



电子科技大学  
格拉斯哥学院  
Glasgow College, UESTC

FINAL YEAR PROJECT REPORT  
BACHELOR OF ENGINEERING

**Autonomous Grasping Control of Soft Robotic System Based on Underwater Visual Tracking**

Student: Zhuheng SONG

GUID: 2357651S

1<sup>st</sup> Supervisor: Li WEN

2<sup>nd</sup> Supervisor: Masood Ur REHMAN

2020–2021

# **Coursework Declaration and Feedback Form**

*This Student should complete and sign this part*

Student Name: Zuheng SONG	Student GUID: 2357651S
Course Code: UESTC4006P	Course Name: INDIVIDUAL PROJECT 4
Name of 1st Supervisor: Li WEN	Name of 2nd Supervisor: Masood Ur REHMAN
Title of Project: <b>Autonomous Grasping Control of Soft Robotic System Based on Underwater Visual Tracking</b>	
<b>Declaration of Originality and Submission Information</b>	
I affirm that this submission is all my own work in accordance with the University of Glasgow Regulations and the School of Engineering requirements  Signed (Student):  I affirm that this submission is completed by the student independently and the quality of the submission meets the requirements for graduation. I consent to the student taking part in the oral presentation.  Signed (1 <sup>st</sup> Supervisor):	 UESTC4006P TEC-IT.COM
Date of Submission:	
<i>Feedback from Lecturer to Student to be completed by Lecturer or Demonstrator</i>	
Grade Awarded:	
Feedback (as appropriate to the coursework which was assessed):	
Lecturer Demonstrator:	Date returned to the Teaching Office:

## **Abstract**

In this article, an autonomous seafood grasping system was designed and tested. This system gained from techniques from a lot of fields such as soft robotics, autonomous control, computer vision, submarine vehicle and embedded system, which in total supported this system to harvest specific undersea species with a high precise, rapidly responsible, reasonably robust in complex natural submarine environment.

# Contents

<b>Coursework Declaration and Feedback Form</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>iii</b>
<b>List of Notations</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
<b>2 Materials and Methods</b>	<b>3</b>
2.1 System Overview . . . . .	3
2.2 Hardware . . . . .	3
2.2.1 Design of the Vehicle . . . . .	3
2.2.2 Design and Actuation of the Soft Robotic Manipulator . . . . .	7
2.3 Software . . . . .	7
2.3.1 AUV Movement Control . . . . .	8
2.3.2 underwater vision restoration and target detection . . . . .	9
2.3.3 ArUco Makrer Detection . . . . .	10
2.3.4 Communication System . . . . .	10
2.3.5 Human-Robot Interaction . . . . .	12
2.4 Overall autonomous grasping procedure . . . . .	13
2.4.1 Landing State . . . . .	13
2.4.2 Grasping State . . . . .	13
2.4.3 Cruising State . . . . .	13
2.4.4 Aiming State . . . . .	14
<b>3 Experiments and Results</b>	<b>16</b>
3.1 Lab Tank Experiments . . . . .	16
3.2 Lab Pool Experiments . . . . .	16
3.3 Oceanic Experiments . . . . .	17
<b>4 Discussion</b>	<b>19</b>
<b>5 Conclusions</b>	<b>20</b>
5.1 Contributions . . . . .	20
5.2 Future Work . . . . .	20
<b>A wow</b>	<b>21</b>

## **List of Figures**

1	Autonomous seafood grasping system overview . . . . .	3
2	Vehicle components . . . . .	4
3	Device topology . . . . .	5
4	Propulsion system . . . . .	6
5	Control Unit . . . . .	7
6	Side View of ZED with waterproof cage . . . . .	8
7	Design and Mechanics of the Underwater Soft Manipulator . . . . .	9
8	Scheme of Underwater Seafood Detector . . . . .	10
9	Communication System of the Autonomous Seafood Grasping System . . . . .	11
10	A Snapshot of the Graphic Interface of the Client Program on Master Computer . .	12
11	Procedure of the Autonomous Grasping System . . . . .	14
12	Trajectory of the Prescribed Cruise Path . . . . .	15
13	Early Experiment Setup at Lab Tank . . . . .	16
14	Grasping Procedure within the Lab Pool and Cruise Route Used . . . . .	17
15	Snapshots of site of oceanic experiments . . . . .	18
16	Grasping Procedure Captured at Offshore Sea Bed . . . . .	18

## **List of Tables**

1	Possible Commands and Sensor Signals for STM32 board . . . . .	6
2	Movements of AUV with Status of Each Thruster for Horizontal Shift . . . . .	9
3	Main Configuration of Jetson AGX Xavier Used . . . . .	22
4	Version of Libraries Used . . . . .	22

# List of Notations

API	Application Programming Interface
AUV	autonomous underwater vehicle
CV	Computer Vision
DoF	degrees of freedom
FPS	Frames Per Second
$g$	gravitational acceleration, $9.8m/s^2$ used
GAN	Generative Adversarial Networks
GUI	Graphical User Interface
$h_d$	depth of the AUV
I-AUV	intervention AUV
IMU	Inertial Measurement Unit
OBSS	opposite-bending-and-stretching structure
PID	Proportional-Integral-Derivative
PMMA	Poly(methyl methacrylate), also known as acrylic
PWM	Pulse Width Modulation
ROI	Region of Interest
ROV	remotely operated vehicle
RTSP	real time streaming protocol
SGBM	Semi-Global Block Matching
SSD	Single-Shot Detector
TCP	Transmission Control Protocol
$\alpha$	pitch angle of the AUV
$\beta$	roll angle of the AUV
$\gamma$	yaw angle of the AUV
$\theta_1$	curvature angle of the upper bending segment
$\theta_2$	curvature angle of the lower bending segment
$\rho_e$	submarine environment density, $1025kg/m^3$ used

# 1 Introduction

## 1.1 Motivation

People's demands on seafood with high nutrition such as sea urchin, sea cucumber and scallop, are increasing rapidly these days. However, because of the requirements of breeding environment, these seafoods are currently kept in offshore huge cages at relatively deep sea floor (more than 10 meters depths) and harvested artificially by seafood divers, which are also called 'Haipengzi' in some parts of China (CCTV, 2014). The seafood divers not only suffer from long working time (about 12hours) but also risk psychological stress, life hazards, hypothermia disease and decompression illness (Barratt *et al.*, 2002). Besides, the maintenance of boat, diving equipments costs a lot (MrAquaImage, 2013).

Therefore, robot solutions for such submarine operations are urgently needed to decrease diver related injury, reduce harvest cost and improve manipulation efficiency. Several ROV solutions have already been carried out recent years. For instance, a modified ROV from Nomia, whose capture is implemented by vacuum absorption controlled by operator onboard the vessel, achieved a notable amount of 6595kg export quality urchins out of total harvest of 1.88 ton in 4.5 days (James *et al.*, 2012). It shows out the advantage of robot in place of divers when it comes to the cold working conditions and dark period working. The disadvantage of this solution is that the vacuum absorption may result in submarine vegetation environment damage. In addition, its bycatch is still too much, result in relatively low efficiency.

There are a big amount of ROVs designed for submarine rescue, observation and similar tasks, including oceanic bottom flying node (Qin *et al.*, 2019), H300 ROV of French ECA group (ECA group). These ROVs share some common features:

- the operations are realized by pilot through remote control
- launched from specific mother vessel with an umbilical cable for communication and energy supply
- large and heavy for transportation and handling
- has complex interface which may lead to pilot fatigue after long hours of operations (Prats *et al.*, 2011)

Different from ROVs, AUVs, such as SAUVIM AUV (Marani *et al.*, 2014), Girona 500 I-AUV (Ribas *et al.*, 2015) could achieve environment perception, analysis and operative tasks autonomously and independently in complicated environment (Barblat *et al.*, 2014). Therefore, National Natural Science Foundation of China (NSFC) launched a group of key projects of underwater vehicle environmental perception and target capture to realize robotic capturing in 2016. The main objectives of these projects are to develop low cost underwater robot capturing platforms on the basis of the following researches:

1. environmental perception of underwater landscapes, plants and cur-rents
2. rapid detection, recognition and tracking of target organism

3. aliveness and rapid capture of target organism

From these researches, not only the oceanic organism harvest is expected to be mechanized, but also the underwater rescue, oceanic engineering and submarine exploring are propelled.

