

A global control scheme for free-floating vehicle-manipulators

Olivier Kermorgant, Yvan Pétillot, Matt Dunnigan

Abstract—This paper proposes a general framework to control underwater vehicle-manipulator systems from external sensors. The design of the control law follows several constraints and criteria to optimize the overall behavior. The task is defined in the sensor space, that is suited for visual servoing from camera or sonar images. The control input distribution between the arm and the body is considered and separated between the approach and the intervention phases. The proposed framework is validated in a realistic simulation environment, imposing constant and varying disturbances at the velocity level.

Index Terms—Mobile manipulation, visual servoing, underwater robots, disturbances rejection

I. INTRODUCTION

A strong trend in underwater robotics is the use of autonomous underwater vehicles instead of the classical Remotely Operated Vehicles and manned submersibles. Risk and cost are highly reduced, as it is never easy to deploy a team on a surface vessel or operators in a submersible. The so-called intervention autonomous underwater vehicles (i-auv) represent a recent class of auv's, that usually have a robot arm and are designed to perform various tasks underwater. We focus on the intervention tasks, that are related to the positioning with regards to an object (that may be free-floating or not) or with regards to underwater structures such as pipelines. Manipulation in the underwater environment is indeed difficult but has been recently demonstrated in the US [10], [11] and Europe [3], [15]. The issue of sensing is a critical one but recent progress has been made on motion and pose estimation from both Video and Sonar Imagery [2], [7], [17]. Reliably tracking features in the images is the main challenge. The control algorithm proposed can be easily adapted to a number of image features, both in sonar and video. The main focus of the paper is the joint control of a vehicle and arm systems with redundant degrees of freedom in a visual servoing framework. Visual servoing is a natural approach underwater as global positioning is extremely difficult whilst relative positioning with respect to a structure should be achievable. Indeed we have started to make progress on motion and pose estimation with respect to structures in sonar and video imagery [2], [7] as part of the trident project. While the first frameworks are based on the arm redundancy [13], more recent works use whole-body motion control with potential field approach [3]. Recently a general control framework based on the quadratic programming formalism has been proposed [6], [8], allowing inequality constraints and task hierarchy specification. The

O. Kermorgant was with Inria Rennes-Bretagne Atlantique, Rennes, France and is now with the ICube laboratory, Université de Strasbourg, Strasbourg, France. E-mail: kermorgant@unistra.fr

Y. Pétillot and M. Dunnigan are with the Ocean Systems Laboratory,
Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, U.K. E-mail:
y.r.petillot@hw.ac.uk, m.w.dunnigan@hw.ac.uk

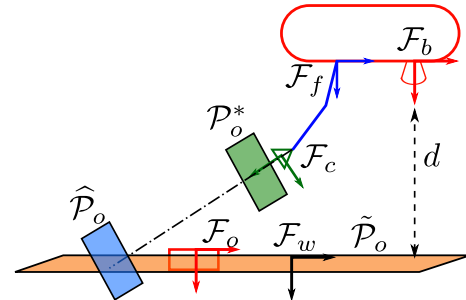


Fig. 1. Considered frames. The transformation between $\mathcal{F}_b$ and $\mathcal{F}_c$ is known through the robot model. $\mathcal{P}_o^*$ (green) corresponds to the desired object plane. The measurement of the altitude d makes it possible to estimate the position of the seabed plane $\mathcal{P}_o$ (orange). An robust plane estimation is $\hat{\mathcal{P}}_o$ (blue), that corresponds to the parallel to $\mathcal{P}_o^*$ passing through the intersection of $\hat{\mathcal{P}}_o$ and the optical axis (dotted line).

control schemes that have been proposed usually assume that the observed object is perfectly detected in the camera or sonar images. In practice, image moments [4] are well-suited (but not perfect) features for the underwater case, as they are robust to image noise and do not require any contour detection. They have been used in [9] for station keeping, and we propose to use these features to control the i-auv. We also consider the estimation of environment disturbances, leading to a feed-forward visual servoing scheme. In this paper we propose a whole-body control framework. The task is to position the camera with regards to an object, while ensuring classical constraints such as joint limits and singularity avoidance. Image moments are used for the target tracking and the disturbances estimation, and we propose a realistic implementation of these features. In Section II we expose the modeling and the proposed control law. We then detail the use of image moments in Section III and propose several ways to estimate their interaction matrix. The disturbance estimation filter is exposed in Section IV, allowing to have the vehicle align against the permanent current. We explain the choices that are made and expose possible alternatives, making it straightforward to use the proposed framework in other mobile manipulation cases, or with other sensors such as sonar or acoustic cameras. Finally the proposed scheme is validated in simulation.

