



**HAL**  
open science

## Quaternion based control for robotic observation of marine diversity

Silvain Louis, Lionel Lapierre, Karen Godary-Dejean, Yadpiroon Onmek,  
Thomas Claverie, Sebastien Villéger

► **To cite this version:**

Silvain Louis, Lionel Lapierre, Karen Godary-Dejean, Yadpiroon Onmek, Thomas Claverie, et al..  
Quaternion based control for robotic observation of marine diversity. OCEANS, Jun 2017, Aberdeen,  
United Kingdom. 10.1109/OCEANSE.2017.8085006 . lirmm-01588991

**HAL Id: lirmm-01588991**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01588991v1>**

Submitted on 18 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Quaternion based control for robotic observation of marine diversity

Silvain Louis,  
Lionel Lapierre,  
Yadpiroon Onmek and  
Karen Godary Dejean  
LIRMM, University of  
Montpellier, France  
{name}@lirmm.fr

Thomas Claverie  
CUFR Mayotte, University  
of Montpellier, France  
thomas.claverie@univ-mayotte.fr

Sebastien Villéger  
MARBEC, University of  
Montpellier, France  
sebastien.villeger@cnrs.fr

## I. CONTEXT

### A. Marine biodiversity assessment

Marine biodiversity is increasingly affected by human activities such as climate change, habitat destruction and fishing. In this context a key challenge for marine biologists is to be able to monitor biodiversity at large scale and high temporal frequency. However, underwater observations of marine biodiversity based on free or SCUBA diving are practically limited in terms of maximal depth and duration of diving because of physiological and technical constraints. In this context, the use of an underwater robotics system has an evident interest, but requires a detailed analysis of the expert protocols, in order to guarantee that the robotic system is in synergy with the expert needs. In this paper, we present 3 different functioning modes for the robotic system that we have established with our biologist partners from the Marbec<sup>1</sup> laboratory. The robotic aspect is handled by the LIRMM<sup>2</sup> laboratory and its underwater division, the Explore team<sup>3</sup>. This paper is structured to present the biological and robotic context first. Then, a generic control will be demonstrated. Then, we present our control structure where the control is applied following the 3 modes. Finally, we present simulation results.

### B. Biologist's protocols

In accordance with our biologist partners, we have defined 3 main observation protocols (Fig. 1), required by most marine biodiversity assessments : Transect, Localized Observation and Species Tracking. This paper presents 2 of these modes (biologically and their transpositions in robotics) and their associated control laws. It also presents the results of the simulation of the 2 modes. The Species Tracking mode is a combination of the first 2 modes.

Transect [21], [2], [22] is used to study the fish community in a given area. First, a reference is virtually defined and the diver mission is to swim along this reference at a constant

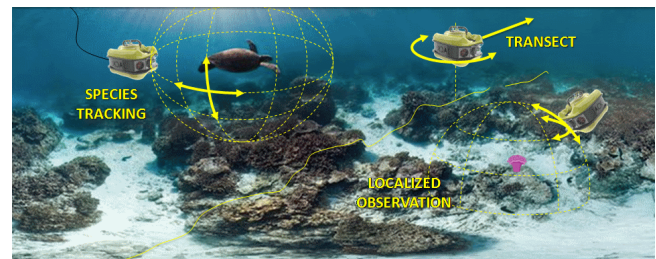


Fig. 1. Observation protocols

depth, and to count and identify all the fishes visible within a given distance [19]. To execute a robotic transect, the robot must be able to autonomously follow the reference, and to acquire pertinent information on the environment to allow the specialist to proceed to the counting and identification of species. Typically, these information are video streams. From a robotics point of view, we define this situation as co-control where some parts on the robotic system is under autonomous control (situation w.r.t the transect's reference), while the mission progression and the direction of observation is left to the operator control.

Localized Observation is the mode where the observation of a static feature (e.g. coral heads) is required. The diver mobility allows to investigate this region of interest with different angles of view. The transposition of this objective within the robotic system implies again a co-control situation, where the user has to be able to change or modify the direction and distance of observation, while the system autonomously centers the interesting feature in the sensors field of view (e.g. video or acoustic camera), and insures its own safety (avoiding collision with environment).

These 2 different protocols are converted into 2 different robotic modes. The mobility of the required system imposes to get rid of the classic gimble lock singularity [8], due to the Newton's angle representation. Hence a Quaternion based formalism has been chosen.

