Sonar-Assisted Autonomous Navigation Implementation on a BlueRov

Ayesha ASLAM, Worachit KETRUNGSRI, Yosef GUEVARA.

Marine Mechatronics
June 3, 2023

Contents

1 Introduction

This report implemented an autonomous behavior, BlueROV capable of navigating in a controlled environment. The goal is to enable the robot to move forward while maintaining a fixed heading, come to a stop when it detects an obstacle in front of it, and subsequently search for a free path to continue its forward movement. The robot will utilize an Inertial Measurement Unit (IMU) for yaw angle measurement and a frontal echosounder pinger to determine the distance to the obstacle.

The behavior design is decomposed into three main steps. In the first step, the robot will utilize a PID control system to maintain a fixed heading while moving forward. The IMU will provide orientation feedback, which will be converted into Euler angles, and the yaw angle will be used for the heading control. Prior to triggering the autonomous heading control, the user will pilot the robot to position it with its longitudinal axis perpendicular to the obstacle.

The second step involves bringing the robot to a stop in front of the obstacle using the feedback from the pinger. A PID control system will be employed to maintain a fixed distance from the obstacle. To improve the accuracy of the distance measurement, an alpha beta filter will be used to estimate the derivative of the measured distance.

In the third step, the robot will rotate in place to survey its surroundings and search for a free path. It will record distance and heading measurements within an exploration angle interval, typically between -45 and 45 degrees. Once a free path is identified, the robot will transition back to the first step, moving forward while maintaining the heading found for the free path.

By implementing this autonomous behavior, the robot will be able to navigate a controlled environment, avoiding obstacles and finding alternative paths when necessary. This report will discuss the design considerations, algorithms, and control systems utilized to achieve the desired behavior. Experimental results and performance analysis will be provided to evaluate the effectiveness and efficiency of the implemented system.

2 Sonar-Assisted Autonomous Navigation

Autonomous underwater navigation using sonar involves the use of sonar technology to allow underwater vehicles or robots to navigate and map their surroundings without direct human control. Sonar, which stands for "sound navigation and ranging," is a system that uses sound waves to detect and locate objects underwater. It operates based on the principle of sending out sound waves and listening for their reflections off of objects in the environment.

1. Sonar System: The underwater vehicle is equipped with a sonar system that includes one or more transducers. These transducers emit sound waves into the water and listen for the echoes or reflections of those waves.

2. Sound Waves: The transducers emit sound waves, typically at ultrasonic frequencies, which travel through the water. The sound waves propagate in all directions and interact with objects they encounter.

3. Echo Detection: When the sound waves encounter an object, such as the seabed, underwater structures, or other obstacles, they bounce off the object and return to the transducer as echoes. The transducer detects these echoes.

4. Echo Processing: The echoes received by the transducer are processed to extract useful information. This may involve filtering out noise, amplifying the signals, and converting them into digital form for further analysis.

5. Object Detection and Mapping: By analyzing the characteristics of the echoes, such as their intensity, time of flight, and Doppler shift, the system can determine the presence and location of objects in the environment. This information is used to create a map or model of the underwater surroundings.

6. Navigation and Localization: Using the map created from the sonar data, the autonomous underwater vehicle can navigate through the underwater environment. It can use the information about obstacles, contours of the seafloor, and other features to plan its path and avoid collisions.

7. Feedback and Control: The sonar system continuously provides feedback to the vehicle's control system, enabling it to make real-time adjustments to its navigation. For example, if the vehicle detects an obstacle, it can change its course or speed to avoid a collision.

8. Localization and Positioning: Sonar data can also be used for localization and positioning of the underwater vehicle. By comparing the echoes received by multiple transducers, the system can estimate the vehicle's position relative to known landmarks or reference points.

Autonomous underwater navigation using sonar is a valuable technology for various applications, including underwater exploration, environmental monitoring, marine research, underwater mapping, and inspection of underwater structures. It allows for efficient and safe operation of underwater vehicles in environments where direct human control may be impractical or impossible.

3 BlueROV Ping360 Sonar characteristics

The Ping360 is a mechanical scanning sonar for navigation and imaging. It has a 50 meter (165 foot) range, 300 meter (984 foot) depth rating, and an open-source software interface that makes it a capable tool for ROV navigation and underwater acoustic imaging.

