Optimal thrusters steering for dynamically reconfigurable underwater vehicles

Maxence Blond, Daniel Simon, Vincent Creuze, Olivier Tempier

▶ **To cite this version:**

Maxence Blond, Daniel Simon, Vincent Creuze, Olivier Tempier. Optimal thrusters steering for dynamically reconfigurable underwater vehicles. International Journal of Systems Science, 2019, pp.2348-2361. 10.1080/00207721.2019.1655603 . lirmm-02268706

HAL Id: lirmm-02268706

https://hal-lirmm.ccsd.cnrs.fr/lirmm-02268706

Submitted on 21 Aug 2019

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.

DRAFT

Optimal thrusters steering for dynamically reconfigurable underwater vehicles

Maxence Blond[a,b], Daniel Simon[c,b], Vincent Creuze[b] and Olivier Tempier[b]

[a]Subsea Tech, Marseille, France; [b] LIRMM - CNRS - University of Montpellier, France; [c] CAMIN team, INRIA Sophia Antipolis-Mediterranee, France

ARTICLE HISTORY
Compiled August 21, 2019

The Version of Record of this manuscript has been published and is available in International Journal of Systems Sciences, http://www.tandfonline.com/, August 20 2019, DOI: 10.1080/00207721.2019.1655603

Abstract
When using a Remotely Operated Vehicle (ROV), on-the-fly reconfiguration of the thrusters orientations allows to adjust its propulsion and maneuverability capabilities according to the mission progress. To optimize the actuation, the interaction between thrusters due to cross flows is modeled and included in the thrust related objective function to be maximized. Run-time effective solutions use a sparse look-up table to initialize a fast direct-search local optimization algorithm. The found thrusters steering configurations show higher thrust gains compared with the traditional fixed 'vectored' configuration of currently available ROVs.

KEYWORDS
Marine Robotics, Reconfigurable Underwater Vehicle, Propulsion Optimization, On-line optimization

1. Introduction

1.1. Context and State of the Art

Remotely Operated Vehicles (ROVs) are extensively used in various fields, e.g., for offshore monitoring and maintenance, or for environment assessment.

ROVs are underwater devices attached to a support vessel by a tether which carries both data (video, sensors...) and electrical power. As their mobility capabilities are limited by the tether and its induced disturbances, they are often used for small or medium range motions, quasi-static observation and manipulation tasks. However, station-keeping while being subject to, e.g., underwater currents, needs high maneuverability and controllability thanks to a well chosen set of thrusters.

Vertical thrusters allow to control the depth, and in some cases to ensure roll and pitch stabilization. The set of horizontal thrusters allows to control motions in the horizontal plane, i.e. translations in all directions of the plane and yaw rotations around the vertical axis.

Traditionally the thrusters have a fixed position and orientation on the ROV, and their number and location is chosen according to the mission assigned to the vehicle. For work-class ROVs, often designed for station keeping or quasi-static tasks, a very common thrusters disposition is the so-called 'vectored' configuration where four –thus redundant– actuators are set-up in a diamond layout. This configuration roughly allows homogeneous thrust and motion for all directions of the horizontal plane.

However, as there is not privileged thrust direction, such ROVs are not optimized for medium range motions, e.g., as needed for pipe-line or wall inspection. In such tasks, the ROV and tether can be subject to strong forces due to sea current, needing to maximize the push a the desired direction which is not necessarily aligned with the vehicle's body. Nevertheless the underwater vehicles which are specifically shaped for speed poorly behave in station-keeping tasks.

Hence the idea of re-configurable ROVs using variable thrusters azimuth angles arose to design multi-purpose

Contact author : daniel.simon@inria.fr

vehicles. Some underwater vehicles, such as Girona 500 [Ribas, Palomeras, Ridao, and Mallios (2012)], SeaDrone [Moreno and Chung (2014)] or e-URoPe [Odetti et al. (2017)] among others, have been designed with off-line variable actuators configuration in mind. The number, location and orientation of the thrusters can be chosen according to a specific mission. This choice cannot be dynamically modified during the mission, and the mission dependent configurations are set according to 'rules of thumb' considerations.

Conversely with terrestrial robots and aerial drones, few underwater robots have been designed for dynamic reconfigurability. However adaptive steering of propellers is already used in some surface ships for Dynamic Positioning (DP) tasks. The approach aims at optimizing the controllability of very big vessels for station keeping in hard environment conditions due to combined variable wind, current and swell. Among various algorithms and heuristics, several optimization methods have been used such as bio-inspired genetic algorithms (GA) (e.g. [Ding, Yang, and Huang (2016)] and [Xu, Wang, Song, Zheng, and Chen (2011)]) and Particle Swarm Optimization (e.g. [Wang, Gu, and Zou (2013)]), but also the 'brute force' approach as in [Patel, Frank, and Crane (2014)], and a Damped Least Squares control approach is recorded in [Benetazzo, Ippoliti, Longhi, Raspa, and Sørensen (2012)]. Even if big vessels may allow for long computation times due to their large time constants, the run-time efficient Sequential Quadratic Programming (SQP) approach has been also successfully applied for DP tasks. The work in [Johansen, Fuglseth, Tondel, and Fossen (2008)] deals with the optimal control allocation for DP using rudders, and in [Johansen, Fossen, and Berge (2004)] singularities of the actuation configuration matrix must be avoided. In the latter example a fast convergence to the optimal solution is recorded in the range of 2 ms using a standard 1 GHz single core CPU. Even closer to the case study described in this paper –and still using the SQP optimization method– [Arditti, Souza, Martins, and Tannur (2015)] considers the interaction between propellers where "efficiency functions" are used to model the efficiency reduction of thrusters due to cross-flows.

Few small underwater robots have been designed using dynamically configurable sets of thrusters. Moreover, the reported works (such as [Pugi, Allotta, and Pagliai (2018)]) focus on control design while thrust optimization is studied through heuristics. A noticeable exception is the work reported in [Vega (2017)] considering an Autonomous Underwater Vehicle (AUV) fitted with on-line steerable propellers. A preliminary optimization process jointly considers both the thrust, position and steering angle of the thrusters and uses a GA to solve the large non-linear problem. As the first algorithm is found to be very slow, it is used only for off-line pre-configuration. A more run-time effective solution –still using a GA– is then proposed to solve on-line thrusters steering considering a fixed set of locations together with a 'computed torque' control algorithm and feedback linearization.

Finally, a new ROV named Tortuga 500 is under design by the Subsea Tech company[1]. It is a small weight/small sized robot intended to inspect underwater structures through manholes. It is intended to be operated in currents up to four knots down to 500 meters from a support boat. It is expected that a dynamic steering of its over-sized propellers can extend its capabilities over a large range of underwater missions.

1.2. Goal and contributions

In this paper, a method aiming at maximizing the global thrust acting on an underwater vehicle in a desired direction is investigated, considering the dynamic steering of four horizontal thrusters actuated in rotation along their vertical axis. A simplified interaction model to take into account the effects of cross flows between thrusters is introduced, together with the interaction of the water flows with the hull of the vehicle. Then two optimization methods are proposed, compared and associated, namely the brute-force computation which is further used to provide a suitable initial solution for a fast local optimization algorithm. These methods provide the best steering angles for the set of thrusters to maximize the achievable speed in a preferred direction, while satisfying several additional constraints. The first constraint considers the maneuverability ratio (i.e. the ratio of achievable speed respectively along the preferred direction and transverse axis of motion). A second constraint aims at enforcing symmetrical solutions to minimize the undesired torque induced in yaw –i.e. rotations around the vertical axis of the vehicle–. Finally, the integration of the configuration method in the mission and motion control schemes of the vehicle is sketched to allow on-the-fly optimization during missions on the field.

Modeling assumptions, including models of thruster interactions due to cross flows, are given in the next section. Several optimization approaches, considering possible objective functions and optimization algorithms, are studied in

[1] https://www.subsea-tech.com/

section 3. The integration of the optimization process with the low-level control loops is sketched in section 4. Finally the approach is discussed and perspectives are given in the conclusion.

