Unmotorize ROV gripper to catch profiling floats

Christophe Viel*

* CNRS, Lab-STICC, F-29806, Brest, France (e-mail: firstname.lastname@ensta-bretagne.fr).

ARTICLE INFO

Keywords:
Underwater robotics, Float, gripper, ROV

ABSTRACT

This paper proposes an ROV (Remotely Operated underwater Vehicle) gripper for retrieving cylindrical profiling floats. In order to fit several types of ROVs and to be easily installed, the gripper is not motorized, using only the robot's movements to catch the floats by simply moving forward on them. Then, the gripper uses the turnstile and freewheel concept to grasp and hold the float, with a safety release system to free an unwanted catch. The effectiveness of the clamp was tested in the pool and in the lake, with two actual rescues of lost floats. The limits of the methods are discussed.

1. Introduction

Profiling floats are a specific kind of Autonomous Underwater Vehicle (AUV) that can only regulate their depth. They are widely used in oceanography to measure data inside the water column like pressure, temperature, conductivity or biochemical. In the past two decades, the profiling float has allowed to collect and process of over two million vertical profiles of temperature and salinity from the global ocean [15, 14]: the Argo project [12, 15, 1] has deployed 4000 profiling float gathering data continuously all over the world. With the success of these missions, the data collection needs of oceanographers are increasing, leading to the development of new inexpensive floats [6, 2].

However, the floats can stop operating for several reasons [4]: the float becomes heavier than expected and can not reach the surface, software bug, sensors return wrong information, the float is captured by a fishery boat... During the development phase, software bugs are particularly frequent, increasing the risk of loss during the first immersions. Even after, the float can run out of electricity. In these cases, it can be complex to catch and bring the float to the surface, requiring the use of an ROV to rescue it.

Underwater robotic manipulation is an inherently difficult task and a slow process [10, 3]. Indeed, ROV pilots face problems such as lack of 3D information on arm position, poor visibility in murky waters, or significant delay in the information transmission [7]. Moreover, obtaining a stable position to operate the arm can be difficult due to the umbilical, water currents, and fluid resistance, especially if the ROV cannot be landed on the sea floor [3].

The shape of the gripper changes according to the object to be gripped and its characteristics (size, weight, fragility...). [10] describe an overview of a large variety of grippers and tools commonly used in underwater sampling for scientific purposes. Gripper with two or three-finger pliers can be sufficient to catch a tube [13], provided they are large enough. However, these tools can be complex to build, and always requires an electrical or pneumatic connection to operate, and therefore cannot always be easily disassembled once the mission is finished. Other works like [8, 5] propose a universal soft membrane gripper to grasp irregularly shaped objects. A membrane allows it to conform to the shape of the object and a hydraulic circuit is used to compress the object with a desired force. If this system is simpler than those used in classic hydraulic arm designs, it still requires equipping the ROV with a hydraulic system with a pump and an incompressible fluid.

In this paper, we propose a gripper for ROV to recover the float in the water column or on the seabed as long as it remains globally vertical. In order to offer a low-cost solution, simple to implement and adapted to a wide range of robots, the gripper is non-motorized to eliminate any electrical or pneumatic connections. Opposite to the gripper exposed below, our method uses the robot movements to activate the clamp and does not require an accurate positioning to catch the floats.An operator assistance is proposed to catch the float easily by centering the middle of the gripper with the float. The effectiveness of the clamp was tested at the pool and in the lake.

The concept of the proposed solution and its requirements are exposed in Section 2.1 and 2.2. The geometry of the gripper is described in Section 2.3 and the lock and unlock mechanism is detailed in Section 2.4. Assistance to the operator is proposed in Section 3. Section 4 presents experimental results of the gripper, and limits of the method are exposed in Section 5.

2. The gripper

2.1. Requirement

This paper design a gripper specific for profiling floats. We consider the following assumptions on the floats:

- the floats can be roughly assimilated by vertical tubes with a stop on them, generally a stability disk or a support platform for sensors (see examples in Figure 1). Name it "float stop" in this paper.
- The radius of the tube R is supposed to be known (measured before the dive for example).
- The floats maintain an approximately vertical orientation when submerged.

ORCID(s):

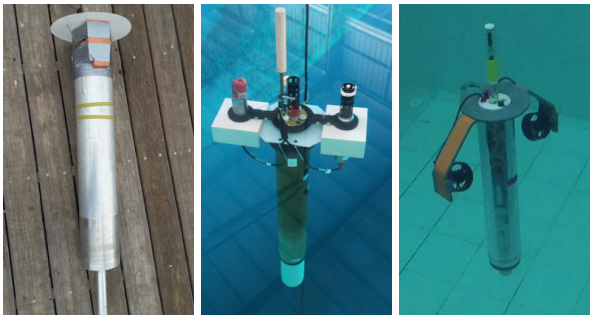


Figure 1: Example of profiling floats from Ifremer, the NAOS Project [1], or ENSTA Bretagne[6].

- The floats have a minimum weight, giving them a minimum inertia underwater.

The second condition is simple to fulfilled because the floats can be measured before their mission, and floats deployed on the same mission are generally identical ([12, 15, 1]). The third condition is not aberrant because the floats are balanced in this way and the majority of the failures are due to problems of exit of the float's piston, which modifies its overall buoyancy but not its balance: the floats sink until they rest vertically on the sea floor. In the following sections, the grippers are thus designed to grip vertical tubes with a stop in the middle or top.