This article tries to give solution to following questions:

1. How to integrate soft manipulator into mobile remote controlled autonomous underwater vehicle robot
2. How to find underwater targets and then collect them safely
3. How to achieve seafood collecting tasks in complex water environment

## 2 Materials and Methods

### 2.1 System Overview

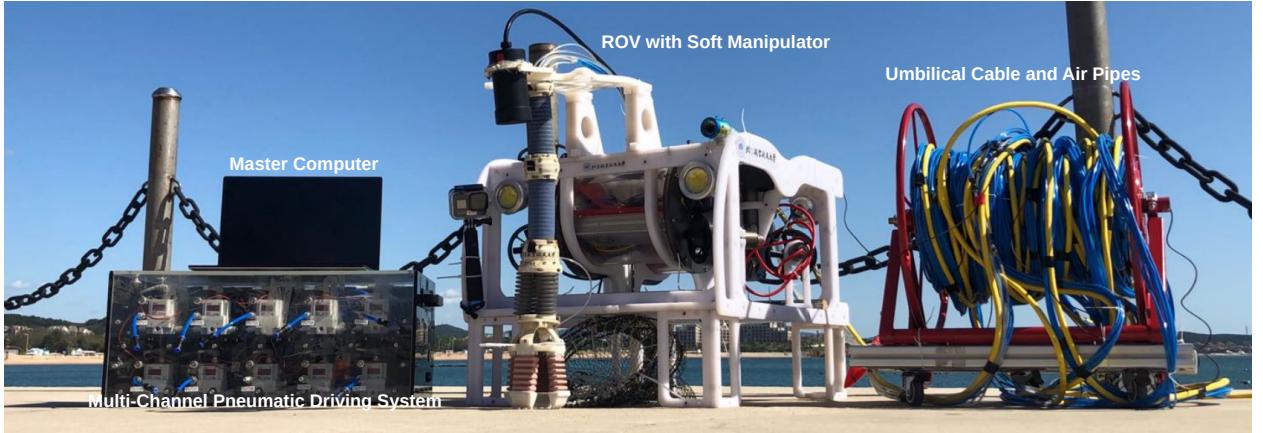


Figure 1: Autonomous seafood grasping system overview: a modified ROV, an OBSS soft manipulator and a master computer

As shown in Figure 1, the auto seafood grasping system consists of a modified ROV, an OBSS soft manipulator, a multi-channel pneumatic driving system and a master computer. The OBSS soft manipulator is installed vertically down and was controlled to pick-and-place seafood into a collecting basket fixed to the bottom front of the I-AUV. A modified 4-DoF (forward/backward, left/right, up/down, spin) ROV is integrated with a binocular camera ZED for cruise and target detection, which placed with an angle to vertical down direction, and a fishing camera to provide a wider view to monitor the whole working environment. These two cameras are connected to the control unit of the I-AUV, where the video stream from the binocular camera was processed. The processed binocular camera video stream together with video stream from the fishing camera were sent to the master computer onboard the vessel by RTSP though a local network. There, on the master computer low latency monitoring and controlling of the I-AUV in emergent situations were allowed. The I-AUV measures 600 mm long, 500 mm wide, and 300 mm tall, with a weight of approximately 30 kg and an operating depth of 0-50 meters.

### 2.2 Hardware

#### 2.2.1 Design of the Vehicle

The entire vehicle is visualized in Figure 2. The vehicle is in shape of  $600\text{mm} \times 500\text{mm} \times 300\text{mm}$ , weighing around 30kg, consist of the frame, a depth sensor, a propulsion system, four LED lights, a binocular camera, a fishing camera, two sealed warehouse and a collecting basket. The white frame are assembled laser-cut PMMA boards with thickness of about 1 cm, fixed to each other with at least four screws. This structure guarantees stability of the robot under any circumstances in the depth range of 0-50 meters. PMMA could gain substantial amount of buoyancy whether in lab pool condition or submarine condition, in the meanwhile, huge hollowed-outs are applied

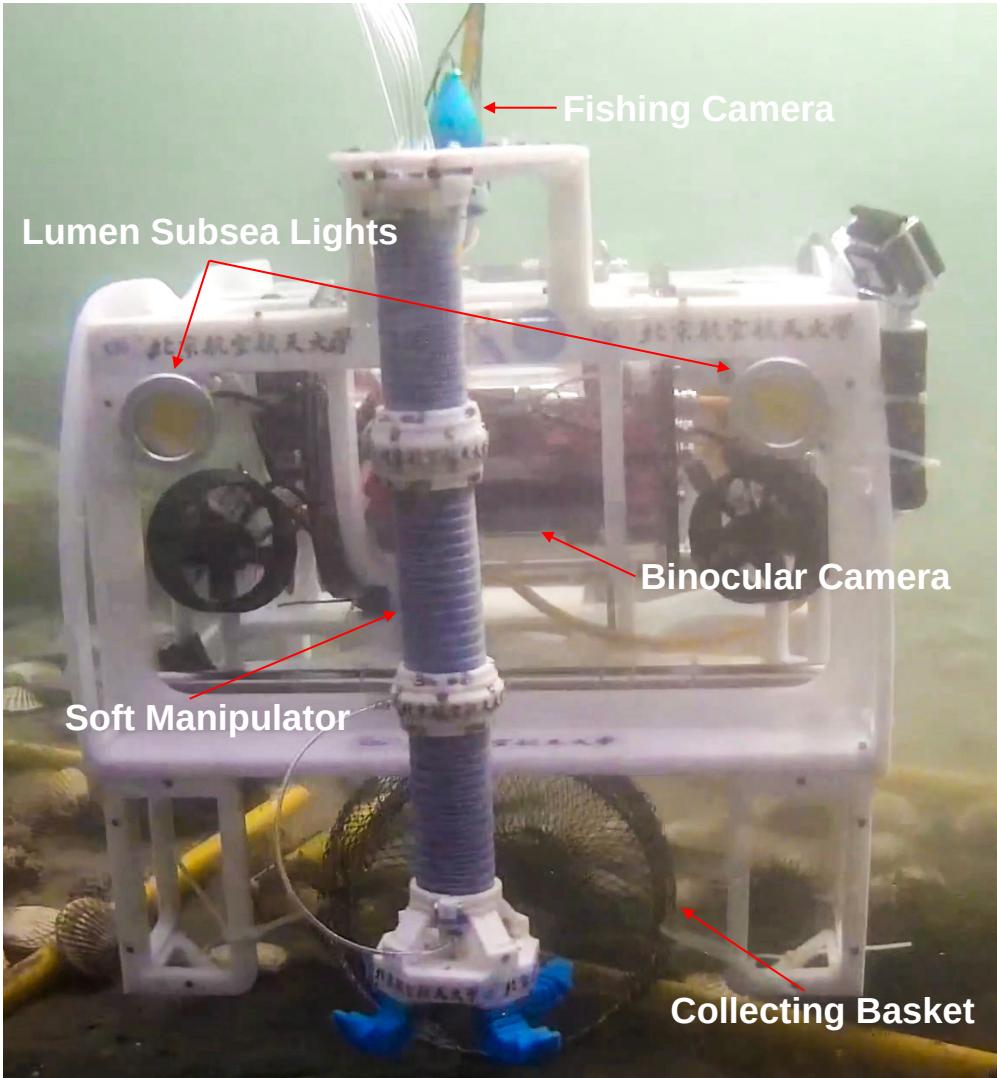


Figure 2: Vehicle components: snapshot of the underwater robot system with a soft manipulator for grasping fragile sea animals.

to the design, both of which helped to reduce the total weight of the vehicle. The natural light condition at more than 10 meters below sea level could be horrible for target detection by CV. Therefore the headlights and taillights were applied to provide supplementary lighting to enhance the quality of images captured by the binocular camera and fishing camera. Two column-shaped sealed warehouse were loaded inside the vehicle. The front warehouse contained a communication unit including a ethernet-to-fiber converter, power unit and the back warehouse contained a control unit, a 6-channel motor driver. The thrusters, depth sensor, two cameras, four lights are all connected to the control unit (Figure 3).

**Depth Sensor** The depth sensor is placed near the bottom of the vehicle with its working surface downward, which direction proved to obtain minimum noise (less than 3 mm in most experiments) caused by the oceanic current. The depth is deduced from

$$h_d = \frac{P_d - P_0}{\rho_e g} \quad (1)$$

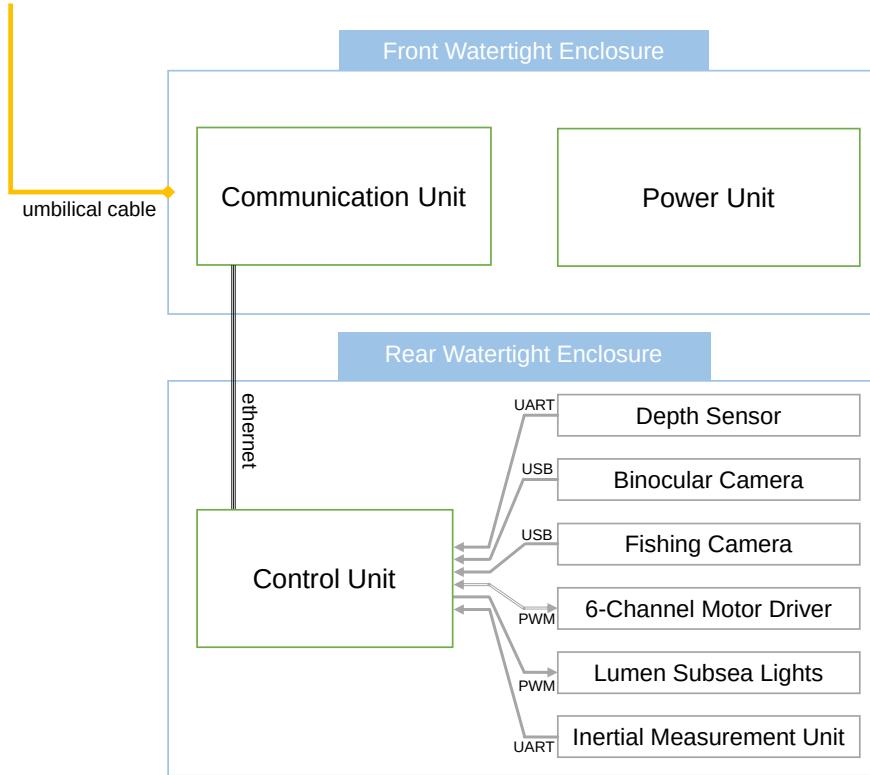


Figure 3: Device topology: topology of all modules and devices integrated on the vehicle

where  $P_d$  is the current pressure detected by the sensor,  $P_0$  is the initial pressure detected when the sensor was powered up,  $\rho_e$  is the underwater environment density ( $1,025\text{kg/m}^3$  is considered as the seawater density), and  $h_d$  is the current depth.