<sup>1</sup><http://www.umr-marbec.fr>

<sup>2</sup><http://www.lirmm.fr>

<sup>3</sup><http://explore.lirmm.fr>

### C. The Robotic System

The system we use is based on a robot ROV 2, which embeds 12 thrusters, a front-looking video camera, a front-looking acoustic camera, a down looking Doppler log and the classic suite of sensors such IMU, temperature or water entrance detectors. It also carries batteries allowing full energetic autonomy. An optical fiber links the system with the operator, allowing for full duplex data transmission.

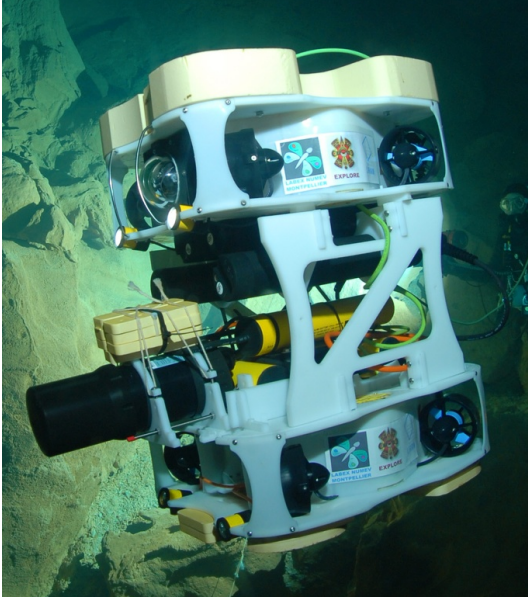


Fig. 2. the experimentator ROV

Considering equations (1) and (2), the dynamic model of the ROV is expressed in (3) with  $M$  the added mass,  $D$  the coefficient dumping,  $C$  the hydrodynamic coupling and  $G$  the hydrostatic phenomena [5].

$$\mathbf{v}_b = [u, v, w, p, q, r]^T \quad (1)$$

$$\mathbf{F}_b = [F_u, F_v, F_w, \Gamma_p, \Gamma_q, \Gamma_r]^T \quad (2)$$

$$\mathbf{F}_b = M \cdot \dot{\mathbf{v}}_b + D(\mathbf{v}_b) \cdot \mathbf{v}_b + C(\mathbf{v}_b) + G(x, y, z, \mathbf{Q}_b) \quad (3)$$

Where  $u, v, w, p, q, r$  are respectively the velocities and the rotations speeds along the axes  $x, y, z$  of the robot. Forces and torques are denoted by  $F_u, F_v, F_w, \Gamma_p, \Gamma_q, \Gamma_r$ .

We will not present in this paper the problematics related to the dispatcher (thrust allocation).

### D. Attitude control : State of the art

The analysis of the different robotic modes exposed before allows decoupling the question of the attitude control from the linear movement. Different formalisms exist for the attitude representation. The usual matrix representation [17] uses Euler angles (yaw, pitch and roll) to express the system attitude. The evident advantage is the direct comprehensive interpretation by human, but this representation faces the known singularity called 'gimbal lock', which expresses the fact that when pitch is  $\pm 90^\circ$ , then yaw angle is no more defined. Nevertheless, for

small variation of pitch and roll angle around zero (especially on ground), it is possible to design attitude control using Euler angles [4], or Rodrigues angles [13] [4] [10].

Another attitude representation formalism is based on Quaternions formalism. The advantage is that the previous singularity disappears. Quaternion based feedback control has been studied in the literature [20], [16], [3], [1]. System dynamics is considered with backstepping approach and non-linear feedback [14], [6], [15], [4], [9].

An interesting point that differentiates the previous solutions from each other is the way to calculate the attitude error function. Indeed the control design requires an expression for the error function, from which convergence to zero will be provided by the control [12], [23]. Two options are reported in the literature : i) attitude quaternion can be considered as normalized 4D vectors and error function can be expressed as the difference between current and desired quaternions [4]. The problem is that this vector difference is no more a normalized quaternion; and ii) the quaternion difference as expressed in (4) remains on the unitary sphere (as done in [7], [14] and [11]), i.e. belongs to (SO3). In the sequel, we will use this formalism by decomposing the error quaternion in the error angle around unit rotation vector.