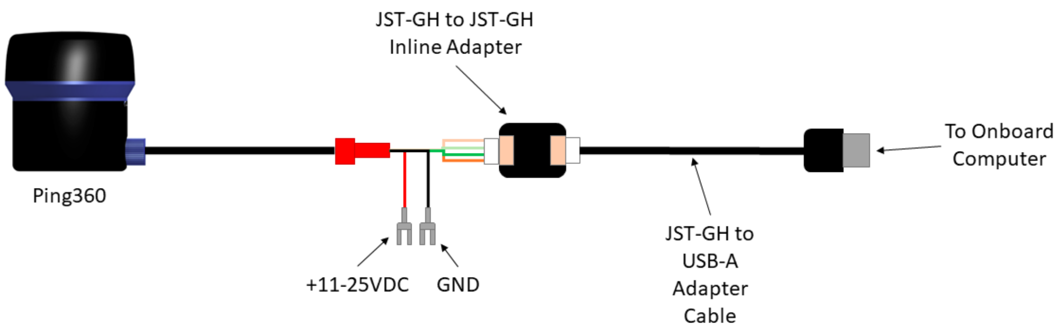
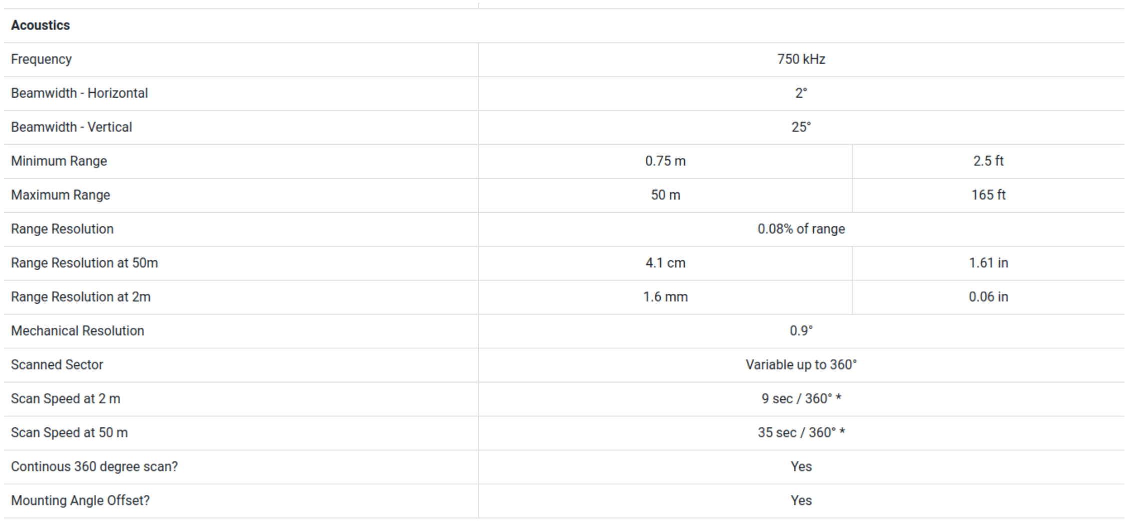


Figure 1: Ping360 Sonar sensor wiring

Inside the Ping360 is an acoustic transducer that sends a narrow beam of acoustic energy into the water and then listens back for echoes. That transducer is mounted to a motor that rotates it in one degree increments and as it does this it generates a circular image of the sonar's surroundings with a maximum range of 50 meters (165 feet). The result is similar to what you might see from a weather radar on the local news or a laser scanner on an autonomous robot. Here's an example of a scan of dock pilings straight ahead of the BlueROV2:



Acoustics		
Frequency	750 kHz	
Beamwidth - Horizontal	2°	
Beamwidth - Vertical	25°	
Minimum Range	0.75 m	2.5 ft
Maximum Range	50 m	165 ft
Range Resolution	0.08% of range	
Range Resolution at 50m	4.1 cm	1.61 in
Range Resolution at 2m	1.6 mm	0.06 in
Mechanical Resolution	0.9°	
Scanned Sector	Variable up to 360°	
Scan Speed at 2 m	9 sec / 360° *	
Scan Speed at 50 m	35 sec / 360° *	
Continous 360 degree scan?	Yes	
Mounting Angle Offset?	Yes	

Figure 2: Ping360 Datasheet

4 BlueROV autonomous navigation Implementation

To achieve autonomous navigation, the PressureCallback function utilizes pressure sensor data to estimate the robot's distance from obstacles. It calculates control signals for precise positioning, implements state-based behavior for obstacle detection and path finding, and enables autonomous exploration by rotating and recording measurements. This crucial function ensures accurate position control, obstacle avoidance, and autonomous behavior in autonomous navigation applications.