2. Modeling and assumptions

2.1. Vehicle and frames

The robot considered in the following is Tortuga 500, which is a Remotely Operated Vehicle (ROV) actuated by six thrusters. Two vertical thrusters allow to control the depth (vertical translation) and roll motions. Pitch is passively stabilized and damped through the hydro-static stability. Four horizontal thrusters exert forces in the horizontal plane allowing horizontal translations and rotations around the yaw axis. The originality of the considered ROV with respect to other commercially available ROVs is the fact that the azimuth angle of each horizontal thruster is actuated by a servomotor. Thus, each horizontal thruster's orientation with respect to the vehicle can be independently controlled in real time during the mission. Figure 1 gives an overview of this setup.

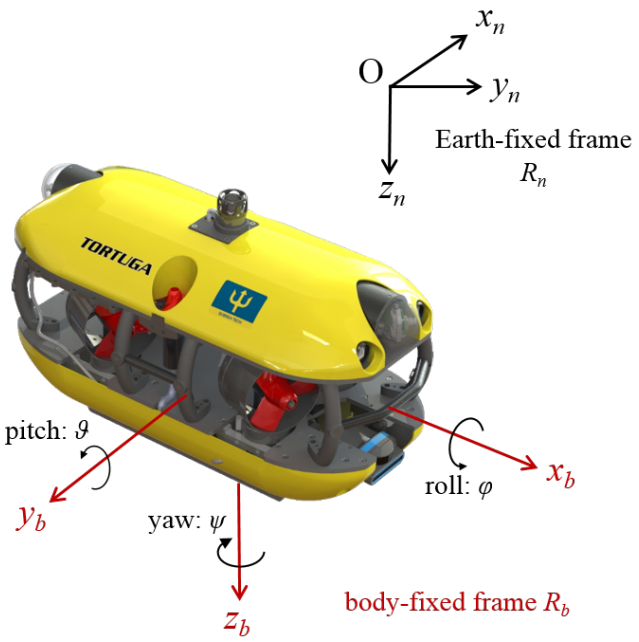

Figure 1. The Tortuga 500 ROV with the R_n Earth-fixed frame and the R_b body-fixed frame. Copyright: Subsea Tech.

In the following, frames are described using the SNAME notation, as defined for instance in [Fossen (2002)]. The state vectors are expressed in two reference frames: the body-fixed frame R_b and the inertial earth-fixed frame R_n, also denoted NED (North-East-Down). The origin of the body-fixed frame is located at the center of buoyancy of the ROV. The state of the robot is described by vectors $\eta = [x, y, z, \varphi, \vartheta, \psi]^T$, giving the position and attitude of the robot in the earth frame, and $\nu = [u, v, w, p, q, r]^T$ expressing the velocities in the body-fixed frame. (x, y, z) are the coordinates of the center of R_b expressed in R_n, while (u, v, w) are the translational speeds expressed in R_b, respectively along Ox_b, Oy_b and Oz_b axes.The sequence of Euler angle rotations $(\varphi, \vartheta, \psi)$ (roll, pitch and yaw) express the orientation of R_b with respect to R_n and thus indicates the attitude of the robot. Finally, $(p, q$ and $r)$ denote the rotational speeds with respect to the Ox_b, Oy_b and Oz_b axes of the body-fixed frame R_b.

According to these notations, the dynamics of the vehicle can be expressed by the following equation:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau + w_d \tag{1}$$

where matrices M, C, and D respectively denote the inertia (including the added masses), the Coriolis-centripetal

forces (also including the added masses) and the damping effects. g is the vector of gravitational and buoyancy forces and torques, τ is the vector of control inputs and w_d denotes the external disturbances.

2.2. Modeling assumptions

In the following, the roll angle is supposed to be regulated to 0 by a separate controller, and the pitch angle is supposed to be naturally stable around 0°so that the ROV remains horizontal when it moves. The depth is also assumed to be controlled by a dedicated regulator. This paper only focus on motions of the robot in the horizontal plane, i.e. translations along the Ox axis (surge) and Oy axis (sway), and rotation ψ around the Oz axis (yaw). The method and results presented here are applied to an ROV, but could be of interest also for any re-configurable underwater vehicle.

In this study, the speeds are low enough to consider that the Coriolis terms as negligible ($C(\nu)\nu \simeq 0$). Due to the symmetry of the hull and due to the fact that the center of the body-fixed frame coincides with the center of symmetry of the hull, it is assumed that the damping matrix D is diagonal. The inertial effects of the rotational accelerations of the thrusters (motor + shaft + propeller + fluid) are also neglected, as well as the gyroscopic effects induced by the combination of the rotation of the thrusters and the vehicle itself. The interested reader can refer to [Maalouf, Creuze, Chemori, and Tempier (2015)] about the quantification of these disturbances and the possible active compensation of their undesired effects.

Local low-level speed controllers are supposed to allow for a direct control of the rotational speed of each propeller –this is a common feature with brush-less thrusters. Therefore it is also assumed that any hysteresis or dead-zone in the thruster's characteristics is compensated at low-level.

Within the optimization process described later on, the steering dynamics of the thrusters is not considered. Indeed the time constants of the thrusters steering process are small compared with the underwater mission sequence of actions, such that under-optimal transients would only have a limited impact on the global efficiency. The actuators induced torque disturbances when steering are assumed to be damped or canceled in real-time at control level.

It is also assumed that the vertical thrusters do not interact with the horizontal ones. This assumption is satisfied for Tortuga 500 as by many other similar underwater vehicles. Firstly, the hydrodynamic interaction of perpendicular water flows in open environment (neither nozzle nor shared tunnel) is negligible. Secondly, in all the possible orientations of the horizontal thrusters, their water flows do not cross the water flows of the vertical ones.

The reaction torque generated by the propellers rotation is not considered. As the vertical thrusters are counter-rotating, the related reaction torques are almost cancelled during symmetric operations –i.e., for vertical motions–, and remaining disturbances around the yaw axis are cancelled by the yaw servo-loop. The disturbances induced by the horizontal propellers are damped and cancelled both by the roll servo-loop and by the natural restoring hydrostatic torque.

2.3. Thrusters

2.3.1. Location and orientation

As previously said, the vertical thrusters of the robot are not considered, and the only thrusters actuating the degrees of freedom in the horizontal plane (x, y and ψ) are the horizontal ones.

These thrusters are labeled T_i with $i = \{1, .., 4\}$. They are located as depicted by Figure 2. Each thruster's coordinates are denoted (x_i, y_i). The vertical axis of each thruster is actuated by a servomotor, which allows to modify the steering angle α_i of thruster T_i during the mission. In what follows, for clarity purposes, the thrusters are assumed to have symmetrical capabilities in forward and reverse directions.

2.3.2. Thrusters Configuration Matrix

The forces and torques $\tau = [f_x, f_y, \gamma_z]^T$ applied to the robot in its horizontal plane are computed from the thrust forces $f = [\tau_1, \tau_2, \tau_3, \tau_4]^T$ produced by the $T_{1..4}$ horizontal thrusters, using the so-called 'configuration matrix' T, defined as :

$$\tau = T \cdot f \tag{2}$$

T depends on the position (x_i, y_i) and orientation angle α_i of each thruster.

$$T = \begin{bmatrix} cos(\alpha_1) & cos(\alpha_2) & cos(\alpha_3) & cos(\alpha_4) \\ sin(\alpha_1) & sin(\alpha_2) & sin(\alpha_3) & sin(\alpha_4) \\ x_1 sin(\alpha_1) + y_1 cos(\alpha_1) & x_2 sin(\alpha_2) + y_2 cos(\alpha_2) & x_3 sin(\alpha_3) + y_3 cos(\alpha_3) & x_4 sin(\alpha_4) + y_4 cos(\alpha_4) \end{bmatrix} \quad (3)$$

2.3.3. Force applied by a single thruster

The force produced by a propeller depends on many parameters for which several models already exist, e.g., as in [Blanke, Lindegaard, and Fossen (2000)] and [Kim and Chung (2006)]. Under the quasi-stationary assumption, the force τ_t produced by a thruster can be approximated by the following equation [Fossen (2002)] :

$$\tau_t = k_1 |n| n - k_2 |n| u_a \quad (4)$$

where k_1 and k_2 are constant parameters depending on the propeller's design as well as the nozzle shape, n is the rotational speed of the propeller, u_a is the so-called advance speed and represents the speed of incoming water with respect to the thruster. As it is in practice very difficult to measure u_a, and assuming that ROVs are most of the time performing station-keeping tasks, the $k_2|n|u_a$ term is neglected in most controllers. However, in the present case, the thrusters orientation can be modified on-the-fly to maximize the thrust in a desired direction, so that they are likely to be sometimes nearly aligned. In this case, the one located downstream from the other is exposed to the high speed water flow ejected by the upstream one. This increases the incoming speed u_a for the downstream thruster, which may drastically reduce its thrust force as stated by equation (4). To be able to optimize the thrusters orientation, it is first needed to define a simplified model of this interaction as detailed in the next section.