## II. GENERIC QUATERNION CONTROL

### A. Notation

Let us define these notations :

- $\mathbf{Q}$  : Attitude quaternion (4x1) vector
- $\mathbf{Q}_d$  : Desired attitude quaternion (4x1) vector
- $\boldsymbol{\omega}$  : Rotational velocity (expressed in the body frame) (3x1) vector
- $\dot{\mathbf{Q}}_d$  : Desired velocity quaternion (4x1) vector
- $\dot{\boldsymbol{\omega}}$  : Rotational acceleration (expressed in the body frame) (3x1) vector
- $\mathbf{K}, \mathbf{K}_2$  : Control gains
- $Q_{(w,x,y,z)}$  : Coordinates (w, x, y or z) of quaternion Q

Let  $\boldsymbol{\omega}_{(\cdot)}$  be a 3D rotational velocity vector (expressed in different frames), then  $\mathbf{W}_{(\cdot)}$  designs the associated imaginary quaternion as  $\mathbf{W}_{(\cdot)} = [0, \boldsymbol{\omega}_{(\cdot)}^T]$ . Let  $\mathbf{v}_{(\cdot)}$  be a 3D translational velocity vector (expressed in different frames), then  $\mathbf{V}_{(\cdot)}$  designs the associated imaginary quaternion as  $\mathbf{V}_{(\cdot)} = [0, \mathbf{v}_{(\cdot)}^T]$ .

### B. From error quaternion to angle and unit rotation vector

To perform the angle control, we need to calculate the angle and unit rotation vector from the error quaternion.

Let  $\mathbf{Q}$  designs the quaternion (and  $\mathbf{Q}_D$  the desired quaternion) corresponding to a rotation of an angle  $\alpha$  around the normalized vector  $\mathbf{n}$ .

$$\mathbf{Q} = \left( \cos \frac{\alpha}{2}, -\sin \frac{\alpha}{2} \mathbf{n} \right)$$

The error quaternion is defined by :

$$\mathbf{Q}_e = \mathbf{Q}_d^* \otimes \mathbf{Q} \quad (4)$$

where  $\otimes$  defines the quaternion multiplication.

If the error quaternion is not equal to the neutral quaternion ( $Q_e \neq [1, 0, 0, 0]$ ), the angle  $\alpha_e$  and unit rotational vector  $\mathbf{n}_e$  are defined as:

$$\alpha_e = 2\text{acos}(Q_{e(w)})$$

$$\mathbf{n}_e = \frac{1}{\sqrt{Q_{e(x)}^2 + Q_{e(y)}^2 + Q_{e(z)}^2}} [Q_{e(x)}, Q_{e(y)}, Q_{e(z)}]$$

Else, the rotational vector is not defined, so we consider an arbitrary axis  $\mathbf{n}_e = [1, 0, 0]^T$  and a null angle  $\alpha_e = 0$ .

### C. Stabilization control

The control objective here is to drive the system to a desired orientation with a desired rotational velocity.

We can now state the following proposition.

**Proposition 1** : The speed control (5) stabilizes the system to the desired attitude with a desired rotational velocity, i.e.  $Q$  converges to  $Q_d$  and  $\dot{Q}$  converges to  $\dot{Q}_d$ .

$$W = Q_e^* \otimes W_d \otimes Q_e + W_0 \quad (5)$$

with (6), (7),  $K$  strictly positive.

$$W_d = 2Q_d^* \otimes \dot{Q}_d \quad (6)$$

$$W_0 = [0, -K\alpha_e \mathbf{n}_e]^T \quad (7)$$

$$\omega_0 = -K\alpha_e \mathbf{n}_e \quad (8)$$

**Proof** :

First, differentiating the error quaternion (4) with (6):

$$\begin{aligned} \dot{Q}_e &= \dot{Q}_d^* \otimes Q + Q_d^* \otimes \dot{Q} \\ &= \dot{Q}_d^* \otimes Q \otimes (Q^* \otimes Q_d \otimes \dot{Q}_d^* \otimes Q + Q^* \otimes \dot{Q}) \\ &= Q_e \otimes (Q^* \otimes Q_d \otimes \dot{Q}_d^* + \frac{1}{2}W \otimes Q^*) \otimes Q \\ &= \frac{1}{2}(Q_e \otimes W - W_d \otimes Q_e) \end{aligned} \quad (9)$$

To demonstrate the stability we use the Lyapunov function candidate (10) with  $\Upsilon = [1, 0, 0, 0]^T$ :

$$L = \frac{1}{2}(\Upsilon - Q_e)^T \cdot (\Upsilon - Q_e) \quad (10)$$

where  $\cdot$  defines the dot product.