II. MODELING AND CONTROL LAW

In this section we present the overall modeling of the vehicle-manipulator. The control scheme is then exposed.

A. Vehicle-manipulator modeling

We assume the vehicle-manipulator is observing an object with an eye-in-hand sensor and retrieves a vector of features $\mathbf{s}$. The considered sensor can be a camera or a sonar as

long as it is sufficiently accurate to localize the object. In this paper we propose using a camera, the extracted visual features being detailed in Section III. The control input is denoted $\mathbf{u} = (\dot{\mathbf{q}}, \mathbf{v})$ where $\dot{\mathbf{q}}$ is the manipulator joint velocities and $\mathbf{v} = (v, \omega)$ is the vehicle velocity expressed in its own frame denoted $\mathcal{F}_b$. The considered frames are represented in Fig. 1.

We denote $\mathbf{J}_q$ the Jacobian of the features with regards to the joints, and $\mathbf{J}_v$ the Jacobian with regards to the robot velocity. In the general case a vehicle is subject to environment disturbances that are denoted $\mathbf{w}$, expressed in the world frame $\mathcal{F}_w$. $\mathbf{w}$ is the uncontrolled velocity twist between the vehicle and the observed object. If the observed object is motionless then $\mathbf{w}$ simply reflects the environment velocity, such as current or waves. If the object has a proper motion then $\mathbf{w}$ reflects both the environment disturbances and the object own motion.

The time variation $\dot{\mathbf{s}}$ of the visual features yields:

$$\dot{\mathbf{s}} = \mathbf{J}_q \dot{\mathbf{q}} + \mathbf{J}_v \mathbf{v} + \mathbf{J}_w \mathbf{w} = \mathbf{J} \mathbf{u} + \mathbf{J}_w \mathbf{w} \quad (1)$$

where $\mathbf{J} = (\mathbf{J}_q, \mathbf{J}_v)$ is the global jacobian of the features with regards to the control input $\mathbf{u}$ and $\mathbf{J}_w$ is the Jacobian with regards to the external motion and can be expressed by:

$$\mathbf{J}_w = \mathbf{J}_v \begin{bmatrix} {}^b\mathbf{R}_w & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^b\mathbf{R}_w \end{bmatrix} = \mathbf{J}_v {}^b\tilde{\mathbf{R}}_w \quad (2)$$

where ${}^b\mathbf{R}_w$ is the rotation matrix between the world frame and the vehicle frame.

In practice, the joint Jacobian $\mathbf{J}_q$ can be expressed by:

$$\mathbf{J}_q = \mathbf{L}_s {}^e\mathbf{J}_q \quad (3)$$

where $\mathbf{L}_s$ is the interaction matrix of the features [5] and ${}^e\mathbf{J}_q$ is the classical robot arm Jacobian. Similarly, the velocity Jacobian $\mathbf{J}_v$ is expressed by:

$$\mathbf{J}_v = \mathbf{L}_s {}^e\mathbf{W}_e {}^f\mathbf{W}_f {}^b\mathbf{W}_b \quad (4)$$

where ${}^e\mathbf{W}_e$, ${}^f\mathbf{W}_f$ and ${}^b\mathbf{W}_b$ are the 6×6 velocity screw transformation matrices respectively coming from the sensor-to-end-effector transformation, from the arm kinematic model and from the arm-to-vehicle base transformation. All matrices of the proposed model can be assumed to be known with a good precision, except for $\mathbf{L}_s$ that depends on the features that are used. In the sequel any model inaccuracy is considered to be part of the world disturbances $\mathbf{w}$.

B. Classical control law

Assuming the task is to have the features reach a desired value $\mathbf{s}^*$, we define the task error $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$. A classical controller can then be obtained by a feed-forward scheme [1] using an estimation of the pseudo-inverse of $\mathbf{J}$:

$$\begin{aligned} \mathbf{u} &= \underset{\mathbf{u}}{\operatorname{argmin}} \left\| \hat{\mathbf{J}} \mathbf{u} + \hat{\mathbf{J}}_w \hat{\mathbf{w}} + \lambda \mathbf{e} \right\|^2 \\ &= -\lambda \hat{\mathbf{J}}^+ \mathbf{e} - \hat{\mathbf{J}}^+ \hat{\mathbf{J}}_w \hat{\mathbf{w}} \end{aligned} \quad (5)$$

where $\hat{\mathbf{w}}$ is an estimation of the velocity disturbances.