Note that the problem of the localization of the float is not treated in this paper, which is focused on the gripper. In practice, the float can be located, for example, by fitting it with a USBL or by locating it using sonar by an operator.

2.2. The concept

In order to be adapted for several kinds of ROVs and be easily assembled and disassembled, we decide to develop an unmotorized grip to eliminate any electrical or pneumatic connections. The closing and opening of the gripper must therefore be activated by a movement of the ROV. Moreover, we desire the gripper can catch easily the floats for the operator.

The method chosen is the turnstile method. The clamp is composed of two arms. The end of the first arm is equipped at its extremity by a freewheel with several branches in the horizontal plane (see Figure 2) turning freely towards the inside of the clamp, and blocking towards the outside. The second arm is a simple rod that creates a closed space with the first arm.

Catching the float is very simple: the ROV moves towards the float by aiming between the two arms. The float pushes the branches of the turnstile, and cannot turn around anymore. By security, an unlock system is proposed to turn in the other side if the torque becomes too important, see Section 2.4. When the ROV ascent, the branches of the gripper come in contact with the float stop. The distance between the arms is kept loose, so the system does not require a lot of accuracy to catch the float. Note however that the float is just

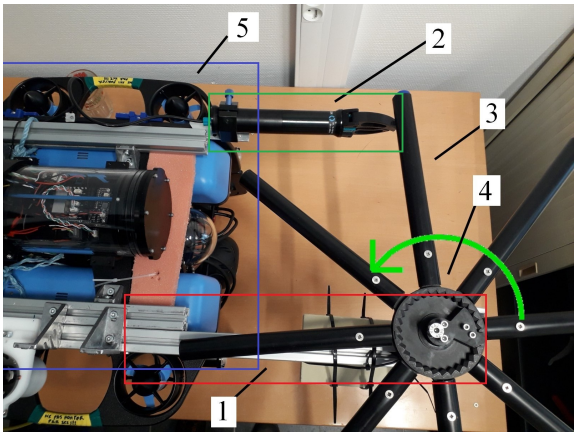


Figure 2: Turnstile gripper. 1: first arm with the freewheel. 2: second arm to close the space (here performed with a bluerov gripper already installed on the robot, but this gripper is not used in the strategy). 3 : branch of the turnside. 4: freewheel. 5: ROV.

locked and not tightened, and downward movement requires descending at the same speed as the float sink.

2.3. Gripper geometry

In the horizontal plane, the float can be assimilated to a circle C_1 of radius R. Let L be the gripper's branches length from the center of rotation C to its extremity, e be the thickness of the branches, D be the distance between the two arms of the gripper, as illustrated in Figure 3. The angle between the branches is defined as $\gamma = \frac{2\pi}{N}$, where N is the number of branches.

To guarantee the freewheel will trap easily the float no matter what the initial position of the wheel, a gripper with eight branches have been chosen. With fewer branches, the turnstile will be less easily dragged along by the float. With more branches, a more cumbersome clamp will have to be made, as it will be shown by the relation (3). L and D must also be chosen such that 1) the float can enter between two branches, 2) the float cannot slide between the branches and the second arm when it is inside the gripper.

As detailed in Appendix A.1, the float is locked by the gripper if the circle C_1 is inscribed inside the triangle performed by two branches and the second arms. The following relations can be obtained

$$L \geq l_1 \tag{1}$$

$$l_1 \leq D < l_2 \tag{2}$$

where the parameters l_1 and l_2 can be expressed as

$$l_1 = \frac{\left(1 + \frac{1}{\cos(\gamma)} + \tan(\gamma)\right)}{\tan(\gamma)} \left(R + \frac{e}{2}\right) \tag{3}$$

$$l_2 = 2R + L\cos(\gamma)\,. \tag{4}$$

where $\gamma = \frac{\pi}{4}$ in the case of eight branches. Since $\gamma = \frac{2\pi}{N}$, remark that the more branches there are, the longer L is, and so the clamp is cumbersome. Note also that the equation (1)

provides a minimum length of the branches' length L: the branches can be chosen longer than l_1 to better lock the float (thus reducing the length b_2 in Figure 3), but the gripper will be bulkier.

Moreover, the freewheel mechanism must not obstruct the float between the branches. Let C_R be the circle of radius R_c and center C, containing the freewheel's mechanism, see Figure 3. The maximum radius R_c that can take C_R without obstruct the gripper is $R_c \leq R_{c\,\max}$

$$R_{c\,\max} = \frac{\left(\frac{a+b+c}{2} - l_1 \tan(\gamma)\right)}{\cos\left(\frac{\gamma}{2}\right)} - R \tag{5}$$

with $a = l_1$, $b = \frac{a}{\cos(\gamma)}$ and $c = a\tan(\gamma)$. The calculations are detailed in Appendix A.2. The length of the plastic tubes attached to the freewheel in Figure 2 can be expressed as

$$L_b = L - R_c. \tag{6}$$

Between two teeth, the freewheel performs an angular step of $\delta = \frac{2\pi}{\mathbb{Z}}$, where $\mathbb{Z}$ is the number of teeth. At each position, the gap a_2 illustrated in Figure 3 increases/reduced of $\delta a_2 = L\left(1 - \cos\left(\frac{2\pi}{\mathbb{Z}}\right)\right)$. It is recommended to have as many teeth as possible to avoid releasing the float too quickly.