Differentiating (10) yields:

$$\dot{L} = -\dot{Q}_e^T \cdot \Upsilon + \dot{Q}_e^T \cdot Q_e \quad (11)$$

Because of  $\dot{Q}^T \cdot Q = 0$ , (11) become:

$$\dot{L} = -\dot{Q}_{e(w)} \quad (12)$$

Considering the control law (5), (9) yields:

$$\dot{Q}_e = \frac{1}{2}Q_e \otimes W_0 \quad (13)$$

By the term by term decomposition of quaternion multiplication and (7):

$$\begin{aligned} \dot{Q}_{e(w)} &= -\frac{1}{2}(Q_{e(x)} \cdot \omega_{0(x)} + Q_{e(y)} \cdot \omega_{0(y)} + Q_{e(z)} \cdot \omega_{0(z)}) \\ \dot{Q}_{e(w)} &= \frac{K}{2}\alpha_e \sin\left(\frac{\alpha_e}{2}\right) \end{aligned} \quad (14)$$

Considering (14), (11) and (7) for  $W_0$  yields:

$$\dot{L} = -\frac{K}{2}\alpha_e \sin\left(\frac{\alpha_e}{2}\right) \quad (15)$$

With  $K$  strictly positive and  $\alpha_e \in ]-\pi; \pi]$  thus  $\dot{L} < 0$ : the speed control converges hence  $\alpha_e$  converges to 0 and  $Q_e^* \otimes W_d \otimes Q_e$  is the feedforward velocity term. ■

Consider this velocity control  $\omega$  in (5) as a reference velocity  $\omega_{ref}$  to control acceleration.

**Proposition 2** : The acceleration control (16) stabilizes the system to the reference rotational velocity  $\omega_d = \omega_{ref}$  with a desired rotational acceleration  $\dot{\omega}_d$ .

$$\dot{\omega} = K_2(\omega_d - \omega) + \dot{\omega}_d \quad (16)$$

**Proof** :

To demonstrate the stability we use the second Lyapunov function candidate (17) :

$$L_2 = \frac{1}{2}(\omega_d - \omega)^T (\omega_d - \omega) \quad (17)$$

Differentiating (17) yields:

$$\dot{L}_2 = (\dot{\omega}_d - \dot{\omega})^T (\omega_d - \omega) \quad (18)$$

With (16), (18) become :

$$\dot{L}_2 = -K_2(\omega_d - \omega)^T (\omega_d - \omega) \quad (19)$$

Considering  $K_2$  strictly positive,  $\dot{L}_2 < 0$ , the control converges, i.e.  $\lim_{t \rightarrow +\infty} \omega = \omega_d$  uniformly. ■

Combining (16) with  $\omega_{ref}$  from (5), we can write the full control of our robotic system as :

$$\dot{W} = \dot{W}_d + K_2(Q_e^* \otimes \omega_d \otimes Q_e - \begin{bmatrix} 0 \\ K\alpha_e \mathbf{n}_e \end{bmatrix} - W) \quad (20)$$

Considering the dynamic model of our system (3), the actuation demand can be written as ( $\rho$  being disturbances and  $M_{pqr}$  partial matrix of  $M$  announced in I-C) according to a classic computed torque approach:

$$\begin{aligned} \Gamma_{pqr} &= M_{pqr} \cdot \dot{\omega}_d \\ &+ M_{pqr} \cdot K_2(Q_e^* \otimes \omega_d \otimes Q_e - K\alpha_e \mathbf{n}_e - \omega) \\ &+ \rho \end{aligned} \quad (21)$$

### III. TRANSPOSITION OF BIOLOGICAL PROTOCOLS TO ROBOTICS

#### A. Specifications

Initially, we have to translate the needs of biologists into robotic constraints. In view of the description of the Transect in I-B, we can enunciate these constraints:

- Constant depth or constant ground distance (possibly modified to respect the robot security constraint).
- Constant reference heading.
- Ability to change the deviation from the heading reference (equivalent to the rotation of the head of the diver).
- Fixed longitudinal forward velocity (with respect to hardware limitations of the actuators).
- Constant pitch.
- Constant roll.

The second mode is Localized Observation. It can be defined as follows:

- Rotation trajectory on the isosphere whose center is the interest point.
- Fixed distance to the interest point.
- Orientation to look at the interest point.

#### B. 2 functioning modes

To link the 2 different modes (Transect and Localized observation) to the generic quaternion control II, we define four bases  $\{B_B\}$ ,  $\{B_C\}$ ,  $\{B_C\}_{desired}$  and  $\{B_S\}$  shown Fig. 3.

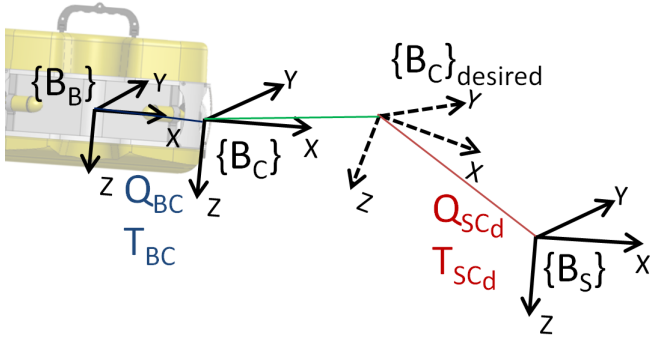


Fig. 3. 4 bases

The target frame  $\{B_S\}$  is a virtual frame projected on the interest point, according to the functioning mode. In Transect mode, the interest point is a virtual point which lives only on the reference and its forward velocity is driven by the operator ( $\{B_S\}$ -X axis is collinear with the transect and  $\{B_S\}$ -Z axis is vertically pointing down).