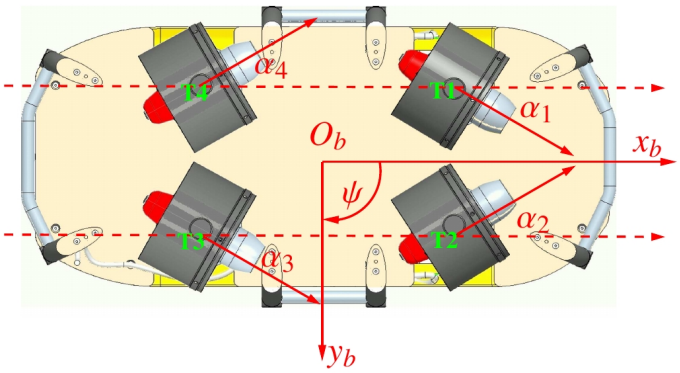

Figure 2. Position of the thrusters actuating the horizontal degrees of freedom of the ROV

2.4. Interactions

When thrusters are closed to each other as in an ROV, the effectiveness of propellers can be lowered by the interaction between cross flows. The phenomenon has been studied for dynamic positioning of surface ships, e.g. [Ruiz, Delefortrie, Vantorre, and Geerts (2012)], where propellers also interact with the surface or with the vessel's hull. This is especially observed when two propellers are set in tandem, as the water jet coming out of the front thruster feeds the tail propeller so that its apparent speed with respect to the fluid increases, therefore lowering the resulting thrust. Experimental results exist for thruster/thruster and thruster/hull interactions, such as in [Dang and Laheij (2004)] considering various diameters, distance and steering angle between tandem propellers. As a matter of interest, this latter reference points out that the thrust reduction of the downstream propeller is almost independent of the cruising speed, especially for low velocities.

As the reported works mainly deal with Z-drives open thrusters, an additional tank experiment using two Blue Robotics T200 thrusters[2] has been set up to roughly assess the interaction model with thrusters shaped for small ROVs.The setup consist of two thrusters (Figure 3). The first one, also called the "downstream" thruster, is a fixed reference (i.e. it cannot be oriented), it is attached at the bottom of a pole. The pole is hold in its middle by a frame. To measure the force (denoted F), a cable is attached at the top of the pole. For symmetry reasons, this force is equal to the force produced by the thruster. The second thruster is hold by a second frame, that can be rotated w.r.t the first frame, around a vertical axis aligned with the first thruster. Moreover, the pole that holds the second thruster can also be rotated w.r.t the second frame. Thus the system allows to test all the possible relative orientations between the two thrusters, while maintaining a constant distance between them. The naming convention of these relative angles is given by Figure 4a.

It is observed that the relation between the thrust attenuation and the relative steering angle between the propellers looks like a bell curve. The maximum attenuation factor is recorded when the thrusters are aligned, and becomes negligible when the projected area of the front thruster blows out of the downstream thruster's nozzle. The order of magnitude and shape of the recorded thrust reduction due to interaction is found to be compliant with the values recorded in the aforementioned references [Dang and Laheij (2004)] and [Arditti et al. (2015)]. Then, a simple interaction model considering that the thrust attenuation is a simple function of the downstream nozzle surface blown by the front thruster's stream (Figure 4b), can be designed.

Among others possible approximations, a Gaussian function is chosen as a convenient mathematical representation of the thrust attenuation factor. The attenuation factor of thruster T_i due to thruster T_j (as in Figure 4a) depends on a geometrical index G_{ji} accounting for the variations of the thrusters steering angles α_i and α_j with respect to the tandem angle θ_{ij} in the vehicle's frame

$$G_{ji} = G_{jia} \cdot G_{jib} = e^{-\frac{(\alpha_j - \theta_{ij})^2}{2\sigma_{ji}^2}} \cdot e^{-\frac{(\alpha_i - \theta_{ij})^2}{2\sigma_{ij}^2}}$$

where $\sigma_{ij} = \sqrt{\frac{\delta_{ij}^2}{2ln(0.05)}}$ allows to parameterize the geometric width of the interaction, i.e. as an approximation of the apparent area of the rear nozzle –limited by angle δ_{ij}– exposed to the incoming flow. The attenuation due to T_j on T_i can be expressed as $At_ji = G_{ji} \cdot max_At_{ji}$ where max_At_{ji} is the maximal attenuation when the thrusters are aligned, it depends on the distance between the propellers and on their respective active diameters [Dang and Laheij (2004)]. For example, the maximal attenuation recorded during experiments is about 0.5 for a 0.21m distance between the considered thrusters.

Finally, considering all possible interactions between the n thrusters acting on the vehicle and the nominal (undisturbed) thrust τ_{i0}, the total thrust τ_i provided by thruster T_i is

$$\tau_i = \tau_{i0} \cdot At_i = \tau_{i0} \cdot \left(1 - \sum_{i=1,\ i \neq j}^{n} max_At_{ji} \cdot G_{ji} \right)$$

assuming that there is no angular overlap between flows coming from several upstream thrusters –which assumption is verified in practice.

Compared with the efficiency functions given as plots in polar coordinates, as in [Arditti et al. (2015)], the attenuation model is here given as a single analytic function that makes it easy to implement inside an objective function to be optimized.

The model only relies on a few parameters related to the geometry of the thrusters and to their location on the vehicle, so that it is easy to adapt it for various configurations and propellers characteristics. In addition, it is easy to handle new experimental data as the measured features, i.e. the maximum attenuation and the bell curve width, can be directly translated in the model parameters max_At_{ji} and σ_{ji} for each thrusters pair. Finally, it is continuous, infinitely derivable and quasi-concave so that it is expected to behave efficiently with usual optimization algorithms.