2.4. Lock and unlock security system

The gripper is locked with a freewheel mechanism. As illustrated in Figure 4, the freewheel is composed of three parts: a central fixed part with the axis of rotation (in green), a rotating gear where the branches are attached (in gray), and a pawl to allows the rotary motion in only one direction while preventing motion in the opposite direction (in red).

In a classic freewheel, the system can turn only in one direction. It can however be dangerous to not be able to release an undesirable item, mostly if this one is a fixed underwater item (example: a submarine pillar, a tree in a submerged lake, etc...). Thus, an unlock security system is proposed here. Since the gripper is unmotorized, the release mechanical uses only the movement of the robot.

First, a mechanical slack is added around the rotational axis to allow the rotation part to tilt and therefore disengage the system (see Figure 6). In classic operation, the rotation part is maintained at the horizontal in contact with the pawl by a spring (name it "lock spring"), see Figures 6 and 4. Secondly, in opposite to a classic freewheel where the teeth are vertically right, vertically inclined teeth are used here, see Figure 5. The angle teeth divide the torque received from the rotating part into vertical and horizontal forces. When the vertical force becomes stronger than the force applied by the lock spring, the pawl slides on the teeth, the freewheel is disengaged and can turn in both directions. Thus, when too much torque is applied to the clamp, it unlocks to release its grip. The gripper can thus be opened by a strong backward push if the object is fixed, or by a strong backward acceleration performed by the robot.

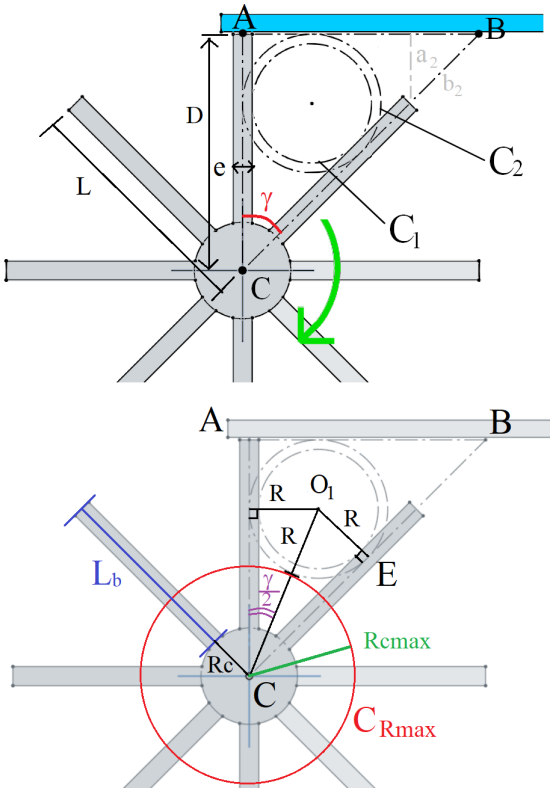


Figure 3: The general shape of the gripper with 8 branches. In gray, the first arm with the freewheel. In blue, second arm. Here, the length of the branches L are equal to the distance between the two arms D. a_2 and b_2 illustrate the gap between the branch and the second arm. C_1 correspond to the diameter of the float, C_2 to the diameter of C_1 plus the thicker the branches e. $C_{R\,\max}$ is the circle of radius $R_{c\,\max}$, center in C and tangent to the circle C_2. Green arrow: rotation side.

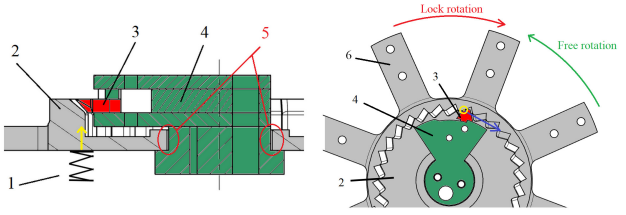


Figure 4: Freewheel system with unlocking security. 1: spring. 2: rotating part (grey). 3: pawl to lock the freewheel (red). 4: Fixed part (green). 5: mechanical slack. 6: branches of the gripper. A second spring not illustrated here allows keeping the pawl 3 in contact with the rotating part 2. Blue arrow: the horizontal force applied on the pawl made by the torque. Yellow arrow: vertical torque applied by the spring

A simple estimation of the unlocking torque C_{open} can be performed:

$$C_{\text{open}} = \frac{L F_{\text{spring}}}{\tan(\alpha)} \tag{7}$$

where L is the branches length, F_{spring} is the force of the unlock spring, α is the vertical inclination performed by the

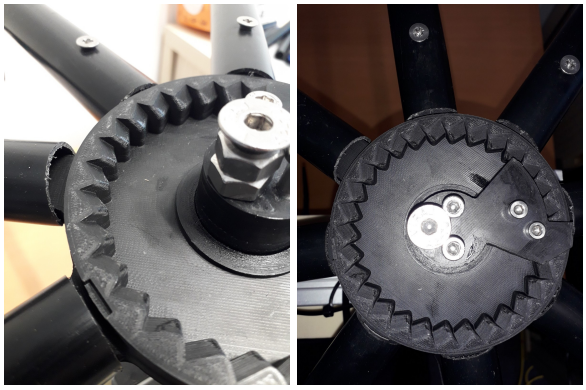


Figure 5: Freewheel with angled teeth.