In Localized Observation, the frame interest point is the same as mode interest point.  $\{B_S\}$ -X axis and  $\{B_S\}$ -Y axis are coplanar to the sea floor and the  $\{B_S\}$ -Z axis is the normal to the plane.

The desired sensor frame  $\{B_C\}_{desired}$  is relative to the target frame, depending on the current functioning mode. The

transformation from target frame to desired sensor frame is denoted by the quaternion  $Q_{SCd}$  and a translation  $T_{SCd}$ .

The body frame  $\{B_B\}$  is attached to the robotic system. The quaternion  $Q_B$  represents the system's attitude with respect to the inertial frame  $\{B_O\}$ .

The real sensor frame  $\{B_C\}$  is attached to the observation system (front camera on Fig. 3). The transformation from body frame to sensor frame is defined by a rotation (Quaternion)  $Q_{BC}$  and a translation  $T_{BC}$ .

The objective of the control is to converge the real sensor frame  $\{B_C\}$  to desired sensor frame  $\{B_C\}_{desired}$ .

The control of the translations being specific to the functioning mode (sensors based or environment based), we will present them in the following section.

#### IV. TRANSECT MODE

In the transect mode, the translation movement is defined relatively to the reference, defined as an absolute path, or relative to an environmental feature.

Mode parameters are :

- Global initial position of transect  $x_{s_{init}}$ . These coordinates can be built from a desired GPS position and depth, or relative to a global map.
- Robot last known absolute position  $x_{b_{init}}$  before position estimation. it is often the last GPS position before immersion.
- Desired transformation between the robot  $\{B_s\}$  and  $\{B_c\}_{desired}$  frames, expressed as  $T_{SCd}$  and  $Q_{SCd}$ . The rotation possibilities (observation on the side of the transect) are expressed in this quaternion.
- Sensor position and attitude in the body frame,  $Q_{BC}$  and  $T_{BC}$ .
- Forward speed along the transect  $v_s = (\dot{s}, 0, 0)$ .  $s$  denotes the curvilinear position of  $\{B_s\}$  on transect.
- Transect attitude  $Q_s$

#### A. Control

The first step is to calculate the target frame attitude and position:

$$V_{s_0} = Q_s \otimes V_s \otimes Q_s^* \quad (22)$$

$$x_{s_0} = x_{s_{init}} + \int_0^t v_{s_0} dt \quad (23)$$

Then, we calculate the desired attitude of sensor frame (24), the attitude of real sensor frame (25) and the error quaternion (26):

$$Q_{cd} = Q_s \otimes Q_{SCd} \quad (24)$$

$$Q_c = Q_b \otimes Q_{BC} \quad (25)$$

$$Q_e = Q_{cd}^* \otimes Q_c \quad (26)$$

We now apply the control (20) to obtain the rotational acceleration command.

Afterwards, we perform the control in translation. For this, the robot is equipped by a Doppler log ( $v_{dopp} = [v_{x_{dopp}}, v_{y_{dopp}}, 0]^T$ ) and a depth meter ( $z_{depth}$ ). The Doppler log position in body frame is expressed in  $Q_{BDopp}$ .

We estimate the robot position :

$$\mathbf{V}_{b_0} = \mathbf{Q}_b \otimes \mathbf{Q}_{BDopp} \otimes \mathbf{V}_{dopp} \otimes \mathbf{Q}_{BDopp}^* \otimes \mathbf{Q}_b^* \quad (27)$$

$$\mathbf{x}_{b_0} = \mathbf{x}_{b_{init}} + \int_0^t \mathbf{v}_{b_0} dt \quad (28)$$

$$\mathbf{x}_{b(z)} = z_{depth} \quad (29)$$

We calculate the desired robot position :

$$\mathbf{x}_{bd} = \mathbf{x}_{s_0} + (\mathbf{Q}_s \otimes \mathbf{T}_{SCd} \otimes \mathbf{Q}_s^*) - (\mathbf{Q}_b \otimes \mathbf{T}_{BC} \otimes \mathbf{Q}_b^*) \quad (30)$$