[2] https://www.bluerobotics.com/store/thrusters/t200-thruster/

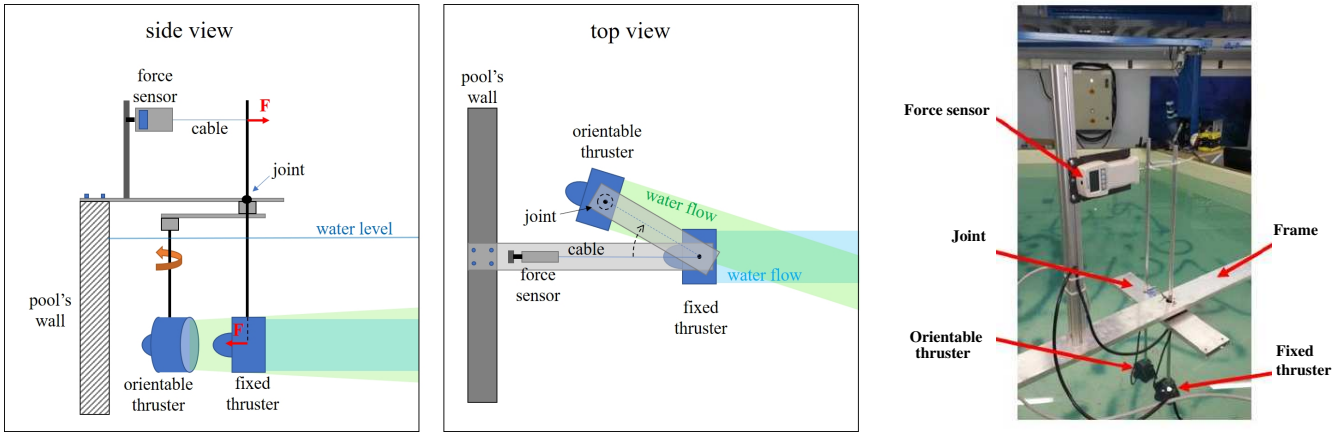

Figure 3. Experimental setup

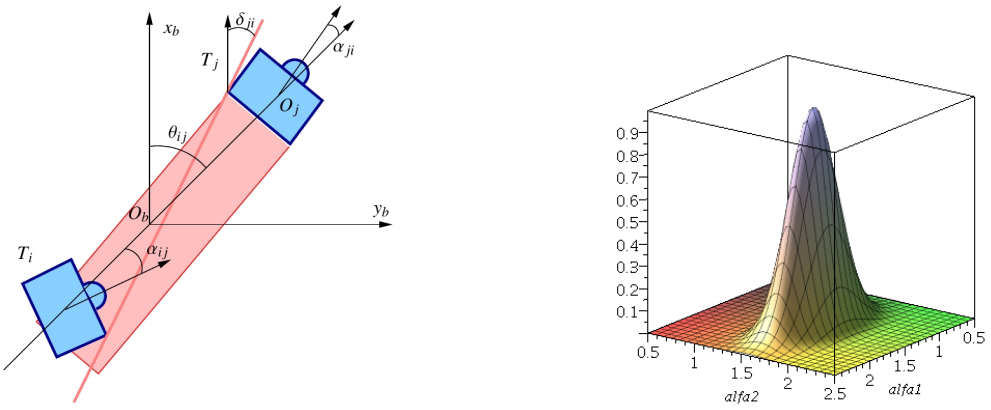

Figure 4. a) flow and nozzle interaction b) Gaussian model for G_{21} (attenuation on T_1 due to T_2)

3. Thrust optimization

The ROV motions in the horizontal plane are subject to several mission dependent constraints. Indeed, the payload (sonar, camera...) may impose a specific orientation of the vehicle with respect to the inspected structure. For instance, as depicted on Figure 5, while performing ship hull inspection with a multibeam imaging sonar, it is necessary to keep a given angle between the ship hull and the line of sight of the sonar to ensure the best observation conditions. Due to this practical constraint, during the inspection, the ROV's velocity will mainly be oriented in a given direction, denoted β_d on Figure 5. Note that, in case of current, V_d represents the relative velocity of the ROV with respect to the surrounding fluid.

Most existing ROVs have a so-called 'vectored' actuation, which means that their 4 horizontal thrusters have a fixed non null orientation (often 45°) with respect to O_x, thus allowing to produce thrust in any desired direction. However, this versatility has an important energetic cost. For instance, while moving straight forward, each thruster T_i produces $\tau_i\sqrt{2}/2$ along the O_x axis (assuming that $abs(\alpha_i) = 45°$), Thus only 70.7% of the overall produced thrust is useful, while the rest is lost in antagonistic forces. It is easy to understand that being able to change the thruster's orientation during the mission will reduce this energetic losses and will increase the velocity capabilities of the vehicle.

However, choosing the optimal azimuths for the thrusters is not as simple as steering them along the β_d direction. Not only the speed capability in the preferred direction needs to be handled, but some level of maneuverability along other directions must be preserved, basically for safety reasons, but also to compensate for disturbances like the tether drag or the sea currents. In what follows, several approaches are presented to optimize the azimuth of each thruster to maximize the speed capability for a preferred β_d direction, while preserving some level of omnidirectional maneuverability.

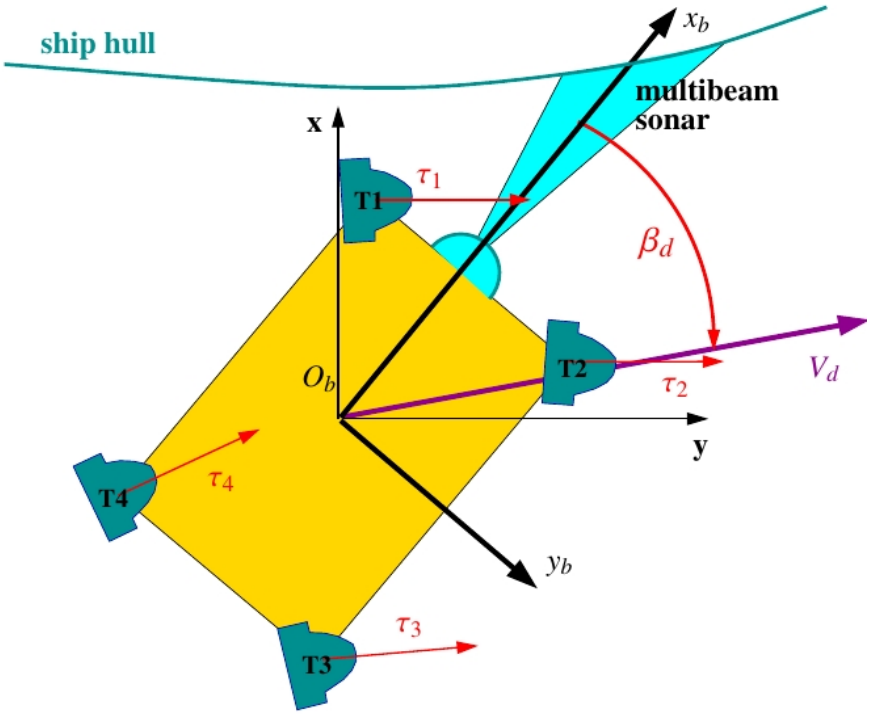

Figure 5. Thrusters oriented to maximize speed along β_d while maintaining some lateral maneuverability

3.1. Multiparametric optimization

In a first approach, the maximal thrust acting on the ROV in a given direction is computed as a joint optimization of both the actual generated forces and orientations of the thrusters, e.g. as already investigated in [Vega (2017)]. Hence, among various possibilities, one of the most effective cost functions to be minimized is found to be :

$$\underset{\alpha_i,\tau_i}{\arg\min} \quad \sum_{i=1}^{n}(At_i\tau i0)^2 \quad \text{s.t.} \left| \begin{array}{l} \arctan(F_y/F_x) - \beta_d = 0 \\ \sqrt{F_x^2 + F_y^2} = F_d \\ |\Gamma| \leq \Gamma_d \end{array} \right. \tag{5}$$

stating that the global thrust power is minimized considering that i) the global thrust direction is aligned with β_d ii) the global thrust acting on the vehicle is equal to a desired value F_d iii) the resulting generated torque is smaller than a desired very small value Γ_d (often chosen as 0), to avoid parasitic torques in nominal conditions.

Note that the optimization problem must be solved over eight variables, i.e. the four orientation thrusters azimuth angles α_i and the four actual thrusts τ_i resulting from the nominal thrusts τ_{i0} attenuated by the cross flows interactions At_i. After testing several available optimization methods, it rapidly appears that the problem has many local extrema disseminated everywhere in the search space. As a consequence, descent based algorithms inevitably fall in local maxima depending on the basin of attraction of the initial guess, and sets of close initial parameters often give different –thus unexploitable– solutions.

Hence solutions may be searched for using 'global optimization' algorithms likely to find the global extremum in the whole search space by avoiding being trapped in local extrema. This is for example the case of Simulated Annealing algorithms where a controlled random search around a guessed solution allows for escaping local extrema. Among others, Genetic Algorithms (GA) are currently popular algorithms for searching a global optimum in a large multi-variable non-convex optimization problem. They rely on a bio-inspired model based on an evolution strategy, where stochastic mutation and selection rules allow for iteratively searching better solutions from a population of initial candidates.

The force and orientation optimization problem has been encoded in C++ using the ISRES optimization algorithm [Runarsson and Yao (2005)], which is a part of the general purpose NLopt package [Johnson (n.d.)]. It runs a GA

and supports both equality and inequality constraints, therefore it is compliant with the thrust optimization problem. Stochastic rules and random search lead to quite long computation times (several seconds for an Intel i7/2 GHz chipset) and the same set of parameters and initial guess produce different –and sometimes obviously meaningless– solutions. Selecting an adequate initial population is a trial-and-error process leading to a trade-off between computation times and solutions credibility. For example, an initial population of 300 individuals and a 4 seconds timeout allows to find a likely solution in roughly 90% of trials. Hence, even if such an approach, based on combined force and orientation optimization, is very general and allows for, e.g., handling asymmetric vehicles, the apparently uncontrollable generation of meaningless solutions –which may happen when the computation time is constrained and a timeout is reached– encourages to investigate more deterministic methods.