Figure 6: Left: lock gripper. Right: unlock gripper when the torque becomes too high. 1: spring (inside a guide tube) to maintain the rotating part at the horizontal in contact with the pawl when the gripper is locked.

teeth such that $\alpha = 0$ corresponds to vertical teeth and $\alpha = \frac{\pi}{2}$ corresponds to "horizontal" teeth (no teeth). One observes that the more the teeth are inclined, the easier the freewheel will open.

Note that (7) does not consider friction on the rotation axis or the teeth. In practice, the friction on the rotation axis induces a minimum torque required to turn the freewheel in classic operation, so the floats have a minimum inertia to create a torque when the ROV moves to the float, as said in Section 2.1. Moreover, the contact between the lock spring and the rotating part adds friction, and so increases the minimal torque to turn the freewheel.

Since the inclined teeth can be difficult to machine, we perform it with a 3D printer. To limit the friction, the rotational axis has been greased and the contact of the lock spring has been made by a ball to obtain a point-contact junction.

2.5. Comments on gripper

2.5.1. Motorized gripper

A choice made here was to unmotorized the gripper, to make it easier to install on a ROV. Moreover, the unlock mechanism security induces a strong push back can release the gripper if necessary. These two choices imply that a tight fit of the float is not possible. Eventually, a second stop could be added to the float so that the clamp can catch the float between the two stops, and thus obtain a better float hold.

However, if we consider the case where the rotation of the turnstile is controlled by a motor, then the float can be tightened and released without any problem. The addition of a soft part on the side of the branches and on a stop inside the clamp would also enable the float to be held perfectly.

2.5.2. Curves and flexible branches

In a previous version of the gripper, curved branches were tested to try to better catch the float and turn the freewheel. The problem was that in order to obtain a curvature that provided real added value while maintaining the necessary spacing between two branches to catch the float, we needed either 1) fewer branches, making it harder to rotate the clamp in certain configurations, 2) or longer branches, making the clamp more cumbersome (which it already is).

Flexible branches could also be considered. However, since it is the float that activates the rotation of the gripper, flexible branches could make it more difficult to turn the freewheel. It could also increase the chances of the float escaping through gap between the two arms.

3. Catch the float: Assistance to the operator

As the float may have been lost in a very complex and cluttered environment, an assistance to the operator is proposed here instead of a complete automation. Note that an automatic movement towards the float could be added before the method described below.

To help the operator to catch the float, the ROV is controlled in depth and orientation to keep them center to the float when the assistance is activated. Then, the operator simply has to go straight ahead, so that the float arrives in front of the gripper and faces the turnstile.

3.1. Notations and assumptions

Let $\left(O, \vec{X}, \vec{Y}, \vec{Z}\right)$ be the global referential and $\left(O_R, \vec{X}_R, \vec{Y}_R, \vec{Z}_R\right)$ be the referential of the ROV and its camera, where O_R is the center of the ROV, $\vec{X}_R$ is the direction front the ROV, $\vec{Y}_R$ the lateral direction and $\vec{Z}_R$ its vertical direction corresponding to the vertical axis of the camera. Let's also define Z the ROV depth, θ the cap orientation of the ROV around axis $\vec{OZ}$, and ϕ its roll angle around axis $O\vec{X}_R$.

Consider the ROV can measure Z, θ and ϕ using for example a barometer, a compass and an IMU. Consider also that the ROV is sufficiently balanced (by static balancing or control) for its roll angle ϕ is maintained in an interval $I_\phi =]-\frac{\pi}{4}, \frac{\pi}{4}[$ (if not, note that the ROV cannot catch a vertical tube with the proposed gripper). At the horizontal position, the ROV can control its cap and depth with the inputs u_θ and u_Z. In practice, these inputs performs respectively a rotation and a translation around axis $O_R\vec{Z}$.

The middle of the gripper is centered with the camera (which is not necessary inside the gripper). Let's name "target" a point on the float where if the ROV keep it at the center of its camera, then the gripper will catch the float below its stop. For a defined target in the camera field, the coordinate $y_{\text{det}} \in [-1, 1]$ and $z_{\text{det}} \in [-1, 1]$ correspond respectively to the horizontal and vertical axis on the image, where $y_{\text{det}} = 1$ is the right edge of the image, $y_{\text{det}} = -1$ is the left edge,

$z_{\text{det}} = -1$ is the bottom of the image and $y_{\text{det}} = 1$ is the top. At $(0, 0)$, the target is centered.

3.2. Detection the target of the float

The target coordinates $(y_{\text{det}}, z_{\text{det}})$ can be measured in different ways:

- By detecting the target with a QRCode on the float using for example "cv2.QRCodeDetector" from openCV python library, then tack it;
- By detecting the target with a colored sticker/marker (preferably red/orange to distinguish from the sea main color) using a color filter, for example openCV python library "cv2.inRange" and "cv2.cvtColor" (see an example of algorithm in Appendix), then track it;
- Tracking a picture of the target taken by the operator during the dive (screenshot of the center of the ROV's camera at an instant t for example);
- Centering the target in the middle of the camera to get $(y_{\text{det}}, z_{\text{det}}) = (0, 0)$, then maintaining the cap θ and depth Z of the ROV constant, as described in the Section 3.3.