With accuracy of no more than 3 mm in most experiments, the precise of this depth sensor could definitely meet the specification required by tasks in this project.

**Propulsion System** Designed similar to a quadrocopter, the propulsion system of this I-AUV consists of 6 thrusters, four of which are placed at right the same level of the mass center, in charge of horizontal movements, and the other two are placed symmetric about the centerline, in charge of vertical movements (Figure 4). By keeping four horizontal movements thrusters at the same level of the mass center and symmetric against the center, when thrust forces are applied, there won't be torques generated against water resistance. The driving force provided by this propulsion system is actually not enriched for high mobility required by the complex operating condition. The max speed of forward/backward movement is approximately  $2.5\text{m/s}$  and max speed of side movement and vertical movement is approximately  $2\text{m/s}$ .

Speed of the motors were regulated with **PWM**, which basically controls the high level time of voltage supplied to the motors in a period (called duty cycle), with resolution of 256 levels (including forward and reverse rotation). Provided this technique, control of motor speed, from stop to full speed, through analog voltage supplied to the motors simply with an 8-bit digital input is allowed. Meanwhile, the voltage supplied to each motor is detected and recorded as sensor signals in real-time.

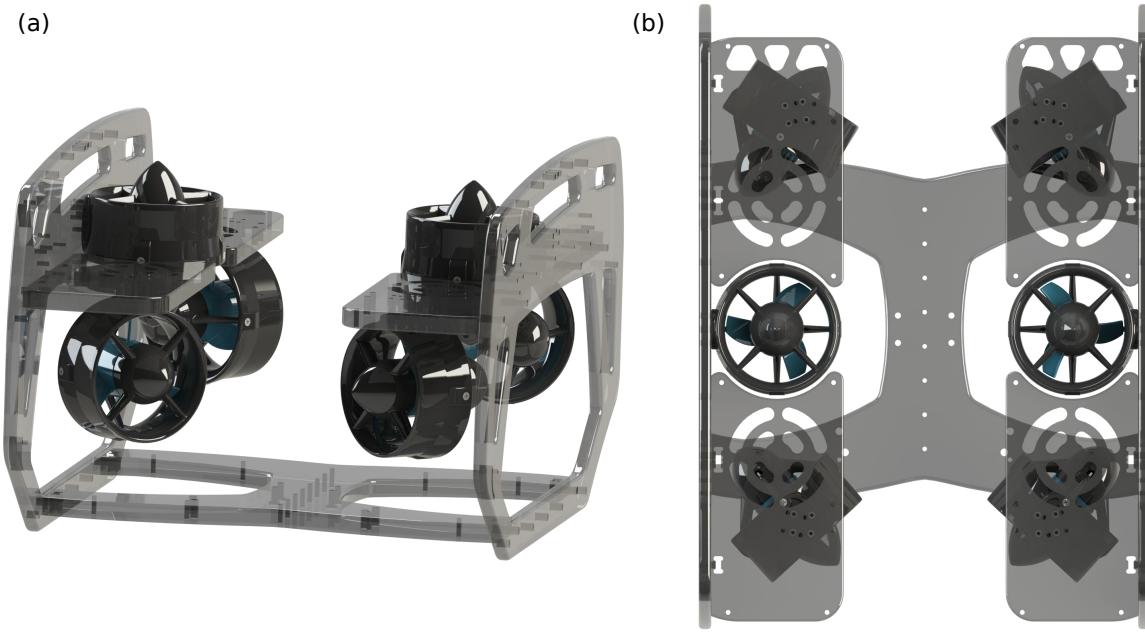


Figure 4: Propulsion system: (a) four thrusters for horizontal moving. (b) two thrusters for vertical moving

**Control Unit** The flow of signals within the control unit is depicted in Figure 5. The control unit is actually made up of two boards: a *Jetson AGX Xavier* for top-level control and a *STM32f407* for low-level control. The *Jetson AGX Xavier* is connected to the local network through ethernet, able to send required pressure values to the multi-channel pneumatic driving system as well as send data packages of video streams and data from sensors integrated on the vehicle to the master computer at about 20 FPS (and may receive commands from the master computer in emergent situations). In the meanwhile the *STM32f407* could only communicate with the *Jetson AGX Xavier*, receiving commands and transmitting data from sensors. Possible commands and sensor signals are listed in Table 1.

Possible commands	Possible sensor signals
lights on/off	depth of the vehicle, $h_d$
move upwards/downwards	voltage supplied to each motor
move forwards/backwards	pitch angle of the vehicle, $\alpha$
move leftwards/rightwards	roll angle of the vehicle, $\beta$
turn left/right	yaw angle of the vehicle, $\gamma$

Table 2: Possible commands sent to the *STM32f407* and possible sensor signals from it (left column and right column were not correspondent)

An 6-DoF IMU (composed of a 3-axis accelerometer and a 3-axis gyroscope) is integrated on the board of *STM32f407*. Both the acceleration information and angular velocity are used to enhance the movement control of the vehicle, but only Euler angles deduced by the angular

velocities are sent to *Jetson AGX Xavier* as signals. *STM32f407* kept sending digital signals to the 6-channel motor driver and 4-way lights at a rate of 30 Hz with resolution of 256 levels and 960 levels respectively, until new command received from the *Jetson AGX Xavier*.

A RTSP server was established on the *Jetson AGX Xavier* to publish live video streams from the *ZED* and fishing camera into the local network. This technique allow any device in the local network to get access to view from *ZED* or the fishing camera through whether browser, player like vlc or self-developed application with RTSP.

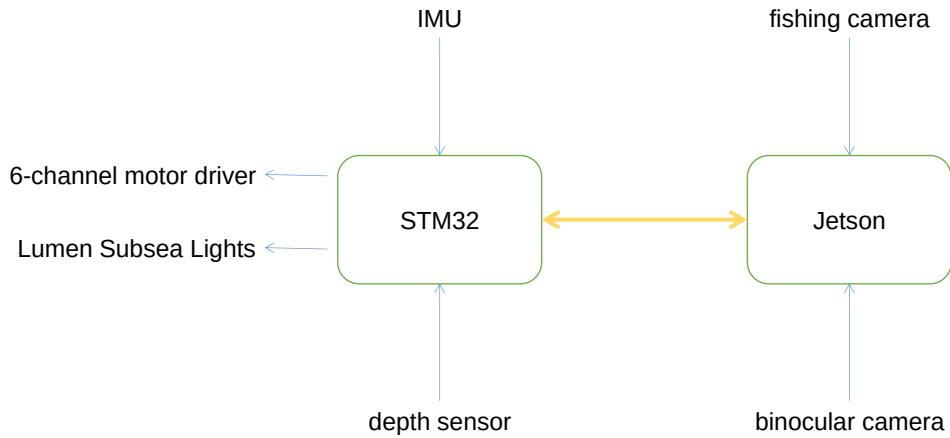


Figure 5: Control Unit: Block diagram of signal flow inside the Control Unit

**Binocular Camera** The *ZED* binocular camera is placed at the front of the vehicle, about the level of the vehicle's center (Figure 2), to get a relatively close view spanning the workspace of the manipulator's end effector. For such purpose, there is an angle of 30 degrees between the direction of the camera and the vertical downward direction. *ZED* is not waterproof by itself, so a 3D-printed waterproof cage is introduced to provide resistance to submersion up to a maximum depth of 30 meters underwater (Figure 6).

## 2.2.2 Design and Actuation of the Soft Robotic Manipulator

An three segments OBSS underwater soft manipulator was integrated onto the vehicle and used to grasp and place targets into the basket (Gong *et al.*, 2021). As indicated in Figure 7, the manipulator consisted of two bending segments, a stretching segment and a soft gripper. The manipulator was actuated with an opposing curvature where  $\theta_1 = \theta_2$ . The two bending segments had a joining angle of 180 degrees. In this assembling pattern, for instance, chamber ① of the upper bending segment is in the opposite position to chamber ① of the lower bending segment, where the intersection angle is 180 degrees.