Yet, such a control law would distribute the error and disturbance terms over all the control inputs to obtain the minimal norm for $\mathbf{u}$. This is usually a nice property, but in the case of a vehicle-manipulator system it leads to two undesired behaviors:

- Depending on the distance to the object, we may prefer to use the vehicle body motion or the arm motion. Typically it is useless to extend the arm towards the object if it is too far.
- Only the vehicle body can take into account constant disturbances (eg. current). Indeed, in this case the arm would only extend until it reaches its joint limits. Besides, varying disturbances may be rejected better by the arm since it usually has better dynamic properties.

We now propose a general control scheme that takes into account these criteria.

C. Vehicle-manipulator control scheme

The framework of hierarchical quadratic programming (HQP) [8] is used to design a control law ensuring the task hierarchy that suits with the particular properties of a vehicle-manipulator system. The control scheme is divided into three priority levels :

- The higher level ensures that the arm joint limits and singularity avoidance are taken into account.
- The second level minimizes at best the visual error, without violating the constraint.
- The third level focuses on the control input distribution, and allows balancing the use of the vehicle and the arm depending on the configuration.

Two configurations are considered:

- The first one corresponds to the approach phase. In this case we want to minimize the use of the arm.
- The second one corresponds to the manipulation or intervention phase. Here the goal is to use the vehicle velocity only to cope with the constant disturbances, while the arm performs the final positioning.

We now detail the formulation of the three hierarchy levels.

1) *Joint limits and singularity*: On the highest level, the arm joint limits and singularity avoidance have to be taken into account. The joint limits $(\mathbf{q}^-, \mathbf{q}^+)$ are classically ensured with two inequality constraints:

$$\begin{cases} \dot{\mathbf{q}} > \alpha(\mathbf{q} - \mathbf{q}^-) \\ \dot{\mathbf{q}} < -\alpha(\mathbf{q} - \mathbf{q}^+) \end{cases} \quad (6)$$

where $\alpha > 0$ tunes the constraint. It is easily shown that if (6) is ensured then $\mathbf{q}^- \leq \mathbf{q} \leq \mathbf{q}^+$. Similarly, a classical way to avoid singularity is to impose a lower bound $d_{\min}$ for $\det(\mathbf{J}_q^T \mathbf{J}_q)$, which again can be expressed with regards to $\dot{\mathbf{q}}$ by:

$$\mathbf{J}_d \dot{\mathbf{q}} > \beta (\det(\mathbf{J}_q^T \mathbf{J}_q) - d_{\min}) \quad (7)$$

where $\beta > 0$ tunes the constraint and where $\mathbf{J}_d$ is the Jacobian expressing the variation of $\det(\mathbf{J}_q^T \mathbf{J}_q)$ with regards to $\mathbf{q}$ and can be computed numerically from the kinematic model. We denote $\mathcal{U}_q$ the control input domain that satisfies (6) and (7).

2) *Task completion*: The second priority level is the one corresponding to the task itself. The corresponding optimization problem is thus:

$$\mathcal{U} = \underset{\mathbf{u} \in \mathcal{U}_q}{\operatorname{argmin}} \left\| \hat{\mathbf{J}} \mathbf{u} + \hat{\mathbf{J}}_w \hat{\mathbf{w}} + \lambda \mathbf{e} \right\|^2 \quad (8)$$

TABLE I
HIERARCHY LEVELS DEFINING THE CONTROL LAWS IN THE APPROACH AND IN THE INTERVENTION PHASES.

	Approach: $\mathbf{u}_a \in \mathcal{V} \subset \mathcal{U}_q$	Intervention: $\mathbf{u}_i \in \mathcal{U} \subset \mathcal{U}_q$
Level 1	$\begin{cases} \mathbf{q}^- < \mathbf{q} < \mathbf{q}^+ \\ \det(\mathbf{J}_q^T \mathbf{J}_q) > d_{\min} \end{cases}$	$\begin{cases} \mathbf{q}^- < \mathbf{q} < \mathbf{q}^+ \\ \det(\mathbf{J}_q^T \mathbf{J}_q) > d_{\min} \end{cases}$
Level 2	$\min \left\ \hat{\mathbf{J}}\mathbf{u} + \hat{\mathbf{J}}_w \hat{\mathbf{w}}^c + \lambda \mathbf{e} \right\ ^2$	$\min \left\ \hat{\mathbf{J}}\mathbf{u} + \hat{\mathbf{J}}_w \hat{\mathbf{w}} + \lambda \mathbf{e} \right\ ^2$
Level 3	$\min \left\ \dot{\mathbf{q}} \right\ ^2 + \left\ \boldsymbol{\omega}_z + \lambda_\omega \arctan \frac{\hat{\mathbf{v}}_y^c}{\hat{\mathbf{v}}_x^c} \right\ ^2$	$\min \left\ \mathbf{v} + {}^b\tilde{\mathbf{R}}_w \hat{\mathbf{w}}^c \right\ ^2 + \left\ \boldsymbol{\omega}_z + \lambda_\omega \arctan \frac{\hat{\mathbf{v}}_y^c}{\hat{\mathbf{v}}_x^c} \right\ ^2$