3.2. Controllability approach

A counterpart of the former approach, where the optimization variables are both the actuators thrusts and orientations, is the complexity and computational cost of the optimization problem. Note that in any case the actuators configuration optimization only provides a nominal layout, and that the actual propulsion forces must be computed on-line by a feedback controller able to track the desired trajectory and to reject both the disturbances and the modeling errors.

Hence it is proposed here to split the control problem in two parts. Firstly find an efficient actuators layout using the controllability properties of the system; secondly design and apply a feedback controller using the enhanced actuators configuration. The controllability properties of the vehicle may be handled in several ways, using the controllability Grammian or using the velocity gains.

3.2.1. Velocity gains

Each actuator provides a contribution to the vehicle's motion through the thrust it gives along its orientation axis. This contribution can be modeled as the velocity gain, i.e. the ratio between the nominal thrust and the resulting steady state velocity. Considering the damping parameters D_u and D_v along the O_x and O_y axes of the vehicle, the overall velocity gains along the x and y axis are given by summing the gains of all n actuators

$$G_x = \frac{\sum_{i=0}^{n} At_i \cos(\alpha_i)}{D_u} \quad G_y = \frac{\sum_{i=0}^{n} At_i \sin(\alpha_i)}{D_v}$$

To maximize the thrust along the β_d desired direction, it is proposed to search for thrusters orientations α_i able to maximize the projection V_g of the velocity gains along the desired axis β_d, with V_g being defined as:

$$V_g = [G_x \cos(\beta_d) + G_y \sin(\beta_d)] \tag{6}$$

Maximizing V_g is not enough to find the best orientations of the thrusters. In fact several constraints must be respected:

Direction The $[G_x, G_y]^T$ vector's argument must be equal to the desired direction β_d. For this purpose the following constraint must be respected:

$$\arctan(G_y/G_x) - \beta_d = 0 \tag{7}$$

Symmetry around the yaw axis z The Γ_z torque induced by the thrusts of the four thrusters (when operated at nominal power) must be smaller than a threshold Γ_d. If it were not the case, a non negligible torque might lead to an undesired rotation of the robot. The choice of the Γ_d threshold must be done carefully. If too small, no solution can be found by a coarse grain optimization process as used in section 3.4. If too large, the controller should waste thrust just to nullify the resulting torque. In practice, it is found that $\Gamma_s = 10^{-3}$ N.m never made the algorithm fail. The expression of this constraint is:

$$|\Gamma_z| \leq \Gamma_d \tag{8}$$

Transverse gain The transverse maneuverability of the robot must be preserved. In what follows, the maneuverability is denoted ρ and is defined as the following ratio

$$\rho = \left| \frac{Gl_x \cos(\beta_d) + Gl_y \sin(\beta_d)}{V_g} \right| \tag{9}$$

with $Gl_x = \frac{\sum_{i=0}^{n} |At_i \sin(\alpha_i)|}{D_u}$ and $Gl_y = \frac{\sum_{i=0}^{n} |At_i \cos(\alpha_i)|}{D_v}$. This is the ratio of the velocity gain absolute value along β_d's orthogonal direction, over the velocity gain along β_d. This ratio must be greater than a threshold ρ_{min} defined by the user depending on the nature of the mission. Thus, the expression of this constraint is :

$$\rho \geq \rho_{min} \tag{10}$$

Torque gain To fight against the disturbing torques acting on the ROV, mainly due to the tether drag, it is needed to keep a high enough torque gain around the z axis. In fact, very low torque gains only happen when all the thrusters point towards the center of the vehicle. Indeed, as verified after optimizations, this configuration never occurs due to the interactions between the thrusters flows and the central body –so that "eye" configurations are always preferred over "X" configurations (Figure 6b). Therefore this constraint is never active and can be removed from the optimization process, as the torque control authority is verified to be always high enough to control the tether induced yaw disturbances –e.g., more than $80N.m$ achievable torque for the Tortuga case study– at the optimal steering solution. A positive consequence is a lighter complexity of the calculations, which is especially useful for the computationally costly brute-force method.

Therefore the whole velocity gain optimization problem can be written as

$$\arg\max_{\alpha_i} \quad [G_x \cos(\beta_d) + G_y \sin(\beta_d)] \quad \text{s.t.} \left| \begin{array}{l} \arctan(G_y/G_x) - \beta_d = 0 \\ \rho \geq \rho_{min} \\ |\Gamma| \leq \Gamma_d \end{array} \right. \tag{11}$$

3.2.2. Controllability Grammian

The classical Kalman's controllability criterion only provides a binary answer through the rank of the controllability matrix. The controllability Grammian gives a more subtle measure as it provides a sensitivity analysis of the dynamic system subject to control inputs along the directions of the state space [Chen (1999)].

For an asymptotically stable linear time invariant system $\dot{x} = A.x + B.u$ the symmetric and positive semi-definite controllability Grammian is given by

$$W_c = \int_0^\infty e^{A\tau} B B^T e^{A^T \tau} d\tau$$

W_c is invertible if and only if the system is controllable. Moreover, considering the impulse response of the system $X(t) = e^{At}B, \quad t > 0$, it appears that the Grammian is the integral of the square of the impulse response as $W_c = \int_0^\infty X(t)X^T(t)\, d\tau$. As such it provides quantitative information about the input energy needed to move the system state towards specific directions.

For run-time efficiency the controllability Grammian of the vehicle is explicitly derived from a linearization of the symbolic model given in section 2.2. The added masses are accounted through $m_u = m + X_{\dot{u}}, \quad m_v = m + Y_{\dot{v}}, \quad I_{z+} =$

$I_z + N_{\dot{r}}$, denoting $\mathcal{C}_i = \cos(\alpha_i), \mathcal{S}_i = \sin(\alpha_i)$. $\mathcal{D}i = x_i\mathcal{S}_i - y_i\mathcal{C}_i$ for concision.

$$W_c = \left[G_{ij}\right] = \begin{bmatrix} \frac{1}{2D_u m_u}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{C}_i^2 & \frac{1}{D_u m_v + D_v m_u}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{C}_i\mathcal{S}_i & \frac{1}{D_u I_{z+} + D_r m_u}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{C}_i\mathcal{D}_i \\ \frac{1}{D_u m_v + D_v m_u}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{C}_i\mathcal{S}_i & \frac{1}{2D_v m_v}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{S}_i^2 & \frac{1}{D_v I_{z+} + D_r m_v}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{S}_i\mathcal{D}_i \\ \frac{1}{D_u I_{z+} + D_r m_u}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{C}_i\mathcal{D}_i & \frac{1}{D_v I_{z+} + D_r m_v}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{S}_i\mathcal{D}_i & \frac{1}{2D_r I_{z+}}\sum_{i=1}^{4}\mathcal{A}t_i^2\mathcal{D}_i^2 \end{bmatrix}$$

To enhance the controllability of the vehicle along the desired direction defined by angle β_d, it is proposed to maximize the composition of the first two diagonal components of the Grammian (G_m), under constraint that the resulting thrust must be aligned with direction β_d as

$$\arg\max_{\alpha_i} \quad \sqrt{(G_{11}\cos(\beta_d))^2 + (G_{22}\sin(\beta_d))^2} \quad \text{s.t.} \left| \begin{array}{l} \arctan(G_y/G_x) - \beta_d = 0 \\ \sum_{i=1}^{n} At_i \sin(\alpha_i - \beta_d) \geq G_{tr}min \\ G_\theta \geq G_\theta min \\ |\Gamma| \leq \Gamma_d \end{array} \right. \tag{12}$$

where G_y and G_x are the velocity gains defined for the velocity gain case. The first constraint ensures that the global nominal resulting thrust is oriented along the desired direction β_d. Some extra constraints can be added to the initial set, e.g. to keep a minimal controllability along the transverse and yaw axis of the vehicle ($G_{tr}min$ and optionally $G_\theta min$ respectively). This is needed to account for the modeling uncertainties and to be able to reject the perturbations due to external forces. Finally a constraint stating that the nominal torque must be smaller than the very small Γ_d value enforces symmetric solutions.