## 2.3 Software

Programs ran in this autonomous grasping system could be defined as three modules: slave low-level module written in C++, slave high-level module written in C++ and master module written

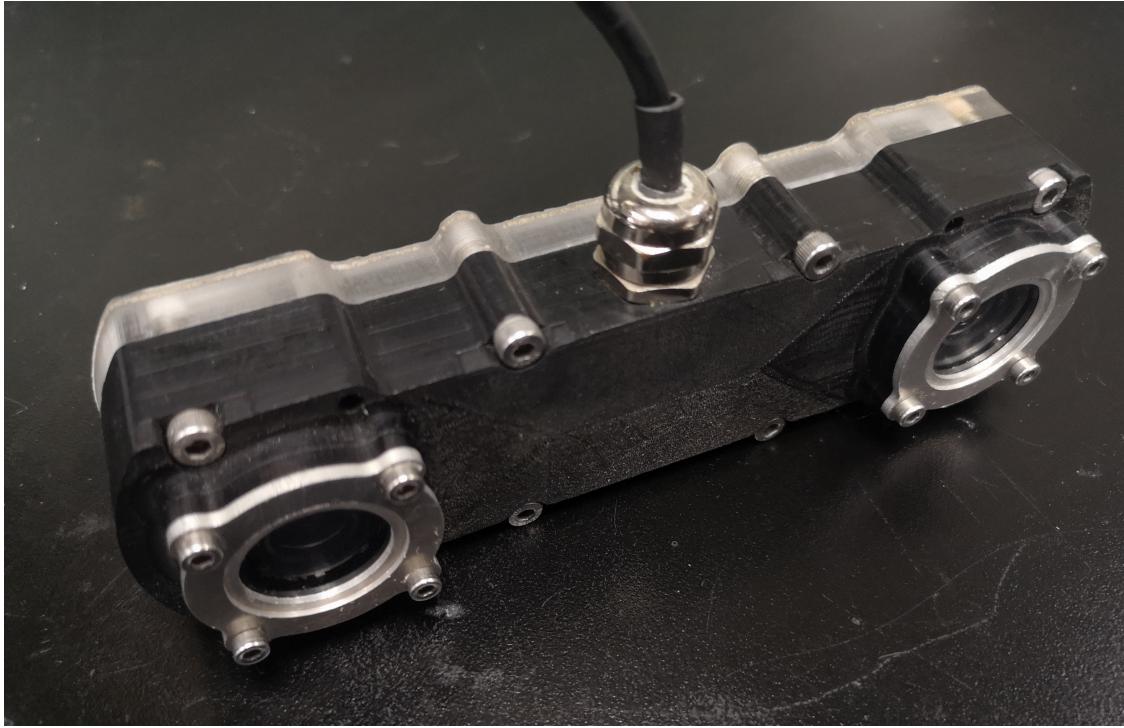


Figure 6: Side view of *ZED* with waterproof cage

in Python, on the *STM32f407*, *Jetson AGX Xavier* and master computer respectively. Most computational tasks were done in the slave high-level module by the *Jetson AGX Xavier*, which brought supercomputer performance (almost same compute capability of a GeForce RTX 2080 desktop version, (Corp., 2021)) to the edge. This module is in charge of a series of tasks which requires relatively high performance listed below:

- keeps pushing video streams from two cameras into the local network with *RTSP* once its system was started up
- realtime underwater vision restoration of video stream from the binocular camera, to enhance quality of the images
- realtime target (sea cucumber, sea urchin or scallop) detection and tracking from the latest frame from the binocular camera
- realtime target distance estimation with depth map deduced by *SGBM* algorithm (triggered if target detected)
- realtime ARUCO marker detection (triggered if grasping started)

Main configurations of the *Jetson AGX Xavier* and software environment used for *CV* in this project can be found in Appendix B.

### 2.3.1 AUV Movement Control

The speed on each directions are regulated by a 6-way *PID* controller on the *STM32f407*. The tuning of the controller was done with the Ziegler-Nichols method (Åström *et al.*, 2004). In

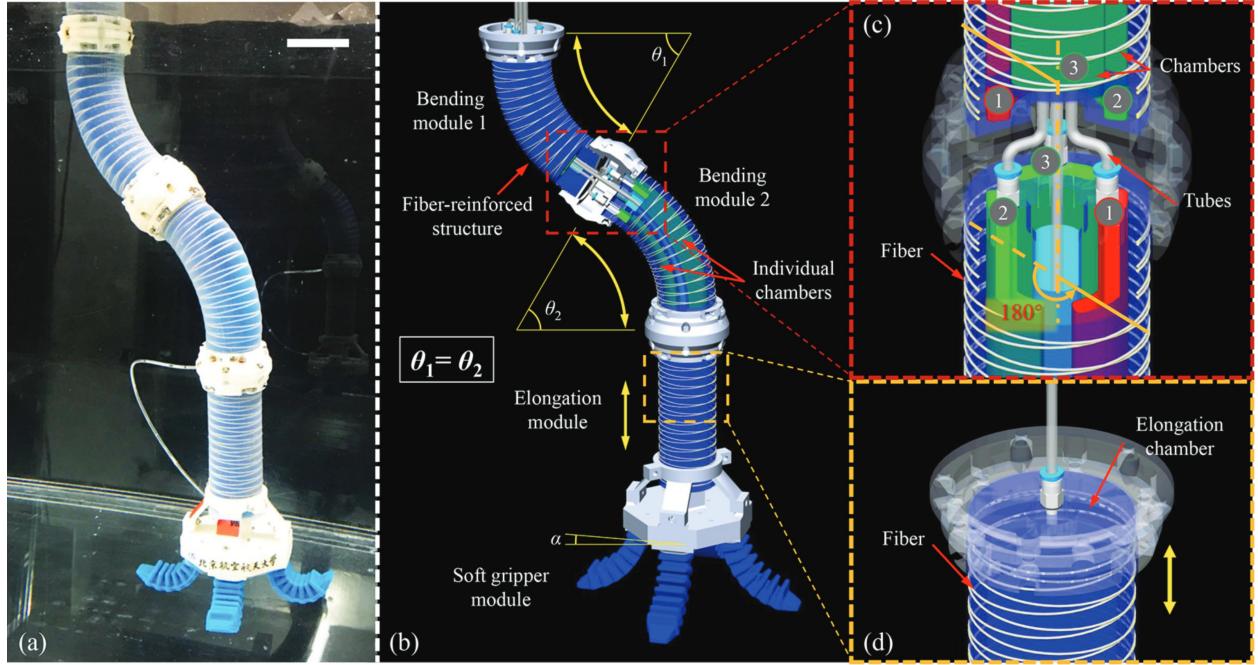


Figure 7: Design and mechanics of the underwater soft manipulator: (a) An overview of the OBSS soft manipulator (scale bar 50 mm). (b) Components of the manipulator with annotations of useful magnitudes. (c) An illustration of how chambers are arranged in the two bending segments. (d) A closer view of the upper end of the stretching segment.

addition, The depth of the I-AUV were also under PID control, which enables hovering stably in underwater environment with small current.

As shown in Figure 4, 6 thrusters are used to provide movements listed in. To simplify the kinematics model (therefore improves the robustness), turing and orientational moving at the same time is not allowed. How thrusters perform when doing horizontal movements are shown in Table 2. Besides, thruster 5 and thruster 6 were always working with the same direction of rotation, rotating clockwise to move upward and rotating anticlockwise to move downward.

movement	left front thruster	right front thruster	left rear thruster	right rear thruster
↑ forward	—	—	+	+
↓ backward	+	+	—	—
← leftward	—	+	—	+
→ rightward	+	—	+	—
○ left turn	+	—	—	+
○ right turn	—	+	+	—

Table 3: Movements of the AUV with status of each thruster for horizontal shift: + and – represent the positive propel (with this I-AUV, anticlockwise rotation) and reverse propel respectively