Alternatively, during the approach we may want to compensate only the constant disturbances in order to avoid oscillations in the vehicle velocity control. Denoting $\hat{\mathbf{w}}^c$ the estimation of the constant disturbances, the control input domain $\mathcal{V}$ corresponding to the approach phase yields:

$$\mathcal{V} = \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}_q} \left\| \hat{\mathbf{J}}\mathbf{u} + \hat{\mathbf{J}}_w \hat{\mathbf{w}}^c + \lambda \mathbf{e} \right\|^2 \quad (9)$$

3) *Approach vs Intervention*: The last priority level balances the control input in the approach and the intervention phases. During the approach the goal is to minimize the use of the arm. A classical desired behavior is also to have the vehicle face the current, as the forward motion is usually more efficient. In the case of a moving object (drifting buoy for example) this would lead to the vehicle aligning with the object motion, which is a nice property. If the forward motion is in the x -axis, the approach control input is:

$$\mathbf{u}_a = \operatorname{argmin}_{\mathbf{u} \in \mathcal{V}} \left\| \dot{\mathbf{q}} \right\|^2 + \left\| \boldsymbol{\omega}_z + \lambda_\omega \arctan \frac{\hat{\mathbf{v}}_y^c}{\hat{\mathbf{v}}_x^c} \right\|^2 \quad (10)$$

where $\boldsymbol{\omega}_z$ is the yaw control, λ_ω is the yaw gain and $(\hat{\mathbf{v}}_x^c, \hat{\mathbf{v}}_y^c)$ is the estimation of the horizontal permanent current, expressed in the vehicle frame. During the intervention the goal is to have the vehicle velocity cope with the constant disturbance. The intervention control input is thus:

$$\mathbf{u}_i = \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} \left\| \mathbf{v} + {}^b\tilde{\mathbf{R}}_w \hat{\mathbf{w}}^c \right\|^2 + \left\| \boldsymbol{\omega}_z + \lambda_\omega \arctan \frac{\hat{\mathbf{v}}_y^c}{\hat{\mathbf{v}}_x^c} \right\|^2 \quad (11)$$

The overall hierarchy levels defining the two control inputs $\mathbf{u}_a$ and $\mathbf{u}_i$ are summed up in Table I. We now expose the transition between $\mathbf{u}_a$ and $\mathbf{u}_i$.

4) *Control input distribution*: Whichever the choice of the features, a scalar function $\xi(\mathbf{s})$ can be defined such that ξ decreases while the task is being performed. ξ can be the inverse of the area of the object in the image, or the distance between the vehicle and the object if it can be estimated. In the general case one can simply set $\xi(\mathbf{s}) = \|\mathbf{s} - \mathbf{s}^*\|^2$. Two values ξ_a and ξ_i can then be defined such that:

- if $\xi(\mathbf{s}) > \xi_a$ the vehicle is fully in the approach phase
- if $\xi(\mathbf{s}) < \xi_i$ the vehicle is fully in the intervention phase

The actual control law is defined as a linear combination of $\mathbf{u}_a$ and $\mathbf{u}_i$:

$$\mathbf{u} = \begin{cases} \mathbf{u}_a & \text{if } \xi(\mathbf{s}) > \xi_a \\ \mathbf{u}_i & \text{if } \xi(\mathbf{s}) < \xi_i \\ \frac{(\xi - \xi_i)\mathbf{u}_a + (\xi_a - \xi)\mathbf{u}_i}{\xi_a - \xi_i}, & \text{else} \end{cases} \quad (12)$$

$\mathbf{u}$ is thus continuous during the intervention and changes from $\mathbf{u}_a$ to $\mathbf{u}_i$ as the vehicle approaches its target.

The global control scheme that has been presented in this section keeps the same local stability properties as the classical control law (5), but ensures the joint limits avoidance and the desired control input distribution between the approach and the intervention phases. The performances are thus depending on the estimation of the model matrices and of the environment disturbances.

III. VISUAL FEATURES

The proposed framework is independent on the visual features that are used. However the nature of underwater images makes it difficult to extract precise features. In this section we present and detail the choice of image moments and propose the estimations that can be used for the underwater case. In this paper we are expecting to perform autonomous manipulation based on vision only when visibility is good using video. We would move to sonar when conditions are poor, though theoretical work is still to be done on sonar.

A. Moments from image

A basic image processing that can be performed underwater is image segmentation from color. In the sequel we assume the observed object has a distinctive color that can be segmented in the image. A binary image is thus obtained.