3.3. Optimization methods

Searching the values of the α_i orientations to maximize the cost function under constraints (G_m (12) or V_g (11)) needs to find effective optimization methods. Note that both cost functions are non-linear and are subject to both equality and inequality constraints. Moreover the solution must be provided within a short time to be compliant with the environmental conditions and mission execution dynamics. Several methods and software packages have been reviewed and evaluated to cope with the above constraints [Parkinson, Balling, and Hedengren (2013)]. Hereafter, a two steps method is proposed. The first step uses the velocity gains to explore the performance of all the possible thrusters configurations for a sparse set of β_d values, while the second step uses the values obtained by the first method to initialize a real-time compliant local optimization algorithm for intermediate β_d values.

3.4. Off-line brute force and Look-Up Table generation

The first approach to optimize the α_i orientations of the thrusters is based on brute force. This method consists in computing the ROV's velocity gains for all possible α_i orientations of the four horizontal thrusters. Furthermore, the interactions between thrusters are accounted, as well as the interactions of the thrusters' water flows with the central body of the robot.

During the search process, the thrusters are set to their maximal power and all positions are tested in the range $[\beta_d - 60°: \beta_d + 60°]$, with an angular step of δ °. For each configuration, the G_x and G_y velocity gains, defined in section 3.2.1, are computed and, if the constraints defined in the previous section are satisfied, the configuration is considered as 'best configuration candidate' and stored if the gain is larger than the previously stored value.

This method is assumed to find the best global solution, provided that the sampling grid is fine enough. The counterparts are its computational cost and its slowness. For instance, running the C++ program on an Intel® Core™i7-5600U 2.60GHz CPU takes about 20 minutes for each desired β_d set point, with $\delta = 1°$and $\Gamma_d = 10^{-3}Nm$. As this is not compliant with mission and vehicle dynamics, a Look-Up Table (LUT) has been filled with pre-computed values. In this LUT, the best $\{\alpha_i\}_{i=1..4}$ thruster configurations are computed for $\beta_d \in [0°, 90°]$, with $\delta = 10°$steps. For each value of β_d, the set $\{0.2, 0.4, 0.6, 0.8, 1\}$ of values for maneuverability ratios are considered.

Table 1. Look-up table obtained with the brute force method. In the last column, the velocity gain over the classical vectored configuration is computed.

β_d	maneuv. ratio ρ_{min}	best orientations α_i, $i \in \{1..4\}$				gain w.r.t vectored config.
0°	$0.2 \cdot D_u/D_v$	12°	−11°	12°	−11°	+38.6%
	$0.4 \cdot D_u/D_v$	21°	−23°	21°	−23°	+31.1%
	$0.6 \cdot D_u/D_v$	31°	−31°	31°	−31°	+21.2%
	$0.8 \cdot D_u/D_v$	39°	−39°	39°	−39°	+9.9%
	$1 \cdot D_u/D_v$	45°	−45°	45°	−45°	0%
50°	$0.2 \cdot D_u/D_v$	66°	39°	58°	38°	+89.9%
	$0.4 \cdot D_u/D_v$	−9°	108°	59°	36°	+44.6%
	$0.6 \cdot D_u/D_v$	−9°	108°	59°	36°	+44.6%
	$0.8 \cdot D_u/D_v$	106°	3°	82°	13°	+21.7%
	$1 \cdot D_u/D_v$	106°	3°	82°	13°	+21.7%

For every solution of the LUT, the velocity gain of the optimized configuration over the so-called 'vectored configuration' is computed. This gain is defined as the ratio of the maximal achievable speed in β_d direction obtained with the optimization method, over the maximal achievable speed obtained with the 'vectored configuration'. The latter, where each thruster's orientation is fixed at 45°, is a frequent commercial solution to provide a convenient maneuverability ratio equal to 1. Even if some commercial vehicles allow manual steering of the thrusters before the mission, thus allowing to change the maneuverability ratio, the 45°is chosen as a reference as it is the more frequently used.

Table 1 is extracted from the LUT and illustrates the results obtained for two significant values of β_d, namely $\beta_d = 0°$and $\beta_d = 50°$. With the first value ($\beta_d = 0°$), the vehicle is better shaped for straight speed. There is no interaction between the thrusters' flows and the vehicle's hull, thus leading to symmetrical orientations of the thrusters. When the maneuvering ratio is set to $\rho_{min} = 1$, the same speed capability is expected in the longitudinal and transverse directions. In this case, the algorithm finds that the best orientations of the thrusters are $[\alpha_1, \alpha_2, \alpha_3, \alpha_4] = [45, -45, 45, -45]°$(for a "squared" ROV where $D_u = D_v$), which corresponds to the standard vectored actuation. In this case, naturally, the gain over the vectored configuration is null, but in every other cases the gain is positive and its value depends on the maneuvering ratio ρ_{min} defined by the user. Note that the maneuverability gain in Table 1 is scaled by the Du/Dv ratio to ease comparisons with the standard vectored configuration.

The case $\beta_d = 50°$has been chosen as it corresponds to the case where the maximal gain over vectored actuation is observed (+89.9%), when the maneuvering ratio ρ_{min} is set to 0.2. Even for $\rho_{min} = 1$, which corresponds to the same maneuverability as a vectored system, the gain in achievable velocity is +21.7%, thanks to the optimized choice of α_i values. Finally, one can also remark that for $\beta_d = 50°$the configuration is not always symmetrical. This illustrates the fact that the algorithm takes into account the thruster interactions and the effect of their water flows when directed to the hull. For the entire LUT, the average gain in achievable velocity is +42.6%, which clearly demonstrates the advantage of the proposed method over conventional fixed vectored configuration, even when the same level of maneuverability is expected.

During computation, all the thrusters are set to their maximal value to compute the overall forces and torque applied to the vehicle. Considering the force produced by each thruster as an input variable of the problem might have been a more precise way, as already proposed in section 3.1. However this would lead to a much higher computational cost complexity of the search process, thus dramatically increasing the computation time.

Although the LUT can be easily stored after been computed, and then can be immediately accessed to find the best configuration, it only provides results for a sparse set of values of β_d and ρ. Therefore, in the next section an efficient way for fast extrapolation of intermediate values of β_d and ρ is proposed, allowing to access an infinity of combinations in a very short time. However, the LUT is a mandatory pre-requisite as it provides initial solutions close enough to the optimal solution, so that the optimization process is started in a concave vicinity of the solution. Indeed, without being initialized with the sparse values of the LUT, the local optimization could be trapped in a local extremum far from the global one.

3.5. *Fast on-line local optimization*

Traditional optimization of analytic cost functions uses some kind of gradient (1st order) or Hessian (2nd order) descent algorithm, following the slopes of the cost function from the current point until the considered derivatives becomes null when reaching the extremum. These deterministic methods are very efficient and well suited for convex optimization problems, where there is only one extremum in the search space. Otherwise, for complex problems for which there are numerous local extrema, the method stops at the first found extremum. Among descent based algorithms, direct search methods are derivative free algorithms that do not require an explicit computation of the derivatives of the cost function.

Although optimization problems of complex real size plants most often have many local extrema –as in problem (5)– such methods may be applied if the search space can be reduced enough to contain few local extrema and if the optimization algorithm can be initialized in the basin of attraction of a satisfactory solution. The maximization of the V_g velocity gains or of the G_m Grammian of the ROV fall in this case.

If there are no disturbances between the thrusters (i.e. $At_i = 1\ \forall i$), the components of the gradient of the objective function V_g (11) can be written as

$$\frac{\partial V_g}{\partial \alpha_i} = \frac{\sin(\beta_d)\cos(\alpha_i)}{D_v} - \frac{\cos(\beta_d)sin(\alpha_i)}{D_u}$$

leading to the optimal direction for every thruster $\alpha_i^* = \arctan[\tan(\beta_d)\frac{D_u}{D_v}]$, i.e. the desired direction for global thrust β_d weighted by the ratio of the drag factors along the O_x and O_y axis.

Accounting that the interaction limit angles δ_{ij} between thrusters are quite small with usual ROV architecture, it is assumed that the optimal thrusters configuration considering interactions is not very far from the optimal one without interactions, which therefore can be used as a convenient initial solution. Another possible initial guess can be given by a coarse grain LUT computed off-line, as described in section 3.4.

In addition, it is assumed that, in a vicinity of this initial configuration, a given downstream thruster can be disturbed by at most one upstream thruster. Therefore, the optimization problem feasibility can be analyzed observing only pairs of interacting thrusters.