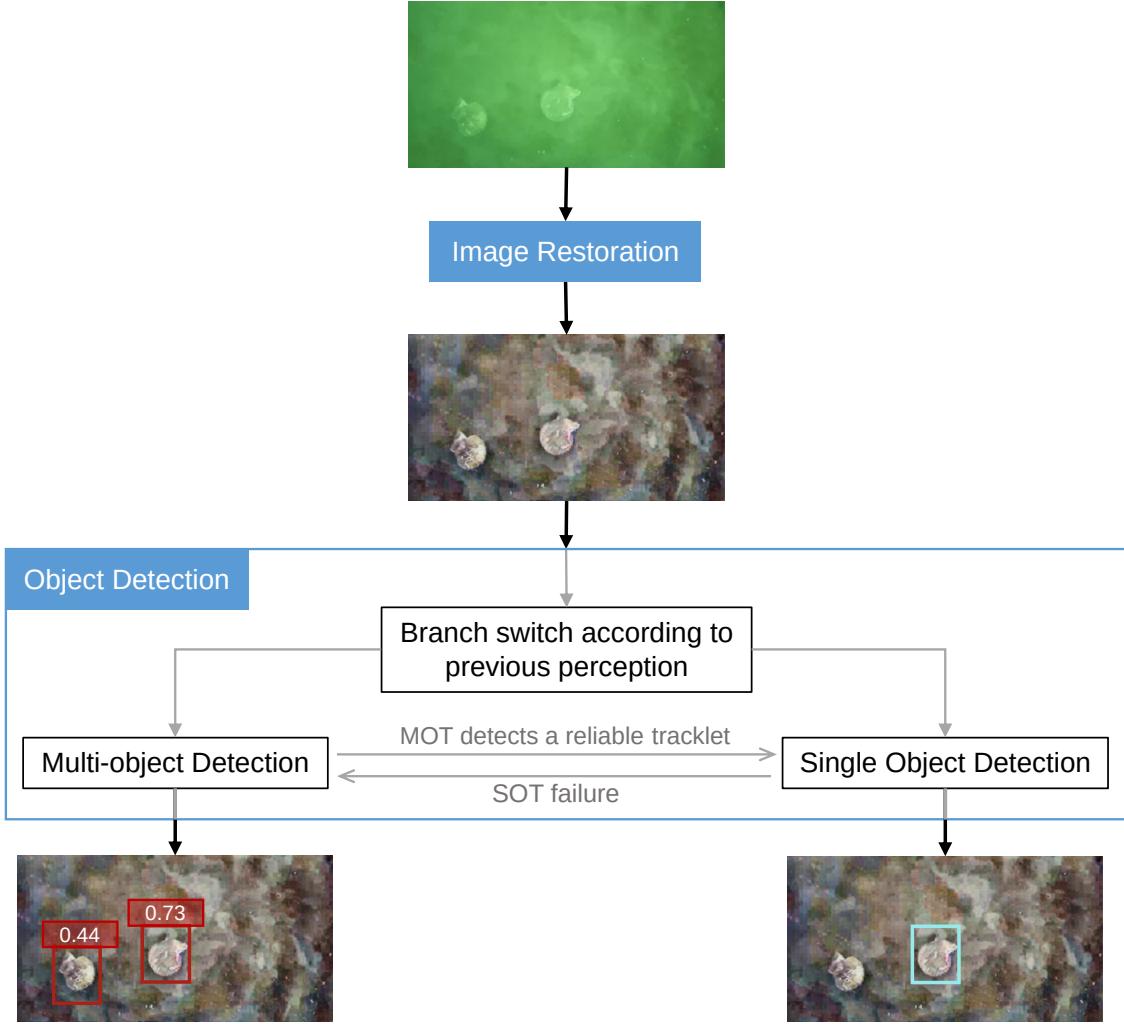


Figure 8: Scheme of underwater seafood detector integrated with TSSD-OTA and GAN-RS

### 2.3.2 underwater vision restoration and target detection

A real time adaptive GAN-based restoration scheme (GAN-RS) ([X. Chen, Yu, Kong, et al., 2019](#)) is integrated to enhance quality of underwater vision and induce an overall superior restoration performance. A temporally identity-aware SSD with online tubelet analysis (TSSD-OTA) is used to real-time detect and track target seafood with high performance and robustness ([X. Chen, Yu, and Wu, 2019](#)). Referring to Figure 8, the GAN-RS and TSSD-OTA are trained into a single PyTorch model, to increase the integration and decrease the complexity to deploy the system onto embedded device, *Jetson AGX Xavier* in this case, and could be called easily with libtorch (C++ API of PyTorch).

### 2.3.3 ArUco Makrer Detection

A tiny but with high performance and strong robustness C++ library ArUco ([Romero-Ramirez et al., 2018](#); [Garrido-Jurado et al., 2016](#)) is used to estimation pose of ARUCO markers placed on the hand of the soft manipulator. With ArUco, we can get real time position, rotation and translation of squared ARUCO markers against the camera. Detecting and tracking of multiple markers at the same time is supported, since an unique identifier could be decoded from the binary pattern of

every marker. Markers with size of  $5 \times 5$  blocks were used in this project.

### 2.3.4 Communication System

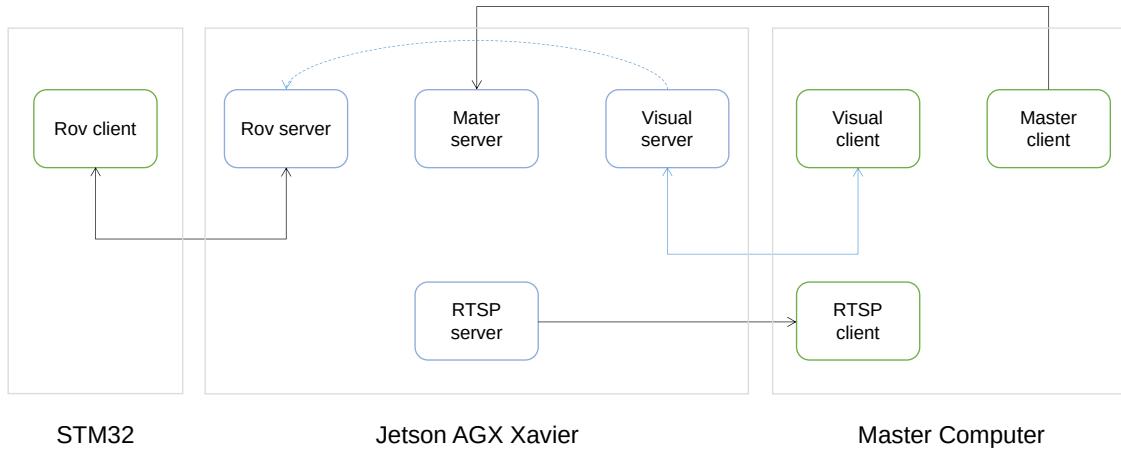


Figure 9: Communication system of the Autonomous seafood grasping system

As illustrated in Figure 9, there were in total 5 TCP socket servers and 5 correspondent TCP socket clients in the communication system of this autonomous seafood grasping system from 4 devices: *STM32f407*, *Jetson AGX Xavier*, master computer and multi-channel pneumatic driving system. Existing communication framework like ROS was not introduced for following three reasons:

- the communication system was of acceptable complexity, extra communication framework is not a must
- introducing huge frameworks may improve the robustness of communication, when it comes to handling unexpected situations, but more efforts were required to adapt the whole program to the framework used
- dependency of huge framework increases the workload to setup the whole system, especially when devices in the system spans a variety of architecture

As the system startup, the *STM32f407* and the *Jetson AGX Xavier* are powered on at the same time, but the *STM32f407* would start working first since it does not have to load a operating system. So the `rov_client` would be the first node initialed in the system and blocked at the `rov_client.connect()`, until the `rov_server` initialized and started listening. The `proxy_server` was actually used to port forwarding commands from the `proxy_client` to the `rov_client`, since a socket client could only connect to one socket server. The three socket servers in *Jetson AGX Xavier* would be started in different processes at almost the same time. The `visual_info` server was in charge of processing data from images from the binocular camera, and then send them to clients in YAML format. A piece of example `visual_info` data package is shown by Listing 1. Multi-threading technique is applied to the `visual_info` server, therefore concurrency of a maximum backlog of 8 is supported. In the meanwhile, the `rtsp_server`

was established and would keep publishing high definition raw video streams from the binocular camera and the fishing camera until the program exit.

```

1 target:
2   has_target: true
3   target_class: 1
4   id: 33
5   center:
6     x: 0.34
7     y: 0.66
8   shape:
9     width: 0.2
10    height: 0.31
11 arm:
12   arm_is_working: true
13   has_marker: true
14   position:
15     x: 0.31
16     y: 0.67

```

Listing 1: example visual info package

On the master computer, a `visual_client` and a `rtsp_client` were used to receive the visual data packages and video streams from the *Jetson AGX Xavier* respectively. Each piece of visual data package and image frame from the binocular camera was processed to a image with detected target and ARUCO markers visualized, shown on the screen and recorded to a log video at the same time Figure 10.