Note that since the three first methods are based on vision, they can become unstable when the target is too close to the camera, *i.e.* when the float has been caught by the gripper: in this case, the operator must deactivate manually the assistance. The second method is also subject to error when the color of the target is altered by distance and light conditions. To avoid this problem and obtain a lower computation time, the use of a tracking algorithm like tracker CSRT [9] can be preferred once the first detection has been made using one of the three first methods. Note also that a tracker can follow a target in an image with several objects that look like it and retain its identity. Moreover, it can still detect the target when its colors and shape are progressively alternated by the distance and light conditions: the new target detected is slightly different from the old one, allowing for example a progressive change from an orange to a gray target (see Figure 7).

3.3. Automation in orientation and depth

To position the ROV camera relative to the float, it needs control of its heading θ and depth Z to keep the buoy in the center of the camera.

If the target is tracked by the camera, $\left(y_{\text{det}}, z_{\text{det}}\right)$ are available and a visual control is performed to keep the target constantly at the center of the image, *i.e.* keep $\left(y_{\text{det}}, z_{\text{det}}\right)$ close to $(0, 0)$. Since the camera is affected by the roll, and so $\left(y_{\text{det}}, z_{\text{det}}\right)$, the roll angle ϕ is used in the control to try to keep the ROV as level as possible. The desired orientation θ_d and depth Z_d are defined as follow

$$\theta_d = \theta + k_y \cdot y_{\text{det}} \cos(\phi) \tag{8}$$

$$Z_d = Z - k_z z_{\text{det}} \cos(\phi) \tag{9}$$

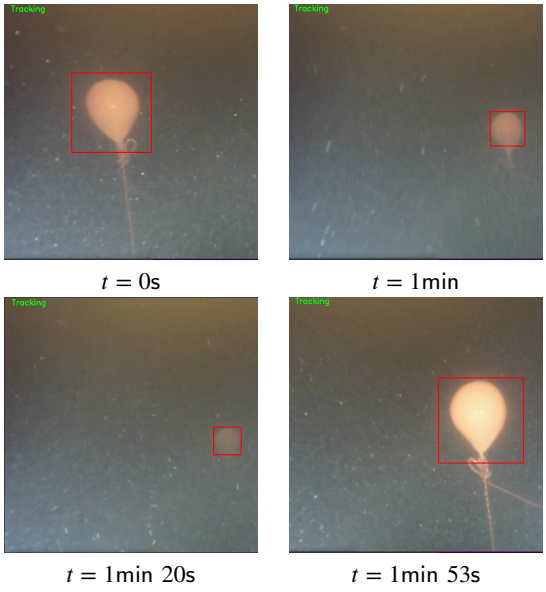


Figure 7: Evolution of the buoy's color with the distance in turbid water at the lake Guerledan (France). In the first three images, the buoy becomes darker with the distance. In the last image, the buoy is brighter due to the ROV's lights. However, by using only the first image for the color detection, the tracking algorithm allows to keep the buoy detected in all conditions.

with $k_y > 0$ and $k_z > 0$ design parameters. In case where there is no tracking strategy or the target is lost at an instant t_{lost}, the ROV maintains the last heading θ and depth Z received, *i.e.* $\theta_d(t) = \theta(t_{lost})$ and $Z_d(t) = Z(t_{lost})$ for $t > t_{lost}$. These parameters correspond to the last input asked by the operator in these directions if there is no tracking strategy.

Using $\left(\theta_d, Z_d\right)$, the inputs u_θ and u_Z can be expressed as

$$\begin{aligned} u_\theta =& k_{p,\theta} \cdot \tanh\left(k_{h,\theta} \text{sawtooth}\left(\varepsilon_\theta\right)\right) \\ &+ k_{d,\theta} \cdot \dot{\varepsilon}_\theta + k_{i,\theta} E_\theta \end{aligned} \tag{10}$$

$$u_Z = k_{p,z} \tanh\left(k_{h,z} \varepsilon_Z\right) + k_{d,z} \cdot \dot{\varepsilon}_Z + k_{i,z} E_Z \tag{11}$$

with $k_{p,\theta}$, $k_{d,\theta}$, $k_{h,\theta}$, $k_{i,\theta}$, $k_{p,z}$, $k_{d,z}$, $k_{h,z}$, $k_{i,z}$ are positive design parameters, the errors $\varepsilon_\theta = \theta - \theta_d$ and $\varepsilon_Z = Z - Z_d$, $\dot{\varepsilon}_\theta$ and $\dot{\varepsilon}_Z$ their derivatives, E_θ and E_Z the integration of the errors, and $\text{sawtooth}(x) = \mod((x + \pi), 2\pi) - \pi$ with $\mod(x, y)$ the modulo of x by y.

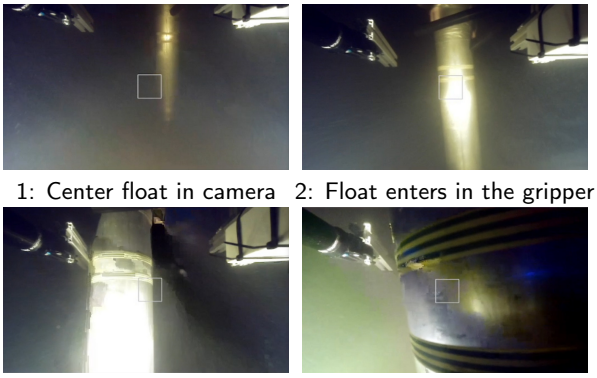
4. Experiments and floats rescues

4.1. Material

A BlueROV2[1] was used for the experiments, initially equipped with a frontal camera, two lights, a barometer and an IMU. The ROV has however been modified to add other instruments, like a sonar Blueview[2] to locate the float and a second camera facing downwards. The ROV was controlled

[1] https://bluerobotics.com/store/rov/bluerov2/
[2] http://www.teledynemarine.com/blueview/

Figure 8: turnstile gripper



1: Center float in camera 2: Float enters in the gripper

3: The gripper turns 4: Float trapped in the gripper.