The moment m_{ij} of an object in the image yields:

$$m_{ij} = \int \int_{\mathcal{O}} x^i y^j dx dy \quad (13)$$

where $\mathcal{O}$ is the projection of the object in the image and (x, y) are metric coordinates. Denoting $I(u, v)$ the intensity of the pixel of coordinates (u, v) , we have $I(u, v) = 1$ if the pixel belongs to $\mathcal{O}$ and $I(u, v) = 0$ otherwise. The actual way to compute m_{ij} is thus:

$$m_{ij} = \frac{1}{p_x p_y} \sum_u \sum_v I(u, v) x^i y^j \quad (14)$$

where (p_x, p_y) is the ratio between the focal length and the pixel size. Once the object is segmented, (14) allows retrieving any image moment without any image processing.

In a classical visual servoing scheme, the visual features are associated with desired values. The desired moments can simply be computed with (14) from the desired image. If a model of the observed object is known, the desired features may also be generated from the desired position using

moments from polygon [18]. Whichever way the desired values are defined, we need to choose a set of visual features and to compute the corresponding interaction matrix. In the next section we recall the classical features that are derived from moments, before estimating their interaction matrix.

B. Visual servoing features

In order to control the full position of the end-effector, 6 different features have to be used in the visual servoing. The first four ones are classically the area of the object a , the position (x_g, y_g) of the centroid and the object orientation α . If the full orientation of the end-effector has to be controlled, two additional features must be used. In the case of intervention-auv, the desired position often corresponds to having the camera in a known orientation with regards to the seabed (typically facing the seabed). Assuming the vehicle body stays approximately horizontal (small pitch and roll), or assuming the pitch and roll are estimated, the two last features are denoted $(\theta_{\mathbf{u}_x}, \theta_{\mathbf{u}_y})$ and simply correspond to the first two terms of the orientation $\mathbf{R}_w$ in the $\theta\mathbf{u}$ representation. These features are not computed from the image, yet they can be associated to image-based features to form a hybrid set of features:

$$\mathbf{s}_{\text{hyb}} = (a, x_g, y_g, \alpha, \theta_{\mathbf{u}_x}, \theta_{\mathbf{u}_y}) \quad (15)$$

The advantage of $\mathbf{s}_{\text{hyb}}$ over a pure set of 6 image features [4] is that the higher-order moments may be subject to image noise. On the opposite, if the vehicle is stable in roll and pitch, and if the desired orientation is known, then $\theta_{\mathbf{u}_x}$ $\theta_{\mathbf{u}_y}$ are less likely to be perturbed.

C. Estimating the interaction matrix

In [4] the interaction matrix of any moment m_{ij} is derived. A common approximation is that the observed object is planar, which is realistic with most underwater cases. In this case the interaction matrix of any moment depends on the parameters of the plane in the camera frame, that correspond to the plane equation:

$$1/Z = Ax + By + C \quad (16)$$

where Z is the positive depth (i.e. the plane is in front of the camera), (x, y) are the image coordinates and A, B and C are the plane parameters. For example the interaction matrix related to the object area a yields:

$$\mathbf{L}_a = [-aA \quad -aB \quad a(3/Z_g - C) \quad 3ay_g \quad -3ax_g \quad 0] \quad (17)$$

where $Z_g = Ax_g + By_g + C$. The analytical form of the interaction matrices of the other visual features (x_g, y_g, α) can be found in [4]. The estimation of the interaction matrix thus depends on the estimation of A, B and C . That is why we now compare the use of three different planes, that are represented in Fig. 1.

1) *Desired object plane*: The first one is the classical estimation, that assumes the object already lies in the desired plane $\mathcal{P}_o^*$ (green on Fig. 1). The advantage is that the parameters of this plane are known in the camera frame, but this leads to a coarse estimation of the interaction matrix.

2) *Seabed plane*: If the altitude d can be measured, and assuming the observed object lies approximately on the

seabed, then the seabed plane $\tilde{\mathcal{P}}_o$ (orange on Fig. 1) is likely to be the best estimation of the object plane. Indeed in this case the seabed plane parameters are easily written in the vehicle frame $\mathcal{F}_v$. As the transformation matrix $\mathbf{M}_b$ is known through the robot model, the plane parameters are easily transposed to the camera frame. Yet this estimation may be subject to noise error, leading to an unreliable orientation (coarse estimation of A and B).

3) *Intermediary plane estimation*: An intermediary plane estimation is represented as $\hat{\mathcal{P}}_o$ (blue on Fig. 1). This plane is defined as the parallel to the image plane that passes through the intersection of the seabed plane and the camera optical axis. The parameters of this plane can be easily retrieved by measuring the altitude d , as we have $A = B = 0$, and C has the same value as the seabed plane $\tilde{\mathcal{P}}_o$. Although this estimation is less precise than using the seabed plane, it is also less subject to measurement noise.