We perform the velocity control, with target speed forward (22) and the centrifugal effect:

$$\begin{aligned} \mathbf{V}_{Bd_0} = & \begin{bmatrix} 0 \\ K_3(\mathbf{x}_{bd} - \mathbf{x}_{b_0}) \end{bmatrix} + \mathbf{V}_{s_0} \\ & + \mathbf{Q}_b \otimes \mathbf{Q}_{BC} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_c \times \mathbf{T}_{CB_c} \end{bmatrix} \otimes \mathbf{Q}_{BC}^* \otimes \mathbf{Q}_b^* \end{aligned} \quad (31)$$

Finally, we express the linear acceleration in the body frame to calculate the forces to apply on the robot through the dynamic model (see I-C).

$$\dot{\mathbf{V}}_b = \mathbf{Q}_b^* \otimes \begin{bmatrix} 0 \\ K_4(\mathbf{v}_{Bd_0} - \mathbf{v}_{b_0}) \end{bmatrix} \otimes \mathbf{Q}_b \quad (32)$$

### B. Simulation

The simulations are performed on a 3D simulator [18] with the exact inverse dynamic model. The restoring torque and force of buoyancy are not considered. The static simulation parameters are defined in Tab. I.

TABLE I  
SIMULATION PARAMETERS

$\mathbf{x}_{s_{init}} = [0, 0, 3]^T$	$\mathbf{x}_{b_{init}} = [0, 0, 0]^T$
$\mathbf{T}_{SCd} = [0, 0, -2]^T$	$\mathbf{Q}_{BC} = [1, 0, 0, 0]^T$
$K = 1$	$K_2 = 3$
$K_3 = 2$	$K_4 = 2$
$\mathbf{T}_{BC} = [0.25, 0, 0]^T$	$\mathbf{Q}_s = [0.9239, 0, 0, 0.3827]^T$ (flat with 45deg from North)

During the simulation, we changed the speed forward  $\dot{s}$  along the transect (waiting stabilization of the robot at the transect starting point and then starting transect at T=11s to T=54s) and the orientation of the camera  $\mathbf{Q}_{SCd}$ , viewing left side (T=20s to T=28s) then right side (T=36s to T=42s).

We observe the robot position error (Fig. 4) which changes every transition (beginning and end of the transect or orientation change) and which nullifies thanks to control.

We also observe the sensor attitude error (Fig. 5) through the angle error and error vector.

### V. LOCALIZED OBSERVATION MODE

In the localized observation mode, the translation control is a visual control to center the point of interest in the camera image. The distance between the interest point and the camera can be measured by a sensor or by visual analysis. Visual analysis is performed with the OpenCV library.

Mode parameters are :

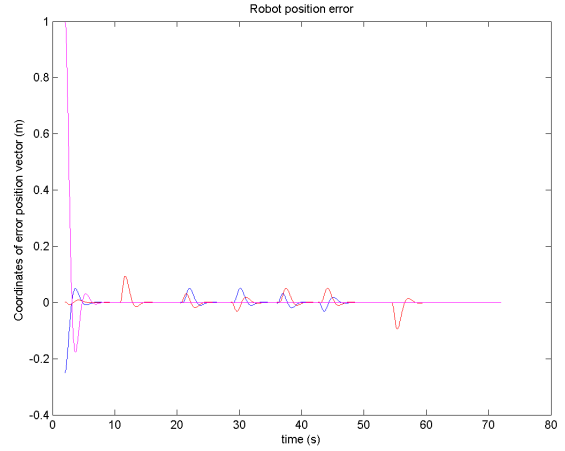


Fig. 4. 3 robot position errors along the axes  $x, y, z$

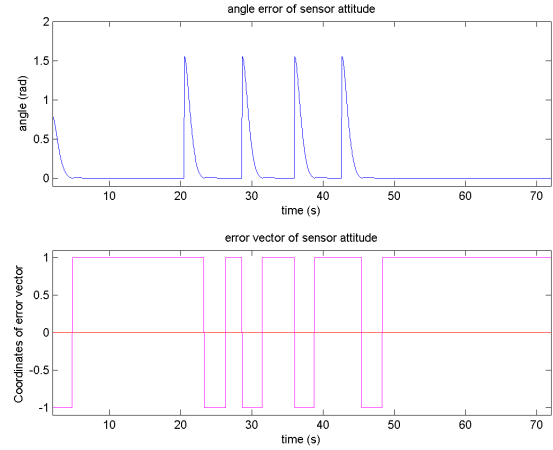


Fig. 5. Sensor attitude error

- Desired distance between the robot and the point of interest  $d_d$ .
- Attitude of the point of view  $\mathbf{Q}_{SCd}$
- Camera mechanical connection on the robot  $\mathbf{Q}_{BC}$  and  $\mathbf{T}_{BC}$ .