Figure 9: Capture of a float with the gripper. First, the float is centered with the ROV's camera. Then, the ROV moves forward to enter the float in the gripper. The turnstile turns and the float is trapped in the gripper.

in depth and heading to keep them constant when the operator does not control them.

The gripper has been built using a 3D printer for the gear part and plastic pipes for the branches of the turnstile. We desire to catch a float of diameter $R = 125$mm. Due to the 3D printer's size, one takes $R_c = 100$mm and branches are made with plastic pipes of length $L_b = 250$mm and thickness $e = 20$mm: one has $L = L_b + R_c = 350$mm. The space between the arms made is $D = 250$mm, thus the branches have been chosen a little larger than the space D to better enclose the float. The number of teeth is $\mathbb{Z} = 30$ with an angle $\alpha = \frac{\pi}{4}$ has been chosen. It has been installed on the ROV so that the center of the clamp coincides with the center of the camera, so that the float hides the camera when it is grabbed, see Figure 8.

4.2. Results

The gripper was tested in a pool of size 3m×4m×3m and at the lake of Guerlédan in France, a former valley flooded by a dam that still has some immersed trees. A large number of tests were performed in the pool, four training tests were performed in the lake, and two true float rescues were performed in the lake at a depth of 39m (first float in Figure 1).

Except for the QRCode detection, all the assistance modes were tested in the pool and in the lake. Since the two rescues were performed on floats different than the ones used in our laboratory, these ones were not equipped with color markers: the assistance of maintaining cap and depth after the operator centered the float in the ROV's camera was used in these cases.

The most difficult part was to find the float inside the lake with the sonar and an approximate GPS location of the lost float. Then, since the ROV was controlled in depth and heading, the operator just needed to center the float in the camera, then moved forward to enter the float in the gripper. The gripper turned and the float was trapped in the gripper, see Figure 9. The ROV ascended then to come in contact with the float stop and bring the float to the surface.

During the two rescues, the floats were grabbed on the first try with the gripper. In the first case, the float was placed vertically on the sea floor without obstacles around it. In the second case, the float was trapped at the top of a tree, creating a most complex environment but still free of direct lateral obstacles. Paradoxically, the most complex cases were performed in the pool where the small environment dimension makes sometimes the seizure of the float impossible when this one is too close to a wall, and so the turnstile cannot turn enough to catch the float. A solution tested was to use a longer second arm with a hook to move the float away from the walls first, then move forward to catch it, but this method requires a much more cumbersome equipment and makes the maneuver much more complex.

The weakness of color detection is that when the water is turbid, color is only detected close to the target. Vision methods must also be deactivated when the target is too close to the camera (when the float is already in the clamp), otherwise the image moves too much and makes the behavior unstable. Data can be found in the previous work [11] where this method was already tested. The tracking of a sticker on the float was also tested. If it performed well in pool, the water of the lake was too cloudy to make it effective in the rescue condition: the centering of the ROV by the operator stays the most constant solution.

Videos of the two rescues can be found at `https://youtu.be/-IElz9GbFic` and `https://youtu.be/VpQA_6xuLfU`.

4.3. Discussion: gripper in presence of waves

In the sea, strong waves could be a problem for the clamp, as it doesn't block upward movement and a strong push back may unlock the gripper. However, the clamp is not unlocked if the ROV is positioned perpendicular to the waves, which will hold the float in the clamp without acting on the turnstile. Moreover, as floats are generally more than a meter long: it takes a lot of amplitude to get the float out of the clamp, and the ROV can remain submerged a few meters below the waves until it's close to the boat. Then, operators can recover the float on the way up, using for example a pole to grasp it by the float stops.

5. Limits of the gripper

If the proposed gripper has many advantages by its simplicity and the absence of motorization, the method has several limitations. First, it is adapted only to catch a vertical cylinder with a stop: it cannot catch a perfect cylinder or a horizontal cylinder. The float also requires to have minimal inertia to overcome the freewheel friction. It is also complex or impossible to catch a float against a wall: the use of a hook on a pole to move the float away from the wall is a possible solution. Finally, the geometry of the gripper makes it bulky.

6. Conclusion

This paper proposes a gripper for cylindrical floats with a vertical stop. The gripper is unmotorized, cheap, and simple to build and set up. The method uses the concept of turnstile and freewheel to trap the float without releasing it. The ROV displacements allow catching the floats by simply advancing on them, without an important accuracy. An unlocking system is proposed to release an undesirable underwater obstacle that could keep the robot stuck underwater. The effectiveness of the clamp has been tested in a pool and in a lake, to which are added two real rescues of lost floats. The method has however some limits, for example it can only catch vertical cylindrical item, and this one require a minimum inertia to turn the turnstile.