As for the interaction matrix of the 3D features $(\theta_{\mathbf{u}_x}, \theta_{\mathbf{u}_y})$, the analytical form is given in [5] and only depends on the current value of $\theta\mathbf{u}$. The corresponding interaction matrix can thus be estimated. In the next section we present the disturbance estimation scheme.

IV. DISTURBANCE ESTIMATION

The estimation of the velocity disturbance $\mathbf{w}$ relies on the differential kinematic model (1). The time variation of the features $\dot{\mathbf{s}}$ can be measured between two images and compared to the command $\mathbf{u} = (\dot{\mathbf{q}}, \mathbf{v})$ that has been applied. The velocity part $\mathbf{v}$ can also be estimated through a Kalman filter if additional sensors (i.e. IMU) are present. The first estimation of $\mathbf{w}$ thus yields:

$$\hat{\mathbf{w}} = \mathbf{J}_w^+(\dot{\mathbf{s}} - \hat{\mathbf{J}}\mathbf{u}) \quad (18)$$

This estimation usually suffers from noise error, hence it is classically associated with a Kalman filter [1].

The constant part of the disturbance, that is used to optimize the vehicle velocity control in Section II-C.4, can then be extracted with a low-pass filter:

$$\begin{cases} \hat{\mathbf{w}}^c(0) &= 0 \\ \hat{\mathbf{w}}^c(k+1) &= (1 - \eta)\hat{\mathbf{w}}^c(k) + \eta\hat{\mathbf{w}}(k+1) \end{cases} \quad (19)$$

where $\eta \in [0, 1]$ allows tuning the filter sensitivity. Such an estimation of the constant disturbances is meaningful because the disturbances are represented in the world frame. This is not the case when the disturbances are only represented in the sensor space, as the relationship between the vehicle velocity and the time variation of the features is non linear. As shown in Section II-C, it allows the vehicle body to cancel the constant disturbance in the intervention phase.

V. EXPERIMENTS

The experiments are carried in simulation with the simulator UWSim [14]. For these experiments we simulate the Girona 500 [16] and a 7-DOFs arm from Graaltech. The image processing and control scheme are designed using ViSP software [12].

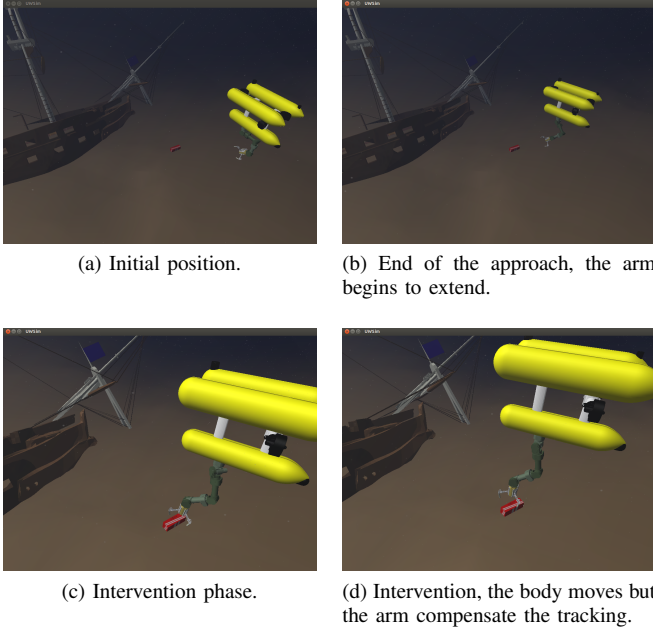


Fig. 2. Experimental setup. The task is to position the end-effector of the intervention-robot with regards to the black box lying on the seabed.

A. Experimental setup

The task is to position the end-effector with regards to a black box that is lying on the seabed, as shown on Fig. 2 (see also the video accompanying this paper). The vehicle body is controlled in the 3 directions and in yaw: $\mathbf{v} = (v_x, v_y, v_z, \omega_z)$. Disturbances, however, take place in all translational and rotational directions. The disturbances are simulated with a combination of 3 sine functions of different amplitudes and pulses. We also consider a permanent, horizontal current that is not aligned with the vehicle at the beginning of the experiment: at this time the current yields $(-0.2, 0.2, 0)$ m/s in the vehicle frame. The global current is thus 0.28 m/s. The hybrid set of features (15) is used to control the vehicle, with intermediary plane estimation $\hat{\mathcal{P}}_o$. Red color extraction is performed in order to segment the black box. The desired features are computed from the desired pose and the moments from polygon [18]. The desired black box position is represented with the green rectangle in Fig. 3. We do not detail the tuning parameters, as they are very dependent on the considered robot.