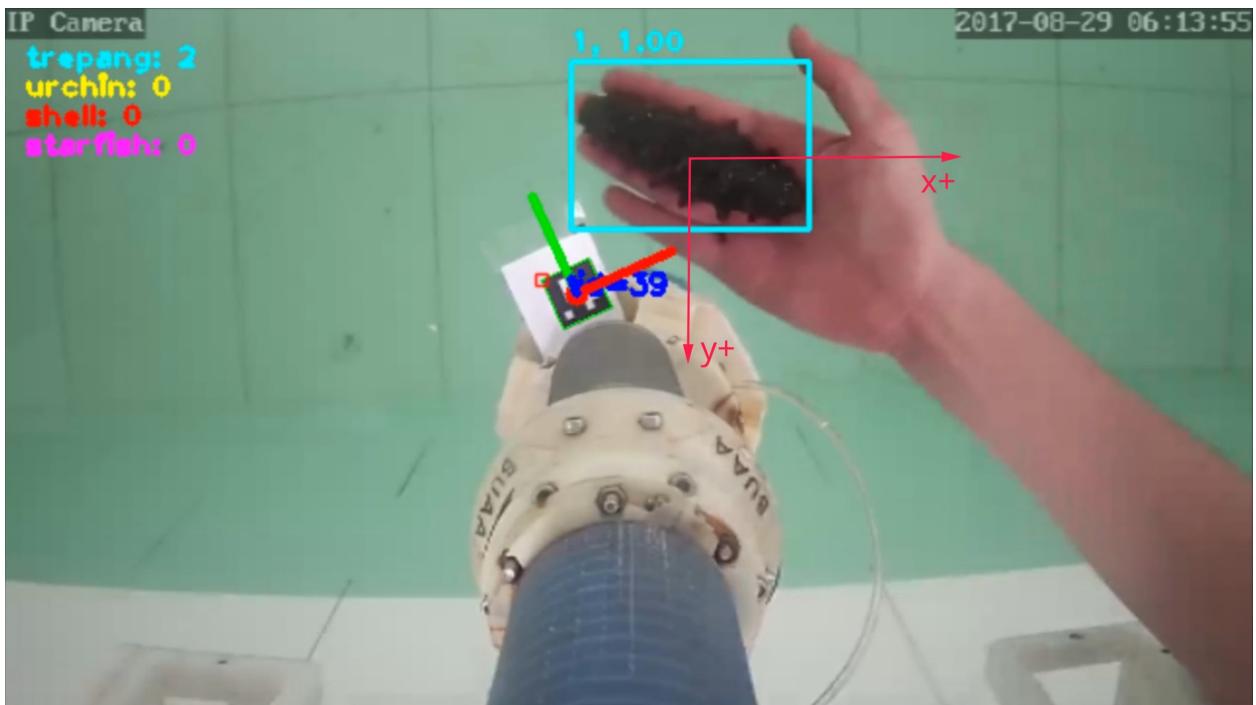


Figure 10: A snapshot of the graphic interface of the client program on master computer

### 2.3.5 Human-Robot Interaction

Operators could control the I-AUV by two input devices of the master computer: keyboard or joystick. Keyboard inputs are caught by the GUI monitoring and controlling I-AUV client program written in Python based on *OpenCV-3.4.11* with GTK GUI library (OpenCV built with Qt could not tell if the letter pressed is capitalized). The joystick was connected to the master computer by a wireless receiver and its raw output could then be read from /dev/input/js0, the first joystick recognized by the Linux system. The output was then parsed by a tiny self-written C program into human readable messages and mapped to simulated keyboard input with xdotool. Available keys with binding events were shown in subsubsection 2.3.5.

event	keyboard input	joystick input
stop	space	left shoulder trigger
forward/backward	w/s	up and down of the right stick
leftward/rightward	a/d	left and right of the left stick
turn left/turn right	A/D	left and right of the right stick
lights on/light off	L/I	
entering autonomous mode	enter	

## 2.4 Overall autonomous grasping procedure

- landing state
- grasping state
- cruising state
- aiming state

As shown in Figure 11, generally speaking, there were four states after the procedure starts:

### 2.4.1 Landing State

the I-AUV kept diving, try to reach the bottom of the water. It determine whether is landed by checking variance of the depth frequently. If variance of depth detected by depth sensor was less than 3 mm in 30 seconds, the I-AUV is considered to be landed.

### 2.4.2 Grasping State

the I-AUV would first detect if there is any target (sea cucumber, sea urchin or scallop) in its ROI (adjusted to fit the working space of the manipulator's end effector). If so, it would periodically send pressure values, calculated based on relative position between target and the manipulator, to the multi-channel pneumatic driving system and start to grasp. One grasp effort was given **1 minute**, leaving enough time for the manipulator to reach the target in relatively strong current.

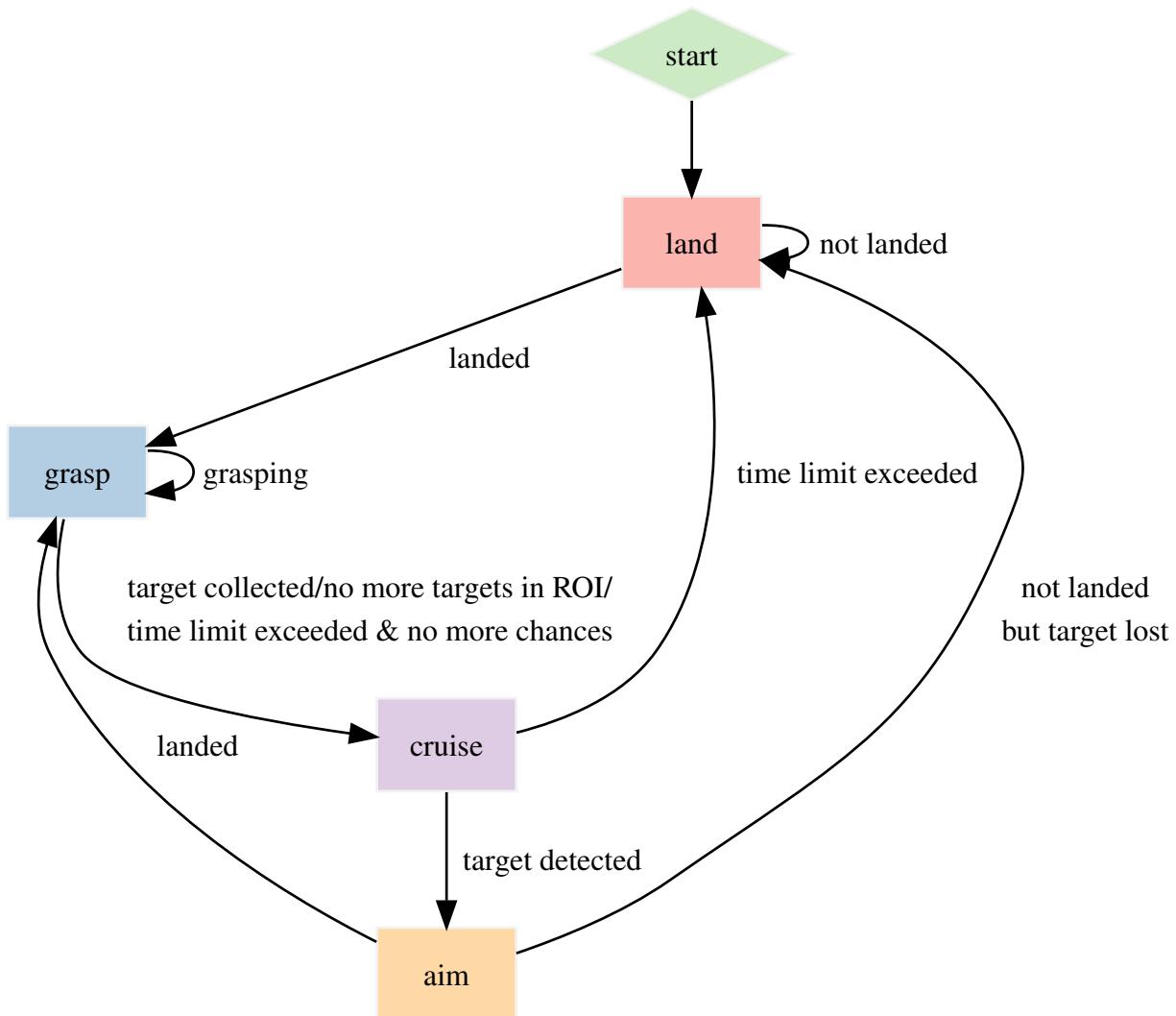


Figure 11: Procedure of the autonomous grasping system

When the time limit reached, the I-AUV would check whether the tracked target still exist in its ROI, if so, a second last grasp effort chance would been given, if not, the situation would be considered as target collected or lost (since there was no sensor in the collecting basket, these two situations cannot be distinguished). After the second effort, if the same target is still in sight, this specific target would be abandoned, considered hard to collect, which situation is actually not so rare in real submarine conditions. If there was simplify no target when entering the grasping state, the I-AUV would jump into cruising state directly.

#### 2.4.3 Cruising State

The I-AUV would move along a prescribed path and search for possible targets at the same time in this state. The I-AUV would go into the aiming state once target found. If nothings was found, the I-AUV would finish the path indicated in Figure 12 and then switch to the landing state. The reason why fallbacks to landing state frequently is that to reset the accumulated errors in the speed and depth PID controllers, as a result of executing tasks in such complex environment with currents.