Future work will attempt to catch inclined or horizontal floats. We are also focusing on an automatic long-range float detection system, the most difficult part of the rescue being finding the float.

Acknowledgment

We acknowledge support from the Centre National de la Recherche Scientifique (CNRS) and Laboratoire des sciences et techniques de l'information, de la communication et de la connaissance (Lab-STICC). We acknowledge the center of resource of department STIC of ENSTA Bretagne. The author declares that there is no conflict of interest.

References

[1] X. André, P-Y Le Traon, S. Le Reste, V. Dutreuil, E. Leymarie, D. Malardé, C. Marec, J. Sagot, M. Amice, M. Babin, et al. Preparing the new phase of argo: technological developments on profiling floats in the naos project. *Frontiers in Marine Science*, 7:577446, 2020.

[2] Y. Bai, R. Hu, Y. Bi, C. Liu, Z. Zeng, and L. Lian. Design and depth control of a buoyancy-driven profiling float. *Sensors*, 22(7):2505, 2022.

[3] J. S. Cely, M. Á Pérez Bayas, M. Carpio, C. E. García Cena, A. Sintov, and R. Saltaren. Control strategy of an underactuated underwater drone-shape robot for grasping tasks. *Sensors*, 22(22):8828, 2022.

[4] MA Danchenkov and SC Riser. Profiling floats lost and caught in the japan sea. *Scientific Journal*, page 192, 2003.

[5] K. C. Galloway, K. P. Becker, B. Phillips, J. Kirby, S. Licht, D. Tchernov, R. J. Wood, and D. F. Gruber. Soft robotic grippers for biological sampling on deep reefs. *Soft robotics*, 3(1):23–33, 2016.

[6] T. Le Mézo, G. Le Maillot, T. Ropert, L. Jaulin, A. Ponte, and B. Zerr. Design and control of a low-cost autonomous profiling float. *Mechanics & Industry*, 21(5):512, 2020.

[7] J. Lemburg, P. Kampmann, and F. Kirchner. A small-scale actuator with passive-compliance for a fine-manipulation deep-sea manipulator. In *IEEE KONA OCEANS'11 MTS*, pages 1–4, 2011.

[8] S. Licht, E. Collins, D. Ballat-Durand, and M. Lopes-Mendes. Universal jamming grippers for deep-sea manipulation. In *In IEEE Monterey OCEANS 2016 MTS*, pages 1–5, 2016.

[9] A. Lukezic, T. Vojir, L. ˇCehovin Zajc, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017.

[10] A. Mazzeo, J. Aguzzi, M. Calisti, S. Canese, F. Vecchi, S. Stefanni, and M. Controzzi. Marine robotics for deep-sea specimen collection: A systematic review of underwater grippers. *Sensors*, 22(2):648, 2022.

[11] S. Prouten, H. Sabatier, E. Essono, M. Gounabou, and Ch. Viel. Autonomous ROV assistant providing remotelighting and vision to operated ROV. working paper, October 2022.

[12] S. C. Riser, H. J. Freeland, D. Roemmich, S. Wijffels, A. Troisi, M. Belbéoch, D. Gilbert, J. Xu, S. Pouliquen, A. Thresher, et al. Fifteen years of ocean observations with the global argo array. *Nature Climate Change*, 6(2):145–153, 2016.

[13] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, et al. Autonomous underwater intervention: Experimental results of the maris project. *IEEE Journal of Oceanic Engineering*, 43(3):620–639, 2017.

[14] G. Voet, D. Quadfasel, K. A. Mork, and H. Søiland. The mid-depth circulation of the nordic seas derived from profiling float observations. *Tellus A: Dynamic Meteorology and Oceanography*, 62(4):516–529, 2010.

[15] A. PS Wong, S. E. Wijffels, S. C. Riser, S. Pouliquen, S. Hosoda, D. Roemmich, J. Gilson, G. C. Johnson, K. Martini, D. J. Murphy, et al. Argo data 1999–2019: Two million temperature-salinity profiles and subsurface velocity observations from a global array of profiling floats. *Frontiers in Marine Science*, 7:700, 2020.

A. Calculation parameters of the gripper

A.1. Calculation of the branch's length

Let define the rectangular triangle ABC with $\widehat{BAC} = \frac{\pi}{2}$ as defined in Figure 3. Let define $a = \text{AC}$, $b = \text{CB}$, $c = \text{AB}$ and the angle $\gamma = \widehat{ACB}$. The length a can be considered as the distance between the two arms, γ the angle between two branches and c a part of the second arm. Let define the circle C_1 of radius R as the section of the float, and consider the circle C_2 of radius $R_2 = R + \frac{e}{2}$ to assimilate the arms to straight lines. The float is locked by the gripper only if the circle C_2 is inscribed inside the triangle ABC: let's find the adapted value of a.

The radius of the inscribed circle can be expressed as

$$R_2 = \frac{2S}{a+b+c} \tag{12}$$

where S is the area of the triangle, which can be expressed from the law of sinus as

$$S = \frac{1}{2}ab\sin(\gamma) \tag{13}$$

Since ABC is a rectangular triangle, one also has $b = \frac{a}{\cos(\gamma)}$ and $c = a\tan(\gamma)$. From these property, and (12)-(13), one gets

$$R_2 = \frac{a\frac{a}{\cos(\gamma)}\sin(\gamma)}{a + \frac{a}{\cos(\gamma)} + a\tan(\gamma)}$$

$$R_2 = \frac{a\tan(\gamma)}{1+\frac{1}{\cos(\gamma)}+\tan(\gamma)}$$
$$a = \frac{1+\frac{1}{\cos(\gamma)}+\tan(\gamma)}{\tan(\gamma)} R_2. \quad (14)$$

In our case, $D = a$ and $L \geq a$, thus (14) provides the minimum length of the branches.