The state flow of our algorithm for free path exploration with obstacle avoidance is shown in figure 3

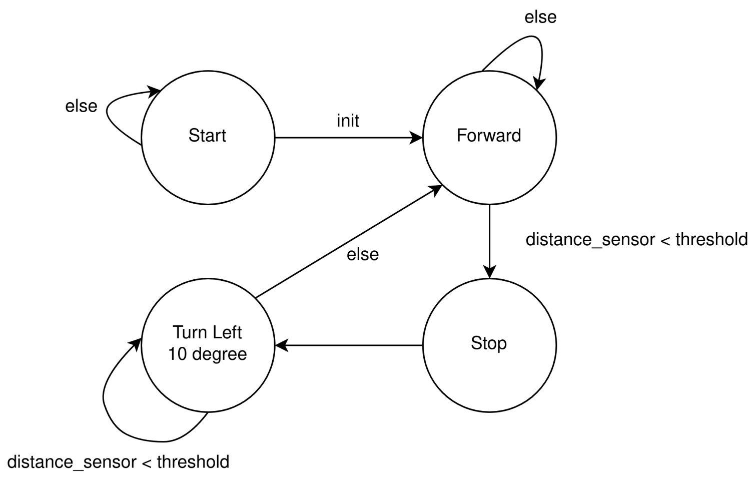


Figure 3: State flow of the free path exploration

The state of the system consists of 4 states

4.1 Start or initialize state

In this state, we initialize the robot parameter and set up every thing and when the robot is ready, the state of the robot move into the forward state.

4.2 Forward state

In this state, we implement the control algorithm that control the heading of the ROV (yaw control) and make the robot moves at the constant velocity. The yaw control is implement using P control shown in

While in this state, the robot will attempt to maintain the yaw angle while moving forward at constant velocity, instead of close loop velocity control from the distance sensor. The constant velocity performs better than PID control of heading due to the delay feedback of sonar distance sensor, and the ramp up velocity due to the initial error is large to calibrate. The robot will maintain this state until the robot encounter that can be detected from the sonar sensor within a certain threshold.

4.3 Stopping state

After the robot encounter with the obstacle, the robot will stop and enter the next state. This help preventing the robot from collide with the obstacle while turning.

4.4 Turning Left state

When enter this state, the robot will rotate a fixed degree angle (10 degree) then check the value from the distance sensor. If the distance is less than a certain threshold, it means that the path in front of the robot is not clear and the robot will keep entering the same state and gradually increase the angle until the path is clear.

5 Code explanation

```
 1 floatability = 2.7*9.8
 2 z_des = 0.1
 3 yaw_tol = 20
 4 x_des = 1000
 5
 6 yaw = float(angle_wrt_startup[2])
 7 z = float(depth_wrt_startup)
 8 x = float(pinger_distance)
 9
10 t_z = pControlYaw(yaw_des, yaw)
11 f_z = pControlwFloatability(z_des, z, floatability)
12
13 Correction_depth = int(force2PWM(f_z/4))
14 Correction_yaw = int(force2PWM(-t_z/4))
15
16 if state_flag == "FW":
17   goFoward(Correction_yaw, Correction_depth)
18   # if find wall
19   if x < x_des:
20     state_flag = "L"
21     # stop the robot
22     setOverrideRCIN(1500, 1500, Correction_depth, 1500, 1500, 1500)
23     yaw_des += 10
24
25 elif state_flag == "L":
26   rotateLeftState(Correction_yaw, Correction_depth)
27   if yaw < yaw_des + yaw_tol and yaw > yaw_des - yaw_tol:
28     if x < x_des:
29       yaw_des += 10
30     else:
31       state_flag = "FW"
```

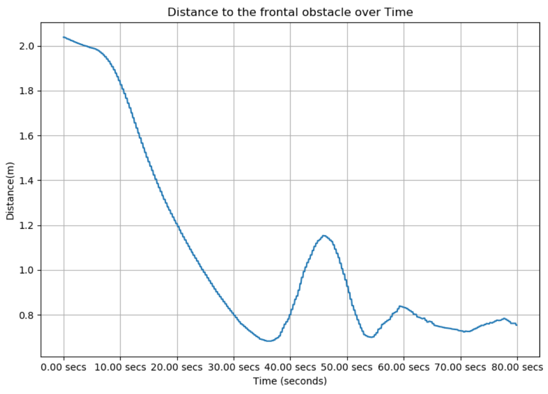
Listing 1: Code implementation

- From line 1 to 3: Initialize the parameters of the controller
- From line 5 to 13: calculate the PWM value by , first, calculate the error between each quantities (depth or yaw angle) then transform the force or torque to PWM using the manufacturing specification of thrusters.
- From line 15 to 22: code for forward state which will transfer to next state (L/Left turing state) when the distance from distance sensor is less than x_des
- From line 25 - 31: if the angle is less than certain threshold then it will move back to forward state. if not it will gradually increase the target angle by 10 degree until the condition is met.