### A. Control

First, we rotate the reference  $\mathbf{Q}_{SCd}$  according to user commands  $\boldsymbol{\omega}_{joy}$ :

$$\mathbf{Q}_{SCd} = \int_0^t \frac{1}{2} \mathbf{Q}_{SCd} \mathbf{W}_{joy} dt \quad (33)$$

We normalize the quaternion (33) then we define the error quaternion:

$$\mathbf{Q}_e = \mathbf{Q}_{SCd}^* \otimes (\mathbf{Q}_b \otimes \mathbf{Q}_{BC}) \quad (34)$$

We perform the acceleration control as explained in II-C. The camera provides the following information: the 3D distance between the camera and the point of interest (X-axis

and Y-axis in camera image). We calculate the translational displacement necessary to center the buoy in the camera image taking into account the drift generated by the camera rotation.

$$\mathbf{v}_c = \begin{bmatrix} K_{3x}.dx_{camera} + \omega_{c_y}.d \\ K_{3y}.dy_{camera} + \omega_{c_x}.d \\ K_{3z}(d_d - d) \end{bmatrix} \quad (35)$$

Let's transfer the translation velocity to body frame (36), and apply the centrifugal effect (37):

$$\mathbf{V}_{c_b} = \mathbf{Q}_{BC} \otimes \mathbf{V}_c \otimes \mathbf{Q}_{BC}^* \quad (36)$$

$$\mathbf{v}_{Bd} = \mathbf{v}_{c_b} + \omega_{b_d} \times \mathbf{T}_{BC} \quad (37)$$

And perform the acceleration control:

$$\dot{\mathbf{v}}_b = K_4(\mathbf{v}_{Bd} - \mathbf{v}_b) \quad (38)$$

### B. Simulation

In the simulation, the point of interest is a yellow submerged buoy with known diameter (to calculate the Z-axis distance). The static simulation parameters are defined in Tab. II.

TABLE II  
SIMULATION PARAMETERS

$$\begin{array}{ll} \mathbf{T}_{BC} = [0.25, 0, 0]^T & \mathbf{Q}_{BC} = [1, 0, 0, 0]^T \\ K = 1 & K_2 = 3 \\ K_3 = 2 & K_4 = 5 \\ d_d = 2m & \end{array}$$

The simulation consists of multiple 3D movements around the buoy (see the attached video). We observe the sensor attitude error (Fig. 6) through the error vector and angle error.

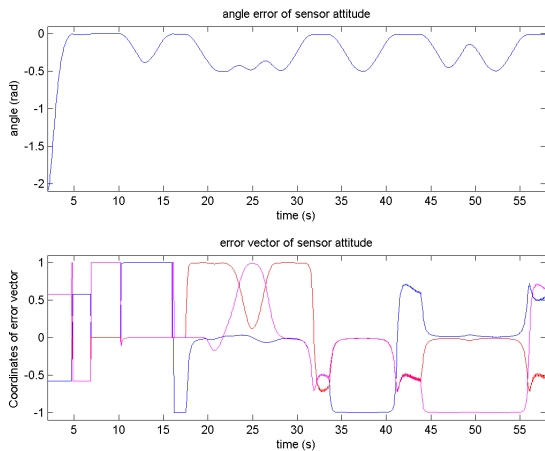


Fig. 6. Sensor attitude error

We also observe the distance error and the position error of the buoy in the camera image (Fig. 7) and the action of the control to minimize errors.

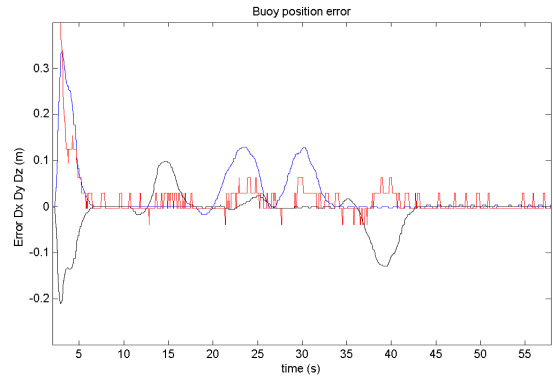


Fig. 7. Distance error

## VI. CONCLUSIONS

Our works are the basis of a collaboration between biologists and roboticists. The goal is to design a robotic system for the assessment of marine biodiversity. The challenge is to control the robot for three specific functioning modes extracted from biologist needs. For that, we defined a quaternion attitude control adapted for our needs and its variants depending on the mode. And we proved its convergence. Finally, we show that our solution (Transect and Localized Observation mode) is efficient and convergent, with a real time simulator. We also began local field experiments to test these modes in water.