B. Analysis

Fig. 4 represents the different aspects of the behavior of the vehicle. The evolution of the feature error is shown in Fig. 4a. As expected the error converges to 0. Because of the various constraints (joint limits and arm singularity) and of the last optimization step (control input distribution as described in Section II-C.4) the decrease of the error is not decoupled. Indeed only control law (5) would lead to an exponential, decoupled decrease of all the features. The pose error is represented in Fig. 4b. t_y (green) represents the translational direction corresponding to the forward motion, and shows that the vehicle initial position is about 2.5 m from the

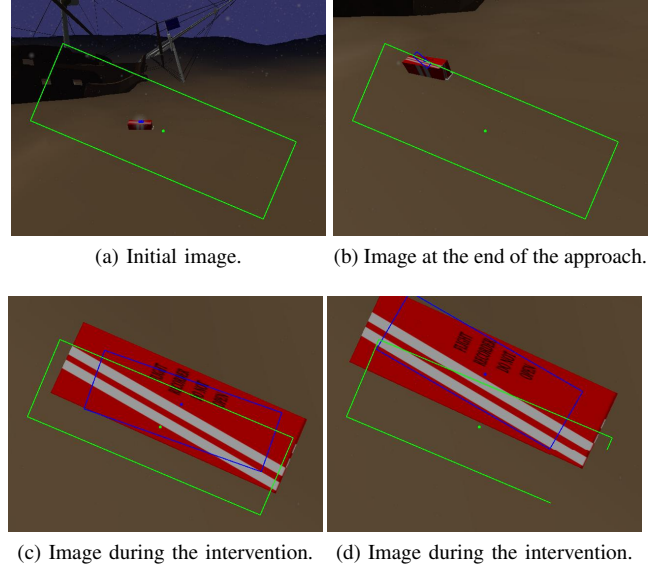


Fig. 3. Images corresponding to the positions in Fig. 2. The green rectangle is the desired position of the black box.

target. The pose error stabilizes around 0 with a very good accuracy (less than 0.1 m). The vehicle velocity is shown in Fig. 4c. As ensured by the control input distribution (11), the vehicle aligns against the constant current that is estimated. The forward velocity v_x (blue) thus stabilizes around the current norm, that is 0.28 m/s. The side velocity v_y (green) first copes with the side current before stabilizing around 0 as the vehicle ends up facing the current. The estimation of the horizontal constant current is represented in Fig. 4d. The dotted lines show the actual current, and the solid lines represent the constant estimation that is computed from the low-pass filter (19). Here we set $\eta = 0.02$: this makes the estimation sensitive to some variations, but it is more reactive during the initial guess. Finally, the global 3D trajectories of the vehicle and the camera are represented in Fig. 5. The facing current first makes the vehicle move backwards. It is then compensated by the forward motion. The yaw between the initial (blue) and final (red) positions is clearly visible, showing that the vehicle has aligned against the current. As they are considered as hard constraints, the joint limits and singularity are of course avoided during the task.

VI. CONCLUSIONS

In this paper we have proposed a global control scheme for free-floating underwater manipulators. Several constraints and realistic criteria are considered in a whole-body optimization scheme, allowing to perform the sensor-based task while ensuring the joint limits and singularity avoidance together with a smart control input distribution between the arm and the vehicle body. The proposed framework has been validated in a dynamic simulation, even if the control law is only at the kinematic level. The considered sensor is a camera with image moments, which can also be used with sonar images. A direct improvement would be to consider the dynamic model in the control law. Also, this work could