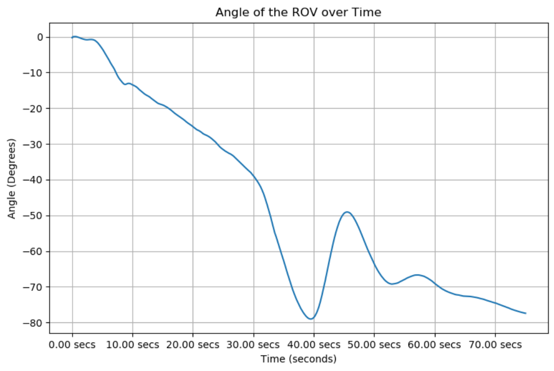
5.1 Free path exploration implementation

5.1.1 Results First experiment

The first experiment's tests were conducted for a duration of 80 seconds at the water surface. During the initial readings, the ROV's sonar measured the distance from the pool wall while facing 0 degrees. As the ROV moved forward, this distance gradually decreased until it reached approximately 70 cm from the front wall after 40 seconds, with an average speed of around 0.035 m/s. Subsequently, the ROV slowly turned left, reaching approximately -30 degrees, and detected an obstacle located approximately 120 cm away from the ROV at around 47 seconds. The ROV then moved towards this second obstacle, positioning itself at a distance of 70 cm. At -64 degrees, the ROV started rotating again, this time between the 60th and 70th seconds, until it encountered another obstacle.



(a) Distance to the obstacle, experiment 1.



(b) Heading of the BlueROV,experiment 1.

Figure 4: Results of first experiment

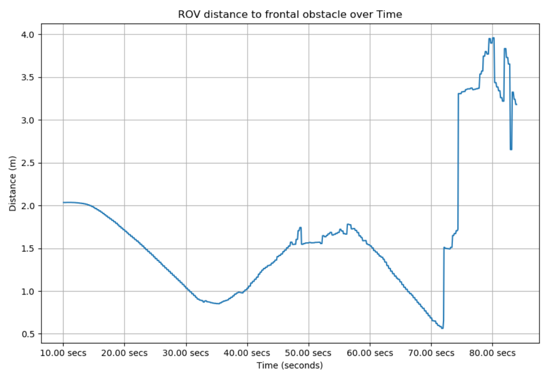
In general, the first experiment demonstrates successful results of the ROV in obstacle avoidance. The ROV is capable of detecting the distance to the obstacle and adjusting its orientation slowly to the left when the distance exceeds 70 cm. However, it lacks precise control in maintaining its heading during movement, as the ROV tends to incline slightly to the right while navigating.

5.1.2 Video results First experiment

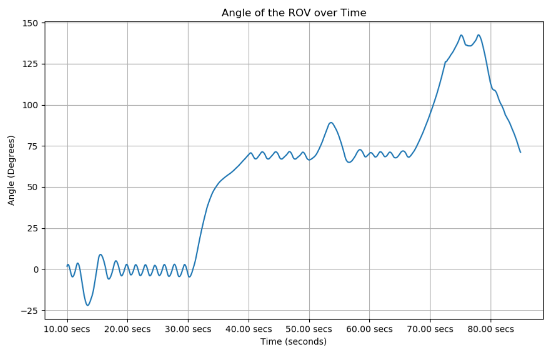
Video - Experimental results, No. 1.

5.1.3 Results Second experiment

In the second experiment, the ROV initially points towards the wall, causing the sensor to fall into the dead zone of readings and reducing its reliability. Therefore, the first 10 seconds of sensor readings are not displayed graphically. The second experiment lasts for approximately 60 seconds, during which the ROV moves towards the front wall while attempting to maintain a yaw control angle of zero degrees. However, it exhibits some oscillations and overshoots around the desired angle of 0 degrees. The ROV starts moving until it detects a distance from the front wall approximately 25 seconds later, with an average speed of 0.08 m/s. It then slowly rotates to the left, reaching approximately 75 degrees at 40 seconds, where it reads a distance greater than 1 meter. Due to this inclination, the ROV manages to maintain a distance of around 2 meters from the front wall until the 68th second, where it begins to rotate again, reaching approximately 150 degrees and pointing along the length of the pool, reading distances of approximately 4 meters.



(a) Distance to the obstacle, experiment 2.