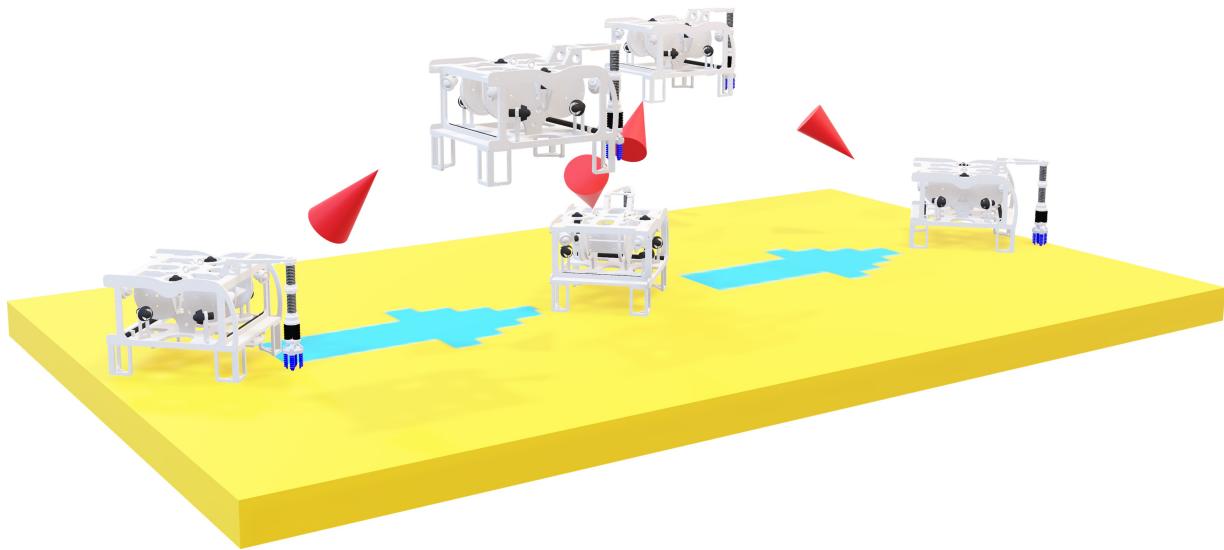


Figure 12: Trajectory of the prescribed cruise path

#### 2.4.4 Aiming State

The I-AUV was given **4 chances** to approach the target. The I-AUV would only be in this state if target was found in the ROI. Two levels of distance threshold in both x axis and y axis are applied to adjust speed of the I-AUV towards the target based on the relative position with the target. In addition, once target lost, it would jump to the landing state rapidly.

## 3 Experiments and Results

### 3.1 Lab Tank Experiments

The system was first validated in a lab tank without the vehicle (Figure 13). The binocular camera was connected to a computer and 2D relative positions of the manipulator's end effector, the target (rubber models of sea cucumber, sea urchin and scallop with counterweight inside) are detected from its video stream there. The processed data was transmitted to a MATLAB program (prototype of the manipulator controlling algorithm at this early stage) to calculate demanded air pressures to apply to the chambers of the soft manipulator. The manipulator was fully submerged into the water and a vibration pump was involved to simulate water flows in the real submarine environment. With several tests, it was proved that the system has the ability to detect and catch the seafood model with acceptable robustness. Limited by the performance of the computer used and unoptimized algorithm, the whole process (including detection, calculation of pressures needed with period of 1 seconds and grasp) took about 3 minutes in average. Besides, this system could not handle targets moving at a low speed, as well as handling static target when the system was moving, as another result of the two factors mentioned above.

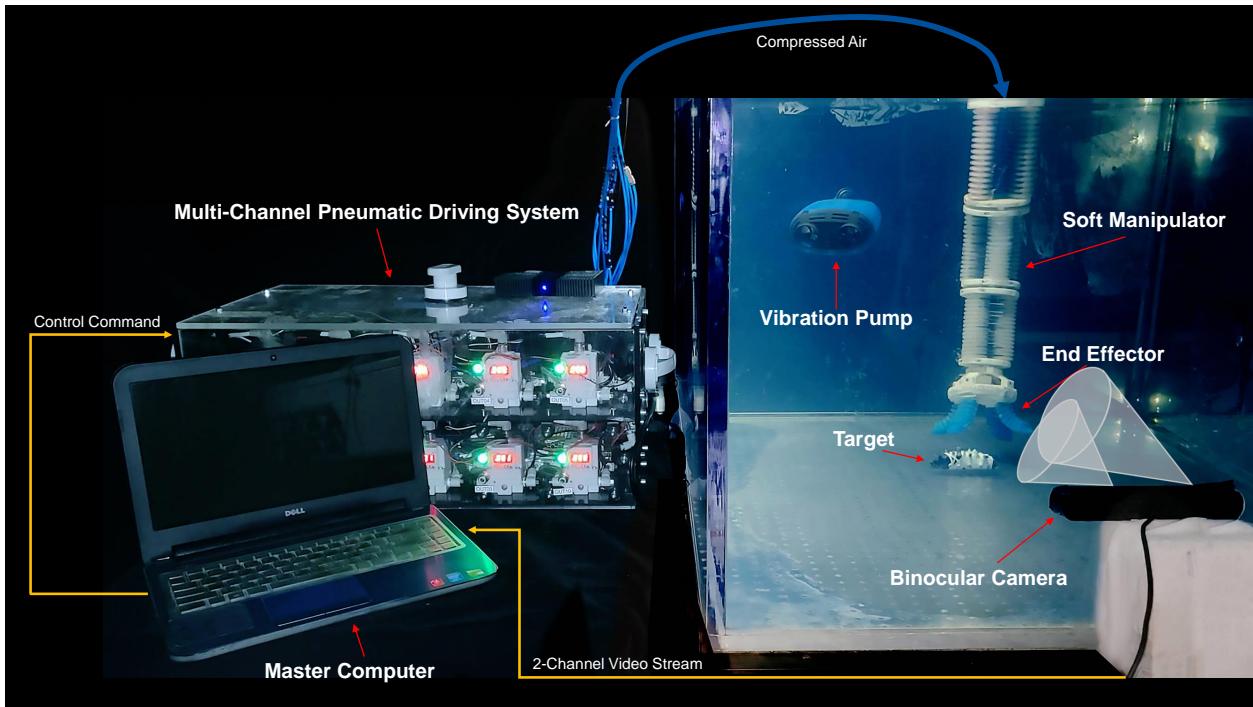


Figure 13: Early experiment setup at lab tank: the system consisted of the OBSS soft manipulator, the pneumatic driving system, a computer, the ZED camera, a vibration pump and a light

### 3.2 Lab Pool Experiments

The second stage Experiments were done in a  $5m \times 7m \times 1.5m$  lab pool at China Academy of Sciences Institute of Automation. The system had been developed to as described in section 2 at this stage. The autonomous seafood grasping system passed both static water test and test

when random moderate current were generated by a human being at side, achieving high success rate (46 success out of 58 attempt). Figure 14 shows snapshots of each step of how the system successfully collected the model into its basket. How the I-AUV traverse the squared pool was shown in Figure 14 (e).

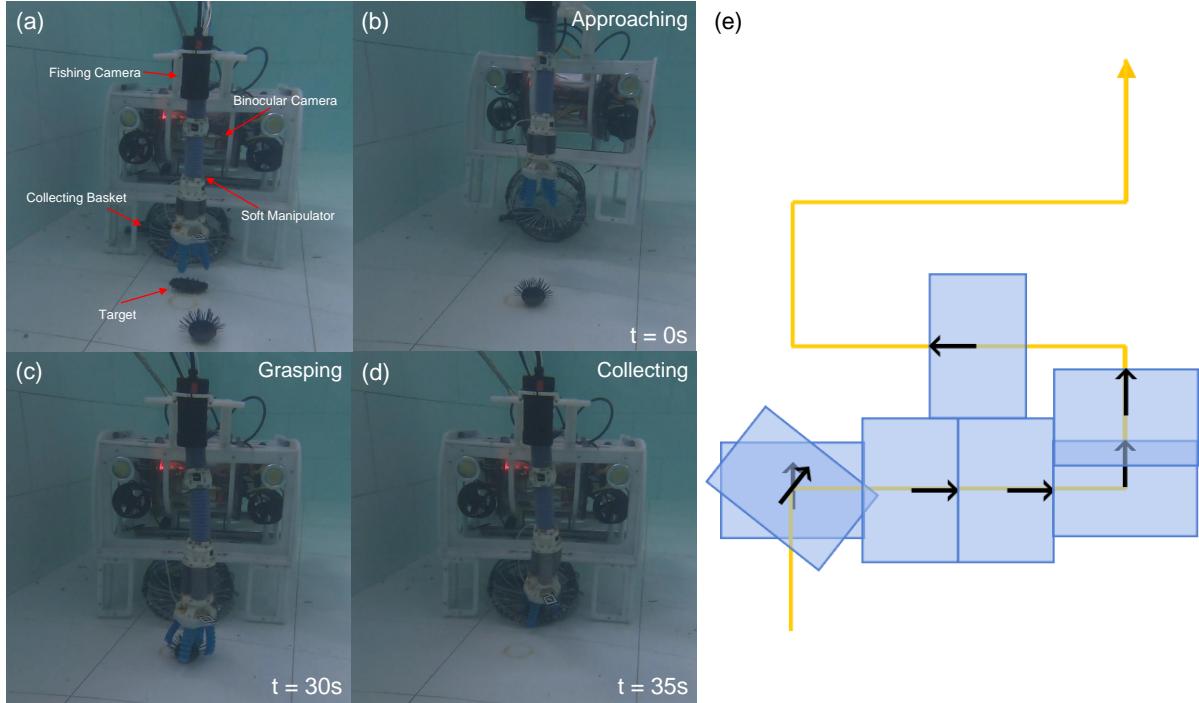


Figure 14: Grasping procedure within the lab pool and cruise route used: (a) A close-up view of the I-AUV. (b) The vehicle was aiming to the sea urchin model based on detected data automatically. (c) The vehicle was landed and the manipulator started to reach the model. (d) Target successfully collected into the basket. (e) Top view of the route used to traverse the pool as the blue rectangles represented the sight of the I-AUV.