Thanks to xmaple[3], Figure 6a shows the reduced objective function V_{g_2} with two variables α_1 and α_2, considering the two thrusters T_1 and T_2 in interaction around their alignment direction $\beta_d = \theta_{21} = \pi/2$.

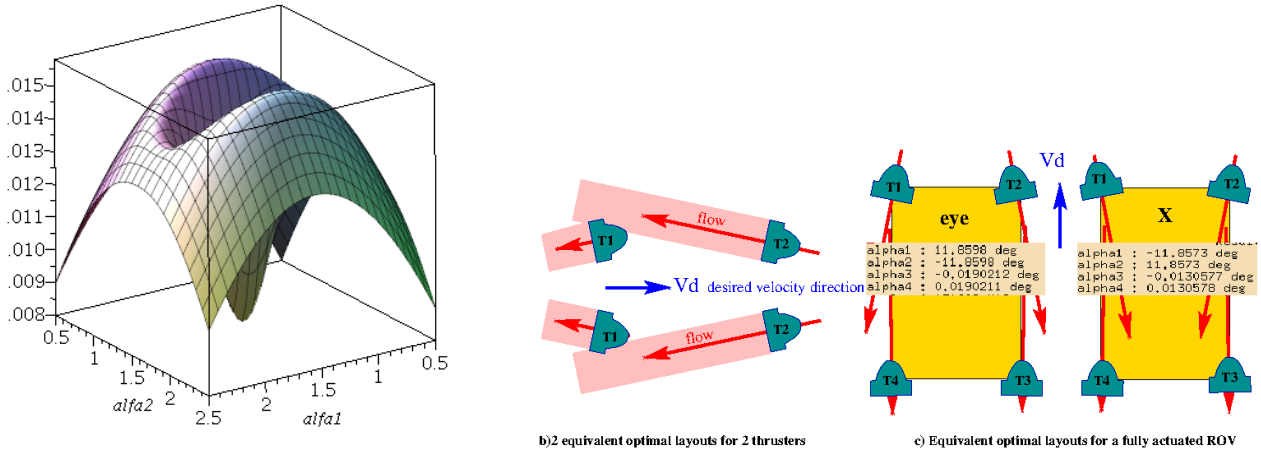

Figure 6. a) Objective function V_{g_2} for T_1 and T2 around $\beta_d = \pi/2$ rad and b) corresponding optimal steering for T_1 and T2 in isolation c) Optimal steering for 4 thrusters (eye on left, X on right)

This function has a local minimum when the thrusters are perfectly aligned, i.e. when the downstream thruster is directly blown by the upstream one. There are two local maxima with equal values on both sides of the thrusters alignment direction. They correspond to null-torque thrust optimal solutions for the steering of a tandem of two thrusters in isolation (Figure 6b).

[3] https://www.maplesoft.com/

Finally, considering the full vehicle, optimal solutions are found to be either in a "eye" or in a "X" thrusters layout (Figure 6c). Additional constraints may be used to favor a particular solution, e.g. to avoid a possible interaction between the thrusters and the central body of the ROV. The Grammian based objective function G_m behaves similarly.

Note that these objective functions show large quasi-concave regions around the optimal solutions. The optimization process converges to convincing solution even when starting from initial value as far as $\pi/2$ from the optimum, except for some particular isolated desired angle β_d where the basin of attraction is reduced, possibly due to numerical artifacts and multiple interaction between thrusters. Although an analytic formulation of Hessian of the objective function can be developed using formal tools (Maple and Matlab), further formal analysis of the Hessian properties exceeds the capabilities of currently available computers and software. However a numerical evaluation of the Hessian negativeness around the optimal solutions α_i^*, $i = 1,...,4$ given by the LUT defined in the previous section show that V_g is actually concave (all its eigenvalues are negative) around at least 0.2 rad far from the optimal steering ($\alpha_i^* - 0.2 < \alpha_i < \alpha_i^* + 0.2, \quad i = 1,...,4$). Indeed descent based optimization methods can be efficiently used with initial solution (α^*) given by a LUT as computed in section 3.4.

When selecting a local deterministic optimization algorithm, it can be chosen inside two sets according to the availability of the cost-function gradient. Gradient-based optimization algorithms need an explicit expression of the cost-function gradient. This is the case of the widely used SQP algorithm [Powell (1978)] which is known to provide fast solutions for optimization problems. However deriving an analytic expression of the cost-function gradient can be difficult when the cost-function is not given as a quadratic function of the optimization variables. Then derivative-free optimization algorithms, such as simplex-based ones, can be used to avoid the explicit computation of the cost-function gradient, usually at the cost of a slower convergence to the optimal solution.

Both cost functions V_g and G_m have been successfully implemented calling algorithms in the C++ version of the aforementioned NLopt optimization software package. Inside NLopt, two local optimization algorithms directly handling both equality and inequality non-linear constraints are available. SLSQP (Sequential Least Square Quadratic Programming) [Kraft (1994)] is a particular implementation of SQP, for which the gradients of V_g and G_m needed to be derived combining Maple and Matlab. Simpler and easier developed implementations are provided using COBYLA (Constrained Optimization BY Linear Approximations) [Powell (1998)], a local optimization derivative free algorithm.

Both algorithms run fast enough –i.e. in milliseconds or less with off-the-shelp laptops– so that both are convenient for the ROV application, and the even faster computations given by SQP are not necessary. Therefore the preferred algorithm is COBYLA, as it allows for a simple implementation and fast testing of new cost functions, avoiding the complex explicit formulation of the cost function gradient, and providing run-time code with smaller footprint. Running on a Linux laptop with i7 2 GHz chipset, and after being initialized using the LUT outputs (see Table 2 for a sample of results), optimal solutions are found within 1-2 ms for both cost functions, and no decisive advantage for V_g or G_m was observed until now. See for example Table 2 for a sample of results. To compare with, running the aforementioned ISRES GA from the same NLopt package always needs several seconds to provide –sometimes unsound– solutions after a long trial-and-error procedure to find effective tuning parameters.

Table 2. Optimal configurations given by COBYLA for $0 <= \beta_d <= 20°$and $\rho = 1$

β_d	α_{init} from LUT	α_1	α_2	α_3	α_4
0.0°	45°, −45°, 45°, −45°	45.0°	−45.0°	45.0°	−45.0°
1.0°	—-	46.8234°	−43.507°	46.1539°	−45.4857°
2.0°	—-	48.0996°	−43.5806°	47.8306°	−44.3519°
3.0°	—-	49.8186°	−42.3043°	49.0182°	−44.5527°
4.0°	—-	50.9131°	−42.7334°	50.817°	−42.997°
5.0°	—-	52.4024°	−41.9916°	52.179°	−42.5913°
6.0°	55°, −35°, 55°, −35°	52.8238°	−43.9855°	54.5375°	−39.46°
7.0°	—-	54.5752°	−42.4327°	55.605°	−39.7767°
8.0°	—-	56.5241°	−40.3634°	56.4328°	−40.5938°
9.0°	—-	58.7053°	−37.6829°	56.9459°	−42.0479°
10.0°	—-	58.9781°	−39.7322°	59.4168°	−38.6675°

4. Control considerations

When the sea current orientation is changing during a mission, or if the ROV's task requires a new orientation of the thrusters, the reconfiguration must be done on-the-fly without loosing the controllability of the robot.

In what follows, it is assumed that the maneuverability ratio ρ has been defined by the user prior to the mission. Then, the main parameter to be determined is β_d along the mission. There are several ways to determine β_d, as for instance :

- If the (x, y) position of the ROV is not measured (open-loop piloting), it is proposed to keep the ROV in a standard vectored configuration as long as the requested velocity –as given by a joystick– remains below a given threshold. When the pilot requests –through the joystick– a velocity value exceeding the thresholds, β_d can be computed as $\beta_d = \arctan(y_{joystick}/x_{joystick})$. Disturbances and deviations are compensated using the joystick thanks to the pilot visual feedback;
- If the (x, y) position and velocity of the ROV is estimated (by a Doppler Velocity Log or by an Inertial Measurement Unit), an estimator can be used to evaluate the force and direction of the sea current or of any other slow varying directional disturbance (e.g. tether drag). The estimated direction of such disturbances, combined with the heading provided by the ROV's magnetometer, can be used to define β_d along the mission, e.g., such as described in (Hegrenaes and Hallingstad (2011)).