Notice that since $a < b$, there is a gap between the branch of CB and the second arm when $L \in [a, \frac{a}{\cos(\gamma)}]$, illustrated by a_2 and b_2 in Figure 3. The projection of this gap on the axis AC, noted l_2, can be evaluated as

$$\begin{aligned} a_2 &= a - L\cos(\gamma) \\ &= D - L\cos(\gamma). \end{aligned} \quad (15)$$

To guarantee the float cannot pass by this gap, the distance D must also respect the following condition:

$$\begin{aligned} a_2 &< 2R \\ D - L\cos(\gamma) &< 2R \\ D &< 2R + L\cos(\gamma) \end{aligned} \quad (16)$$

A.2. Calculation $R_{c\max}$

Let's define O_1 the center of the circle C_1, as illustrate in Figure 3. We desire the circle C_R does not intersect with the circle C_1. Thus, one must have $\|CO_1\| \geq R + R_{c\max}$. Let's define the rectangular triangle O_1CE. One has

$$\begin{aligned} \|CO_1\| \cos\left(\frac{\gamma}{2}\right) &= \|CE\| \\ \left(R + R_{c\max}\right) \cos\left(\frac{\gamma}{2}\right) &= \|CE\| \\ R_{c\max} &= \frac{\|CE\|}{\cos\left(\frac{\gamma}{2}\right)} - R \end{aligned} \quad (17)$$

Let's find known the value of $\|CE\|$. Using the formula of Héron, one may write $\|CE\| = \frac{a+b+c}{2} - \|AB\|$, with $\|AB\| = \tan(\gamma)\|AC\|$ and $\|AC\| = l_1$ as expressed in Appendix A.1.

B. Detection target using HSV vision

To detect the target regardless of float orientation, an example of target is an orange band around the float, if possible of height equal to float diameter. Thus, the target seen by the camera in 2D will be an orange rectangle or even a square.

An algorithm based on HSV vision is performed, based on OpenCV2 tools. First, the image received from the ROV's camera is filtered to blur it and so smooth the details. Then, a color HSV filter is applied to keep only the target's color of the image. Small particles are removed using erosion and dilatation masks: one gets a mask of the image with the pixels in the selected color in white and the others in black. From this mask, the contours of the detected shapes are calculated. Since the target is an orange rectangle, a rectangular contour is sought. If at least one contour is detected, the largest contour is kept. Then, two parameters are evaluated: the largest side $d = max(h, w)$ including its contour, and the proportion of white pixels in this circle P. If R and P are larger than minimum values $R_{\min}$ and $P_{\min}$, the algorithm 1 returns the coordinate $(y_{\det}, z_{\det}) \in [-1, 1]^2$ and the size of the box including the target in the image. If no contour is detected or d or P are too small, the algorithm stops and returns that there is no detection.

Algorithm 1 Detection orange circle

Require: $Image$, HSV_color_limits, $d_{\min}$, $P_{\min}$
  $Target\_box \leftarrow None$
  $Mask \leftarrow BlurFilter(Image)$
  $Mask \leftarrow ColorFilter(Mask, HSV\_color\_limits)$
  $Mask \leftarrow Erosions(Mask)$
  $Mask \leftarrow Dilations(Mask)$
  $Contours \leftarrow FindContours(Mask)$
  if $Contours$ not empty **then**
    $Contour \leftarrow LargestContour(Contours)$
    $(y_{\det}, z_{\det}, w, h) \leftarrow MinEnclosingRectangle(Contour)$
    $P \leftarrow ComputeProportion(Mask, y_{\det}, z_{\det}, w, h)$
    if $max(w, h) \geq d_{\min}$ and $P \geq P_{\min}$ **then**
      $Target\_box \leftarrow y_{\det}, z_{\det}, w, h$
    end if
  end if
  return $Target\_box$

The functions used in the Algorithm 1 can be realized by using the following OpenCV functions in python3:

- BlurFilter() : cv2.GaussianBlur()
- ColorFilter() : cv2.inRange(hsv, hsvLower, hsvUpper) where hsv is obtained using the function cv2.cvtColor()
- Erosions() : cv2.erode()
- Dilations() : cv2.dilate()
- FindContours() : cv2.findContours()
- LargestContour() : c = max(cnts, key=cv2.contourArea) where cnts is the matrix find after using FindContours()
- MinEnclosingRectangle() : cv2.boundingRect(c)
- ComputeProportion() :

  # for a image named "frame" and the associated filter image named "mask" , one computes the coordinates of the minimum enclosing box and make sure the box is not outside the image

  1. ymin, ymax = max(0, int(y - w)), min(frame.shape[1], int(y + w))
  2. zmin, zmax = max(0, int(z - h)), min(frame.shape[0], int(z + h))

  # compute the proportion of detected pixels in the box

  1. number_of_white_pix = np.sum(mask[zmin:zmax, ymin:ymax] == 255)
  2. $P = \frac{\text{number\_of\_white\_pix}}{h*w}$
  3. return P