### 3.3 Oceanic Experiments

The offshore experiments were taken place at the beachside of Jinshitan in Dalian, China. The results are shown below.

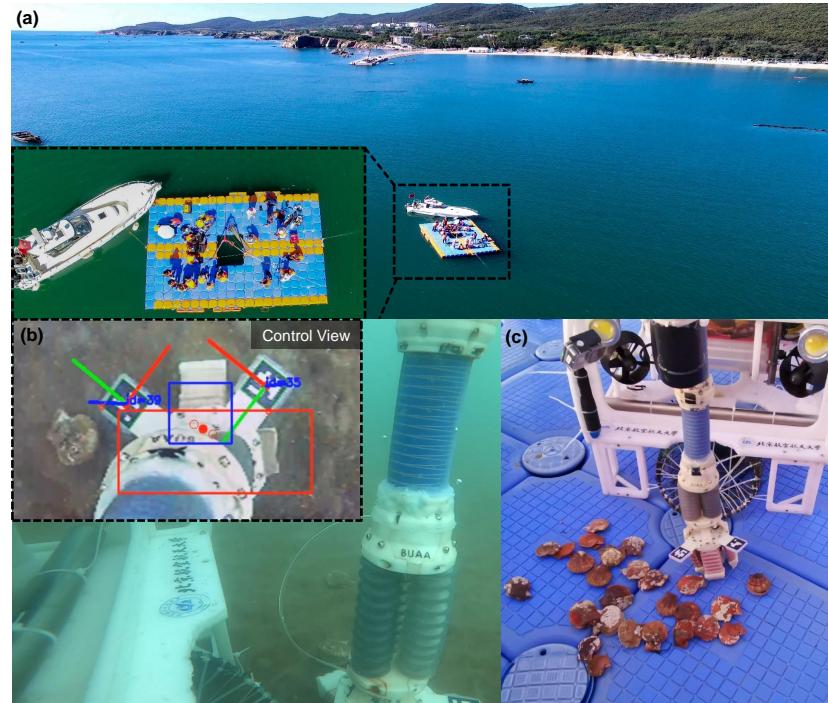


Figure 15: Snapshots of site of oceanic experiments: (a) The operating field of human beings (b) View from the master computer and fishing camera at the same time when grasping. (c) The harvest.



Figure 16: Grasping procedure captured at offshore sea bed

## 4 Discussion

In this article, an autonomous seafood grasping system was designed and tested. This system gained from techniques from a lot of fields such as soft robotics, autonomous control, computer vision, submarine vehicle and embedded system, which in total supported this system to harvest specific undersea species with a high precise, rapidly responsible, reasonably robust in complex natural submarine environment. The benefits of introducing soft robotics into undersea tasks is obvious:

- Compliance of the soft manipulator guarantees the integrity of fragile targets, and performs much more adaptive compared to traditional rigid robotic mechanism.
- As a result of materials used, the weight of the manipulator is relatively low, contributing low inertia while locomotion (weighted 1.05 kg while a normal rigid hydraulic manipulator could claim tens of kilograms).

Results from this article provides a promising solution for high-performance, rapid-responsible, low-cost underwater autonomous operation system for undersea missions in the future. undersea facility maintenance, seabed pollutants recycling, biological sampling are covered.

## 5 Conclusions

### 5.1 Contributions

The main contributions of this article are listed:

1. Design and construct a **I-AUV** system, and carry out a set of feasible, robust solution for autonomous seafood grasping and collecting in submarine, including target searching, target detecting, path planning, inverse kinematics algorithm and so on.
2. Explored how soft robotics could be developed and integrated on a modified **ROV** platform.
3. A simple, low-cost, high efficient solution to resist disturbance of currents in a complex underwater environment is proposed. With this solution, the robot could search the field in a short time.

### 5.2 Future Work

In this study, the design shows out to have character of high integration, high performance and high precise. However, compared to solutions with tradition rigid manipulator, this system is visibly less efficient, with slow response which is a notable disadvantage of soft actuators. The effect is especially obvious in this system because of the 50 meters long pneumatic pipes. In future works, the pneumatic driving system will be integrated to the vehicle and high pressure pipes capable of delivering pressure change in a very short time will be applied. Another improvement direction could be developing a large scale seafood searcher to provide general direction for the **I-AUV**.

## References

- [1] D. M. Barratt, P. G. Harch, and K. Van Meter, “Decompression illness in divers: A review of the literature,” *The Neurologist*, vol. 8, no. 3, pp. 186–202, 2002.
- [2] K. J. Åström and T. Hägglund, “Revisiting the ziegler–nichols step response method for pid control,” *Journal of process control*, vol. 14, no. 6, pp. 635–650, 2004.
- [3] M. Prats, J. C. Garcia, J. J. Fernandez, R. Marin, and P. J. Sanz, “Advances in the specification and execution of underwater autonomous manipulation tasks,” in *OCEANS 2011 IEEE - Spain*, 2011, pp. 1–5. DOI: [10.1109/Oceans-Spain.2011.6003619](https://doi.org/10.1109/Oceans-Spain.2011.6003619).
- [4] P. James and S. I. Siikavuopio, “Test of rov-based harvesting methods for sea urchins and scallops. part one: Report on sea urchin (kråkebolle) collection trial,” *Nofima rapportserie*, 2012.
- [5] MrAquaImage. “Careers in diving - sea urchin diving - interview with diver chris nelson.” (2013), [Online]. Available: <https://www.youtube.com/watch?v=0cqX-03fXoc> (visited on 04/18/2021).
- [6] C. Barblat, M. W. Dunnigan, and Y. Pétillot, “Dynamic coupling and control issues for a lightweight underwater vehicle manipulator system,” in *2014 Oceans-St. John's*, IEEE, 2014, pp. 1–6.
- [7] CCTV. “"haipengzi" on xiaochangshan island.” (2014), [Online]. Available: <https://tv.cctv.com/2014/01/17/VIDE1409105554118179.shtml> (visited on 04/18/2021).
- [8] G. Marani and J. Yuh, “Introduction to autonomous manipulation,” in *Case Study with an Underwater Robot SAUVIM*, volume 102 of *Springer Tracts in Advanced Robotics*, Springer, 2014.
- [9] D. Ribas, P. Ridao, A. Turetta, C. Melchiorri, G. Palli, J. J. Fernández, and P. J. Sanz, “I-auv mechatronics integration for the trident fp7 project,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2583–2592, 2015.
- [10] S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer, “Generation of fiducial marker dictionaries using mixed integer linear programming,” *Pattern Recognition*, vol. 51, pp. 481–491, 2016.
- [11] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, “Speeded up detection of squared fiducial markers,” *Image and vision Computing*, vol. 76, pp. 38–47, 2018.
- [12] X. Chen, J. Yu, S. Kong, Z. Wu, X. Fang, and L. Wen, “Towards real-time advancement of underwater visual quality with gan,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9350–9359, 2019.
- [13] X. Chen, J. Yu, and Z. Wu, “Temporally identity-aware ssd with attentional lstm,” *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2674–2686, 2019.

- [14] H. Qin, H. Chen, Y. Sun, and L. Chen, “Distributed finite-time fault-tolerant containment control for multiple ocean bottom flying node systems with error constraints,” *Ocean Engineering*, vol. 189, p. 106 341, 2019.
- [15] N. Corp. “Your gpu compute capability.” (2021), [Online]. Available: <https://developer.nvidia.com/cuda-gpus#compute> (visited on 04/22/2021).
- [16] Z. Gong, X. Fang, X. Chen, J. Cheng, Z. Xie, J. Liu, B. Chen, H. Yang, S. Kong, Y. Hao, *et al.*, “A soft manipulator for efficient delicate grasping in shallow water: Modeling, control, and real-world experiments,” *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 449–469, 2021.
- [17] ALIENTEK, *Stm32f407*. [Online]. Available: <http://www.alientek.com/>.
- [18] N. Corp., *Jetson agx xavier*, version P2888-0001 module, P2822-0000 carrier board. [Online]. Available: <https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit>.
- [19] S. Inc., *Zed*. [Online]. Available: <https://www.stereolabs.com/zed/>.

A wow



## B Hardware and Software Configurations Used for Computer Vision

### Main Configuration of *Jetson AGX Xavier* Used

<b>GPU</b>	512-core Volta GPU with Tensor Cores
<b>CPU</b>	8-core ARM v8.2 64-bit CPU, 8MB L2 + 4MB L3
<b>Memory</b>	32GB 256-Bit LPDDR4x   137GB/s
<b>Storage</b>	32GB eMMC 5.1
<b>DL Accelerator</b>	(2x) NVDLA Engines
<b>Vision Accelerator</b>	7-way VLIW Vision Processor
<b>Encoder/Decoder</b>	(2x) 4Kp60   HEVC/(2x) 4Kp60   12-Bit Support
<b>Size</b>	105 mm x 105 mm x 65 mm
<b>Deployment</b>	Module (Jetson AGX Xavier)

Table 4:

### Version of Libraries Used

CUDA	10.0
cuDNN	7.6.5
OpenCV	3.4.11 (with GTK)
PyTorch	1.1.0 (LibTorch)
ArUco	3.1.12

Table 5: