles will be competing against other universities in Maryland this year. The winner of the competition is the team whose AUV scores the most points.

In order to "see" and navigate through the course, the team utilized a number of techniques such as machine learning with software like YOLOv4 and OpenCV as well as implementing a state machine within the AUV.

1.2. Design Principles:

The design is a modular approach using ROS to separate our code into separate packages. We wanted our code to be in separate packages so each piece can be worked on separately and also reused in future designs for the Robosub. The communication that happens between the packages are with ROS Topics and well defined custom messages so the package knows exactly what form the data it receives will take. The sensors and controls are part of one package, we did overway splitting them to separate packages, but decided on keeping them in one package with different subfolders.

1.3. Design Benefits:

The benefits with the design approach we took is that it's easy to create new ROS packages if we want to add additional functions. The functionality of the Robosub is clearly separated into the packages keeping related code in the same place. It is easy to replace the packages we already have with other similar packages if needed at a later point.

1.4. Achievements:

From the beginning of the fall semester of 2021 to the spring semester of 2022, the RoboSub senior design team has developed a skeleton code for this year's AUV (Autonomous Underwater Vehicle). All the important parts of the AUV have a basic

skeleton code that can be used on the AUV itself, which, at the time of writing, has not yet been completed. Because of this, all parts of the code are only skeleton codes, but once the AUV is completed and operational, the code can be easily modified to make it work with the AUV.

Therefore, all parts of the code are only done as far as possible with an incomplete AUV. For example, we have successfully trained our machine learning, image recognition model, YOLOv4, to recognize important objects in the RoboSub competition; this is an important part of the AUV's computer vision, because the AUV must be able to recognize many different images. We have also done work on the first few states of the state machine, which is important when deciding what the AUV should do after something has been done. For the sensor part of the AUV, which is just as important as the image recognition, we have tested our code with dummy input from the sensors, and have confirmed that our code is able to receive and work with all the data. However, because the AUV is incomplete, we have not been able to test these things on the AUV itself. Finally, more details about all the parts of the AUV will be explained in the sections to come.

2. Related Technologies:

2.1. Existing Solutions:

We have inherited and looked into the previous year's Robosub team code which they have inherited from the year before and so on. The previous year's code lacks a lot of the documentation required for the amount of code that was created and because of this we've been working on interpreting the previous year's code and expanding upon it.

We decided to divide and approach the code within our subgroups that is code related to a certain subgroup that will then be documented and expanded by that group. With that we divided the inherited code by our subgroups that is mission planning, controls and sensors, computer vision, and website.

The mission planning group inherited the code that is related to the states that the AUV will be in. The code is to be interpreted by them, expanded, and lastly documented by the mission planning group. The code from last year's mission planning group has been expanded upon by creating new code to abide by this year's competition objectives.

The controls and sensors group inherited the code that is related to controlling the AUV and receiving data from the sensors itself. Most of the code that has been inherited from the sensors have only been slightly altered considering that the sensors that the AUV this year will be using the same sensors.

The computer vision group inherited the code that is related to training the AUV to recognize and determine objects that are surrounding it. The code is being expanded upon to coincide with the objectives for this year's competition.

The website group has inherited last year's code for the Robosub website. The code has been altered to now include images and members of this year's robosub team.