(b) Heading of the BlueROV,experiment 2.

Figure 5: Results of Second experiment

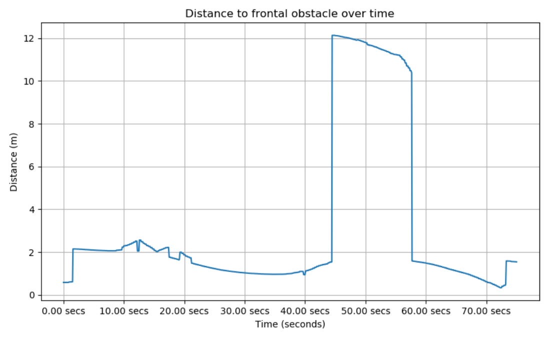
The second experiment demonstrated an overall improvement compared to the first one, despite the oscillations observed in maintaining the desired angle. The ROV exhibited increased speed, approximately three times faster than in the first experiment, allowing it to reach the target obstacle more swiftly. However, the ROV encountered dead zones in sensor readings, requiring operator intervention to adjust its orientation for reliable measurements.

5.1.4 Video results Second experiment

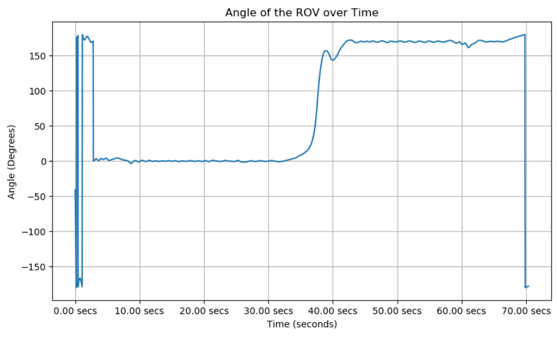
Video - Experimental results, No. 2.

5.1.5 Results Third trial

The sudden and rapid change in trajectory caused by the collision with another ROV during the 60th minute of the third experiment introduced a new dynamic to the ROV's movement. Initially pointed towards the surface wall, the ROV quickly adjusted its orientation within the first 10 seconds to face the side wall of the pool. Despite encountering some sensor noise, the ROV exhibited smooth and controlled navigation towards the wall, maintaining a steady orientation at 0 degrees for a significant duration of time. As the experiment progressed, the ROV executed a deliberate and gradual left rotation, gradually reaching approximately 150 degrees. This intentional rotation aligned the ROV with the length of the pool, allowing it to maintain a consistent heading and trajectory. However, the unforeseen collision with another ROV disrupted its planned path and abruptly altered its course.



(a) Distance to the obstacle, experiment 3.



(b) Heading of the BlueROV,experiment 3.

Figure 6: Results of Third experiment

In general, the third experiment exhibits reduced oscillation in maintaining orientation, allowing the ROV to navigate at a consistent speed and successfully avoid obstacles while maintaining a constant heading. Due to the proximity of the robot to the pool corners, the ROV is forced to make a nearly 150-degree turn to align itself longitudinally with the pool's direction.

5.1.6 Video results Third experiment

Video - Experimental results, No. 3.

5.2 Conlcusion

In conclusion, the state-based algorithm shows good potential for achive obstacle avoidance in a ROV based on the findings of tests 1, 2, and 3. The system efficiently recognizes impediments, steers clear of them, and modifies the direction of the ROV as necessary. To improve the algorithm's capacity to take longer paths without assistance from humans, additional tuning is necessary.

1. In experiment 1, the algorithm identified obstacles with accuracy, kept a fixed heading, and inadvertently came to a stop at a certain distance from the obstacle. However, it had some difficulties maintaining heading control, as the ROV had a tendency to veer slightly to the right when moving ahead.

2. Experiment 2 showcased improvements in terms of speed and obstacle avoidance compared to the first experiment. The ROV navigated at a faster pace and displayed better obstacle avoidance capabilities, although it occasionally encountered dead zones in sensor readings, necessitating manual adjustments.

3. The third experiment demonstrated additional refinements in orientation maintenance, reduced oscillations, and consistent speed. The ROV efficiently navigated towards obstacles, adjusted its orientation, and avoided collisions. However, collisions with other ROVs resulted in sudden changes in trajectory, highlighting the need for collision avoidance mechanisms.

In general, obstacle identification, avoidance, and reorientation activities in autonomous navigation are efficiently facilitated by the state-based approach. The algorithm has the potential to significantly improve ROV autonomy and navigation by maintaining desired orientations, adjusting movements in response to sensor feedback, and exploring free pathways while avoiding obstacles.