The following of these works will include a mission to Mayotte to test the 3 functioning modes in real environmental constraints. We also plan to make use of the data gathered during the Mayotte mission (videos) to study the robot impact in the context of counting fishes. Then, the next step is to study and manage the switching between these functioning modes and the result on the system (e.g. control, software architecture). Modes switching causes modifications of the configuration set. We plan to manage it with linking functions, as well as specific mechanisms at the middleware and mission management levels.

## ACKNOWLEDGMENT

The authors want to express their grateful thanks to CUFR Mayotte<sup>4</sup> to finance these works and associated missions.

## REFERENCES

- [1] S. P. Bhat and D. S. Bernstein. Finite-Time Stability of Continuous Autonomous Systems. *SIAM Journal on Control and Optimization*, 38(3):751–766, 2000.
- [2] G. J. Edgar, N. S. Barrett, and A. J. Morton. Biases associated with the use of underwater visual census techniques to quantify the density and size-structure of fish populations. *Journal of Experimental Marine Biology and Ecology*, 308(2):269–290, 2004.
- [3] O. Egeland and J. Godhavn. Passivity based adaptive attitude control of a rigid spacecraft. *Automatic Control, IEEE Transactions on*, 39:842–846, 1994.
- [4] O. E. Fjellstad and T. I. Fossen. Position and attitude tracking of AUV's: a quaternion feedback approach. *IEEE Journal of Oceanic Engineering*, 19(4):512–518, 1994.
- [5] T. I. Fossen. Guidance and control of ocean vehicles, 1994.

<sup>4</sup><http://www.univ-mayotte.fr>

- [6] T. I. Fossen. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. 2002.
- [7] E. Fresk and G. Nikolakopoulos. Full Quaternion Based Attitude Control for a Quadrotor. *European Control Conference*, pages 3864–3869, 2013.
- [8] H. Goldstein. *Classical Mechanics*. 2nd edition, Ch. 5, 1980.
- [9] B. Graf. Quaternions and dynamics. arXiv:0811.2889, 2008.
- [10] Boskovic, D. M., Krstic, M. Global attitude / position regulation for underwater vehicles. *International Journal of Systems Science*, 1999.
- [11] H.-Y. Kim and C. A. Woolsey. Global Directional Control of a Slender Autonomous Underwater Vehicle. *Journal of Guidance, Control, and Dynamics*, 30(1):255–259, 2007.
- [12] M. Kleppmann. Simulation of colliding constrained rigid bodies. University of Cambridge, Computer Laboratory, 2007.
- [13] V. Koshlyakov. Generalized euler equations in Quaternions. *Ukrainian Mathematical Journal*, 46(10):1414–1416, 1994.
- [14] R. Kristiansen, P. J. Nicklasson, and J. T. Gravdahl. Satellite attitude control by quaternion-based backstepping. *IEEE Transactions on Control Systems Technology*, 17(1):227–232, 2009.
- [15] M. Krstic. Nonlinear and adaptive control. Technical report, 2006.
- [16] R. Leverage, R. Marcos, and S. Ryan. Attitude Control Systems. page 62, 2008.
- [17] Raptis, Ioannis A and Valavanis, Kimon P and Moreno, Wilfrido A A novel nonlinear backstepping controller design for helicopters using the rotation matrix. *IEEE Transactions on Control Systems Technology* 2011
- [18] S. Louis, D. Andreu, K. Godary, and L. Lapiere. HIL Simulator for AUV with ContrACT. CAR 2015.
- [19] D. Mallet and D. Pellerier. Archimer Underwater video techniques for observing coastal marine biodiversity : A review of sixty years of publications ( 1952 2012 ). *Fisheries Research*, 154(June):44–62, 2014.
- [20] T. Minh-Duc Hua, P. M. Hamel, and C. Samson. Introduction to Feedback Control of Underactuated VTOL Vehicles. *IEEE Control systems magazine*, (february):61–75, 2013.
- [21] M. A. Samoilys and G. Carlos. Determining methods of under water visual census for estimating the abundance of coral reef fishes. *Environmental Biology of Fishes*, 57:289–304, 2000.
- [22] R. A. Watson and T. J. Quinn. Performance of transect and point count underwater visual census methods. *Ecological Modelling*, 104(1):103–112, 1997.
- [23] S. a. Whitmore. Closed-Form Integrator for the Quaternion (Euler Angle) Kinematics Equations. *Ntrs.Nasa.Gov*, 2000.