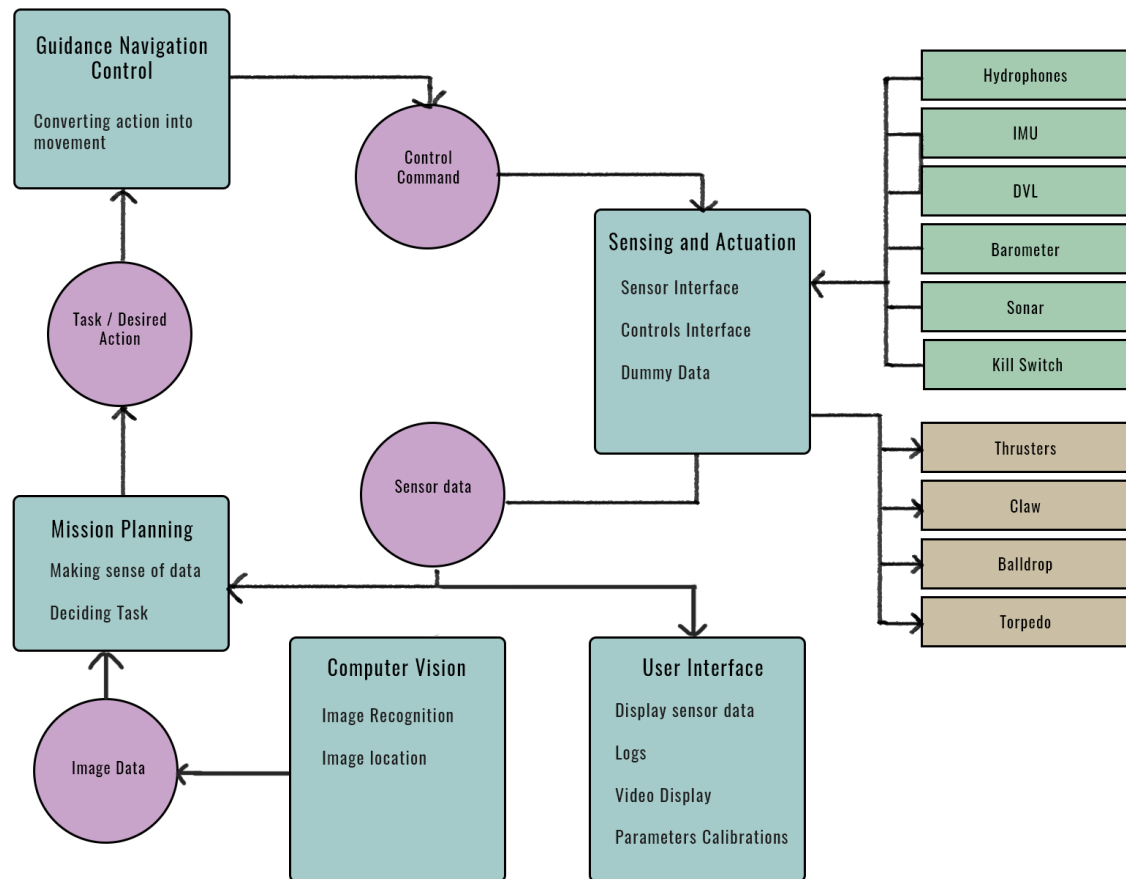
2.2. Reused Products:

Darknet is a repository in which a classification machine learning model named Yolo4 is used to classify objects within images and videos. The implementation for SMACH which is a ROS library that allows the creation of state machines for the robosub. Hardware to be used by the robosub which includes: VectorNav IMU, Teledyne Pathfinder TVL, Pressure Sensor, Ping Sonar, Hydrophones.

3. System architecture:

3.1. Overview:

Under you can see how our system is split into different packages and how they are communicating among each other.



3.1.1 Mission Planning:

In the mission planning we have a state machine that controls what task the Robosub will be doing and what actions to take to complete those tasks.

3.1.2 Guidance Navigation Control:

The guidance navigation control package helps the Robosub when it comes to maneuvering. It takes in commands from the mission planning and translates it into commands for the controls.

3.1.3 Computer Vision:

This package entails everything related to computer vision. It connects with the camera and translates what we see into data which it sends to the Mission Planning.

3.1.4 Sensing and Actuation:

All the code for sensors and controls are contained in this package. It has code for the Sonar, Barometer, IMU and Thrusters currently.

3.1.5 User Interface

This package is for the user to interface with the robosub. Currently it's used to display all the sensor data.

3.2. Data Flow:

Data is handled and shared amongst the different packages to help with the movement of the sub and completion of the tasks. Data is collected from the different sensors and sent to mission planning, where it is interpreted and used to determine the next instruction to be completed. Mission planning also receives image data from computer vision that it takes into account when forming the following instructions. Guidance navigation control uses instructions from mission planning to form commands that are sent to the different controls to move the sub and assist it in fulfilling the specified tasks. The user interface receives and displays information such as the data being collected by the sensors, as well as images being captured by the camera.

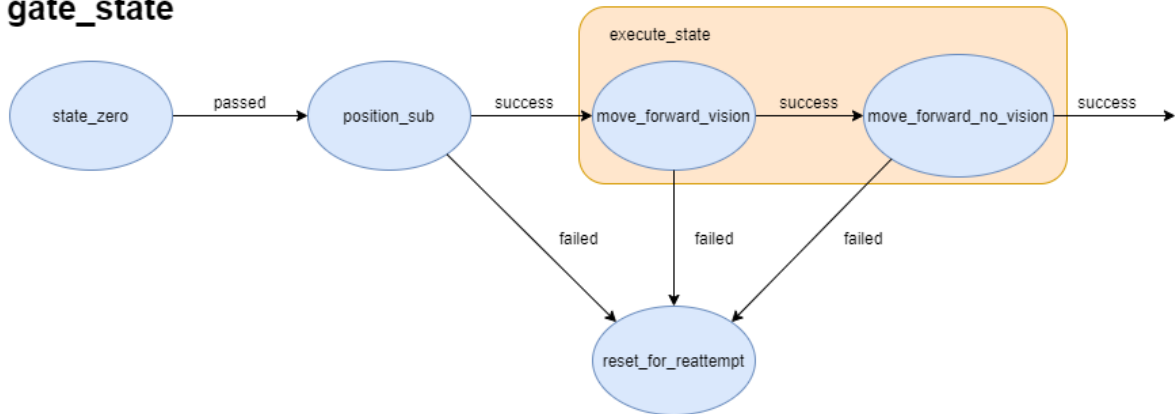
3.3. Implementation:

The project was split into four sections to allow for efficient development: Mission Planning, Controls & Sensors, Computer Vision, and User Interface. Each section plays a key role in presenting the progression of the project.