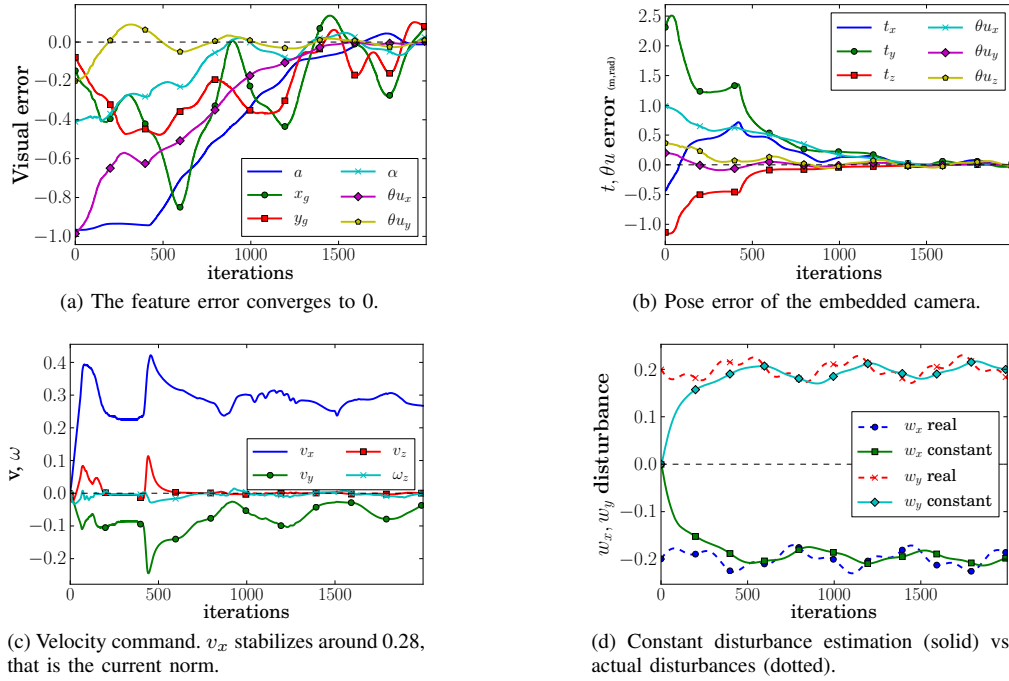


Fig. 4. Behavior of the control law. Visual error (a), pose error (b), vehicle velocity command (c) and disturbance estimation (d). The vehicle aligns itself against the current. The estimation of the constant disturbances quickly converges to the actual value, then varies slightly.

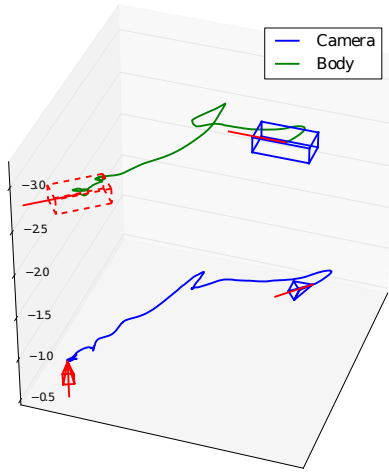


Fig. 5. Trajectories of the camera (blue) and of the vehicle (green). The initial positions are in blue, and final positions in red.

benefit from a better disturbance estimation filter. Future works will consider auto-regressive filtering to improve the disturbance estimation. In the near future, our algorithm will be tested on the Girona 500 platform.

REFERENCES

- [1] F. Bensalah and F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Pittsburgh, USA, August 1995.
- [2] F. Bonin, A. Burguera, and G. Olivier, "Imaging systems for advanced underwater vehicles," *J. Mar. Res.*, 2011.
- [3] G. Casalino, E. Zereik, E. Simetti, S. Torelli, A. Sperinde, and A. Turetta, "Agility for underwater floating manipulation: Task & subsystem priority based control strategy," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, September 2012.
- [4] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. on Robotics*, Aug. 2004.
- [5] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, 2006.
- [6] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming," Report, Oct. 2012.
- [7] N. Hurtos, X. Cuf, Y. Petillot, and J. Salvi, "Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.
- [8] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task," *IEEE Trans. on Robotics*, 2011.
- [9] J. Lots, D. Lane, E. Trucco, and F. Chaumette, "A 2D visual servoing for underwater vehicle station keeping," in *IEEE Int. Conf. on Robotics and Automation, ICRA'01*. IEEE, 2001.
- [10] G. Marani and S. Choi, "Underwater target localization," *IEEE Robot. Autom. Mag.*, 2010.
- [11] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions auvs," *Ocean Engineering*, 2009.
- [12] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, December 2005.
- [13] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, 1987.
- [14] M. Prats, J. Pérez, J. J. Fernández, and P. J. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, October 2012.
- [15] M. Prats, D. Ribas, N. Palomeras, J. C. García, V. Nannen, S. Wirth, J. J. Fernández, J. P. Beltrán, R. Campos, P. Ridao *et al.*, "Reconfigurable auv for intervention missions: a case study on underwater object recovery," *Intelligent Service Robotics*, 2012.
- [16] D. Ribas, P. Ridao, L. Magí, N. Palomeras, and M. Carreras, "The girona 500, a multipurpose autonomous underwater vehicle," in *IEEE OCEANS*, Santander, Spain, June 2011.
- [17] H. Sekkati and S. Negahdaripour, "3-d motion estimation for positioning from 2-d acoustic video imagery," in *IbPRIA (2)*, 2007.
- [18] C. Steger, "On the calculation of arbitrary moments of polygons," *Munich Univ., Munich, Germany, Tech. Rep. FGBV-96-05*, 1996.