Once β_d has been estimated and the corresponding α_i thrusters' orientations have been optimized, the T configuration matrix (3) can be computed, and then used to compute $f = [\tau_1, \tau_2, \tau_3, \tau_4]^T$ the forces that the thrusters must generate :

$$f = T^+ \cdot \tau \tag{13}$$

where T^+ is the Moore-Penrose pseudo-inverse of T and τ (defined in 2.3.2) is the input computed by the controller to servo each controlled degree of freedom.

Finally, the control input of each thruster is computed taken into account the thrust attenuation due to other thrusters' flows and due to the interaction of the thruster's flow with the ROV's hull. For this purpose, the attenuation coefficient At_i of each thruster is computed according to the attenuation model described in section 2.4, to compensate the attenuation by updating the controller gains.

The optimization of the thrusters' orientations has to be performed at a quite low frequency compared to τ updates, the low-level control input of the ROV in charge of servoing the degrees of freedom of the robot. In fact, the overall control scheme is made of an inner high-frequency control loop, in charge of servoing the position and attitude of the ROV, and an outer lower-frequency optimization loop sequentially reading the LUT (for initialization) and runing COBYLA (for fine grain optimization) to maximize the thrust performance in the privileged β_d direction. Of course, to allocate the thrusts, the inner-loop controller takes into account the orientation values delivered by the outer-loop optimization module.

Due to the fact that, during the optimization, some orientations are avoided because of the interactions between thrusters, discontinuities in the α_i values are observed, even if β_d varies continuously. For this reason, it is also necessary to implement an hysteresis in the reconfiguration process to avoid chattering in the thrusters' orientations. This may lead to a transient drop in performance (decreasing the maximal achievable speed in direction β_d), which anyway remains above the performance of a fixed vectored actuation while keeping the robot under control at any time.

5. Conclusion

This paper describes an effective method for on-the-fly reconfiguration of an ROV equipped with steerable thrusters. The goal is to optimize the velocity of the vehicle in a specific direction during underwater operations, under varying environmental and operational conditions. The proposed method first models the flows interaction between the thrusters. Then it proceeds along a two-steps optimization process. Firstly a pre-computed Look-Up Table (LUT) provides solutions for a sparse set of parameters, i.e., the desired direction and maneuverability ratio. The LUT is further used to initialize a local optimization algorithm running on-line. Several actuation constraints, for instance holding

a minimal transverse maneuverability, are accounted during optimizations. The results is an optimized thruster configuration computed in a short time compliant with operational constraints at sea. The average velocity gain over a conventional 45°vectored actuation is 42.6%. Some control considerations are sketched to explain how to handle online re-configurations in the control scheme and to give some pointers about the estimation of the desired direction for which the velocity must be optimized.

Acknowledgements

The authors thank Subsea Tech for providing the technical data of Tortuga 500 underwater vehicle, as well as the CAD views used in certain figures.

Funding

This project has been funded with support from ANRT (French National Agency for Research and Technology, CIFRE grant #2015/1232).

References

Arditti, F., Souza, F., Martins, T., & Tannur, E. (2015). Thrust allocation algorithm with efficiency function dependent on the azimuth angle of the actuators. *Ocean Engineering*, *105*, 206–216.

Benetazzo, F., Ippoliti, G., Longhi, S., Raspa, P., & Sørensen, A. (2012). Dynamic positioning of a marine vessel using dtvsc and robust control allocation. *20th Mediterranean Conference on Control and Automation (MED) Barcelona, Spain, July 3-6*.

Blanke, M., Lindegaard, K.-P., & Fossen, T.-I. (2000). Dynamic model for thrust generation of marine propellers. In *5th IFAC conference on manoeuvring and control of marine craft* (pp. 363–368). Aalborg, Denmark.

Chen, C.-T. (1999). *Linear system theory and design* (Third ed.). New York: Oxford University Press. (ISBN 0-19-511777-8.)

Dang, J., & Laheij, H. (2004). Hydrodynamic aspects of steerable thrusters. In *Dynamic positioning conference,*. Houston, USA.

Ding, F., Yang, D., & Huang, W. (2016). Study on stabilizing power of dynamic positioning ship based on thrust allocation. *IEEE International Conference on Mechatronics and Automation*.

Fossen, T. (2002). *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. Trondheim, Norway: Marine Cybernetics. (ISBN 82-92356-00-2)

Hegrenaes, Ø., & Hallingstad, O. (2011, April). Model-aided ins with sea current estimation for robust underwater navigation. *IEEE Journal of Oceanic Engineering*, *36*(2), 316-337.

Johansen, T. A., Fossen, T. I., & Berge, S. P. (2004). Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. *IEEE Transactions on Control Systems Technology*, *12*(1).

Johansen, T. A., Fuglseth, T. P., Tondel, P., & Fossen, T. I. (2008). Optimal constrained control allocation in marine surface vessels with rudders. *Control Engineering Practice*, *16*, 457–464.

Johnson, S. G. (n.d.). *The NLopt nonlinear-optimization package.* `http://ab-initio.mit.edu/nlopt`.

Kim, J., & Chung, W. (2006). Accurate and practical thruster modeling for underwater vehicles. *Ocean Engineering*, *33*(5-6), 566–586.

Kraft, D. (1994). Algorithm 733: TOMP-fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software*, *20*(3), 262–281.

Maalouf, D., Creuze, V., Chemori, A., & Tempier, O. (2015, 01). Feedforward inertial actuation for roll stabilization of an underactuated underwater vehicle. *International Journal of Robotics and Automation*, *30*, 2015.

Moreno, E., & Chung, S.-Y. (2014). SeaDrone: A modular and reconfigurable underwater robot for task optimization. In *Oceans*. Taipei, Taiwan.

Odetti, A., Bilbuli, M., Bruzzone, G., Caccia, M., Spirandelli, E., & Bruzzone, G. (2017). e-URoPe: a reconfigurable AUV/ROV for man-robot underwater cooperation. In *20th IFAC World Congress*. Toulouse, France.

Parkinson, A.-R., Balling, R., & Hedengren, J.-D. (2013). *Optimization methods for engineering design*. Brigham Young University. http://apmonitor.com/me575/uploads/Main/optimization_book.pdf.

Patel, D., Frank, D., & Crane, C. (2014). Controlling an overactuated vehicle with application to an autonomous surface vehicle utilizing azimuth thrusters. In *14th Int. Conf. on Control, Automation and Systems ICCAS 2014*. Seoul, Korea.

Powell, M. J. D. (1978). A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson (Ed.), *Numerical analysis* (pp. 144–157). Berlin, Heidelberg: Springer.

Powell, M. J. D. (1998). Direct search algorithms for optimization calculations. *Acta Numerica*, *7*, 287–336.

Pugi, L., Allotta, B., & Pagliai, M. (2018, 01). Redundant and reconfigurable propulsion systems to improve motion capability of underwater vehicles. *Ocean Engineering*, *148*, 376-385.

Ribas, D., Palomeras, N., Ridao, P., & Mallios, A. (2012). Girona 500 AUV: From survey to intervention. *IEEE/ASME Trans. on Mechatronics*.

Ruiz, M., Delefortrie, G., Vantorre, M., & Geerts, S. (2012). Propulsion and steering behaviour of a ship equipped with two contra-rotating Z-drives. *10th International Conference on Hydrodynamics October 1-4, 2012 St. Petersburg, Russia*.

Runarsson, T. P., & Yao, X. (2005). Search biases in constrained evolutionary optimization,. *IEEE Trans. on Systems, Man, and Cybernetics Part C: Applications and Reviews*, *35*(2), 233–243.

Vega, E. P. (2017). *Task-based design and optimization of reconfigurable propulsion systems for autonomous underwater vehicles* (Doctoral dissertation). Université de Bretagne Occidentale (UBO), Brest.

Wang, Y., Gu, J., & Zou, C. (2013, Sept). Thrust allocation in dynamic positioning system based on particle swarm optimization algorithm. In *2013 oceans* (p. 1-6). San Diego, CA, USA.

Xu, R.-H., Wang, Q.-R., Song, Y.-N., Zheng, R.-R., & Chen, M.-J. (2011). Study on ship dynamic positioning system's thruster allocation based on genetic algorithm. *International Conference on Information Science and Technology March 26-28, Nanjing, Jiangsu, China*.