3.3.1. Mission Planning

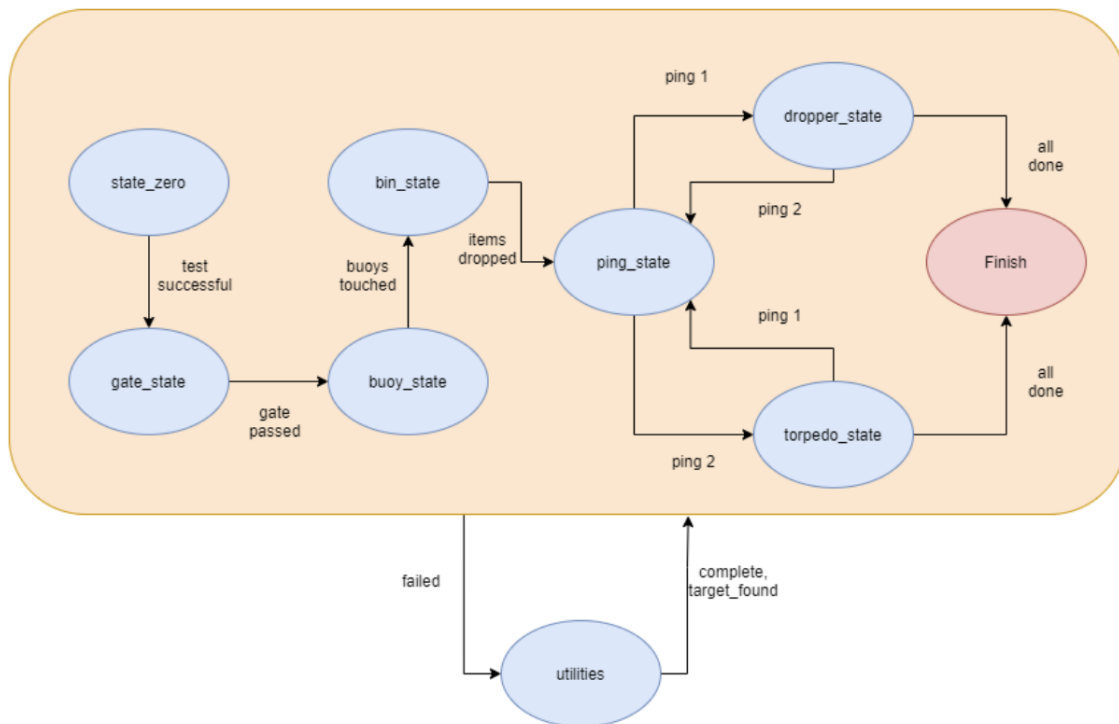
Mission Planning uses SMACH, which is a ROS library, to create state machines that allow the sub to perform different tasks. Mission planning continuously receives data from other systems, such as sensors and computer vision to determine the exact position of the sub at any given point and time and also sends commands to controls and actuation systems to maneuver the sub throughout the competition. SMACH states are sequential subroutines that are developed to perform a specialized task. This will usually lead to an execution state that triggers a transition to a new state. To use the AUV as an example, the gate task starts with the task of finding and recognizing the gate with computer vision data. The AUV then transitions to the tasks of moving toward the gate and going past it with thruster instructions provided by controls. Once the AUV has completed these tasks, it will transition to the next task given to it. These tasks also contain outcomes that transition the AUV to a state designed to reset the sub's positioning if something goes wrong. We also implemented SMACH Viewer, which is another ROS library, to easily view the possible transitions between states as well as the currently active state and the values of user data that is passed around between states.

gate_state



The figure below demonstrates the flowchart of all the states and the overall objectives that mission planning needs to complete during the competition

Flowchart



3.3.2. Controls & Sensors

The sensors subgroup's focus is to retrieve data from the various sensors on the AUV and send the data collected from the sensors to the mission planning subgroup to accurately determine the AUV's action and to the user interface to be displayed when

testing the AUV. The data from the sensors will constantly be retrieved and sent to the mission planning group until no further actions are needed for the AUV to be performed. The multiple sensors that the AUV will retrieve data from are doppler velocity log (DVL), pressure sensor, inertial measurement unit (IMU), sonar, and hydrophones. The type of data received from the sensors are direction from hydrophones, roll, pitch, and yaw from IMU and DVL, distance and confidence score from sonar, and depth from barometer. The data received will also be sent to the user interface to allow for testing and fine tuning for the movement aspect of the AUV. This will only be done however when testing the AUV.

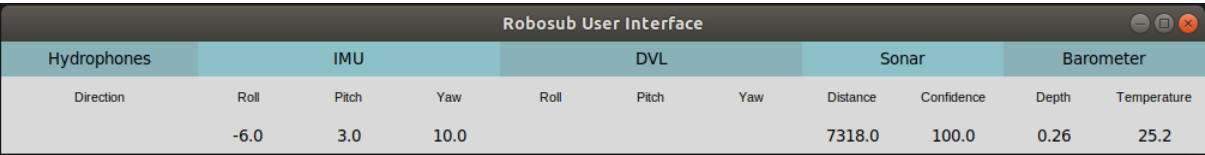
The controls subgroup's focus is to retrieve the commands sent from the mission planning subgroup and execute the action to the best of the AUV's abilities. The controls also focus on the movement of the AUV by changing the strength from the thrusters as well as the claw, ball drop, and torpedo aspect of the AUV for the competition objectives.

3.3.3. Computer Vision

Computer Vision uses Darknet/Yolov4 to detect and classify objects within a live feed or an image. Using a camera that is connected to the AUV, a trained model is then used to determine the priority of an object that is within the camera's view and will send data to the AUV to move towards the target object. To train the model to classify an object within an image two files are necessary, the first being the names of all the objects that are part of the obstacle course and labeled images of the object. The images will need a complementary txt file with coordinates of the object within the image. These coordinates are needed to train the machine learning model to recognize a similar object within another image. The model during the training stage creates weights to determine how accurate the model can classify an object within an image. The model will iterate as many times dictated by the user to update the weights to increase the accuracy of the model.

3.3.4. User Interface

A user interface was designed specifically for this project to visually present the sensor data. We used Python and Tkinter to implement our user interface. You can see in the image under how the sensor data is displayed. The User Interface is subscribed to the ROS Topics that publish information about the sensors.



Robosub User Interface										
Hydrophones	IMU			DVL			Sonar		Barometer	
Direction	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Distance	Confidence	Depth	Temperature
	-6.0	3.0	10.0				7318.0	100.0	0.26	25.2

4. Conclusions:

4.1. Results:

This year's team had a code base from previous teams to build upon. We have added more code onto the existing base.

The State Machine that the RoboSub uses to make decisions has State Zero and Gate State. This allows the RoboSub to begin the competition. State Zero initializes the RoboSub and makes sure all the sensors are working. Gate State allows the RoboSub to attempt the first task of the RoboSub competition - to pick a side and pass through a gate.

We also have a Graphical User Interface that will allow the RoboSub to communicate with a laptop on the surface that will display information.

The team has also put together a computer for training the image recognition of the competition targets. This would give us an advantage over using cloud computing such as Google Colab by allowing us to use an Nvidia GPU with CUDA cores. This allows us to use configurations for the training software that will give us faster training times and not have to worry about the limitations of Colab.

The team has also connected some of the sensors to the Arduino Mega. This will allow the RoboSub to take in data and use it to make decisions about how to complete a task.

4.2. Future:

There is still room for improvement with our AUV in which the group is planning on continuing to work on the project until the competition begins at the end of July. As section 4.1 states, we have added more to what the previous senior design group has originally left for us by giving the AUV a set of behaviors/tasks with Mission Planning and being able to train and recognize images with the AUV's camera using Computer Vision. The Robosub website also got major updates in which it now includes information about the 2022's Senior Design team (needs to be changed for future groups) and information about the competition, the Robosub website can be found here <https://www.robosubla.com/>.

We are proud to represent California State University, Los Angeles and to be a part of this yearly competition that is taking place in Maryland this year. There is still plenty that needs to be done to improve our AUV and we are excited to know that this project

will be carried forward by following senior design groups at California State University, Los Angeles. The following are improvements that will be attempted by this years Senior Design group but is recommended for the next year's team to work on:

- Changing the information in the website to the new members information and images.
- Testing the AUV to its full potential once the missing thrusters and claw are added to the AUV.
- Testing the system in the water to make sure it can maneuver properly.
- Recognizing the desired objects underwater with our AUV's camera.

5. References:

RoboSub 2022 Handbook:

https://robonation.org/app/uploads/sites/4/2022/05/2022-RoboSub_Team-Handbook_v2.0.pdf

ROS: <http://wiki.ros.org/>

SMACH: <https://wiki.ros.org/smach?distro=melodic>

Smach Viewer: http://wiki.ros.org/smach_viewer

