

A Dynamic Manipulation Strategy for an Intervention: Autonomous Underwater Vehicle

Ridolfi A*, Conti R, Costanzi R, Fanelli F and Meli E

Department of Industrial Engineering, Mechatronics and Dynamic Modelling Laboratory, University of Florence, Italy

Abstract

This paper presents the modelling and the control architecture of an Autonomous Underwater Vehicle for Intervention (I-AUV). Autonomous underwater manipulation with free-floating base is still an open topic of research, far from reaching an industrial product. Dynamic manipulation tasks, where relevant vehicle velocities are required during manipulation, over an additional challenge. In this paper, the accurate modelling of an I-AUV is described, not neglecting the interaction with the fluid. A grasp planning strategy is proposed and integrated in the control of the whole system. The performances of the I-AUV have been analysed by means of simulations of a dynamic manipulation task.

Keywords: Autonomous underwater vehicle; Underwater manipulation; Dynamic manipulation; Underwater robotics

Introduction

The development of the Autonomous Underwater Vehicles (AUVs) has a great importance in modern society, because of their fundamental role in the military field, in underwater explorations (e.g. archaeological field) and in the industrial field (e.g. Oil and Gas applications). These devices, efficient and reliable, can undoubtedly lead to substantial economic and technological benefits. More specifically, in the technical evolution of the AUVs the following important topics are still characterized by many open problems: the dynamic performances and the control of the vehicle, the mobile tele-manipulation of a single vehicle and the cooperation among vehicles (whether including the manipulation operations or not). In this paper the modelling and the control architecture of an AUV specifically thought for the underwater mobile manipulation, usually called I-AUV (Intervention-AUV), are described.

Nowadays, a considerable number of operations in sea-rescue, research and maintenance of oil rig appliances, got ahead using Unmanned Underwater Vehicles (UUVs), need manipulation capacity to be concluded successfully. In such scenario, most of the intervention missions at high depths are faced up by remotely controlled vehicles equipped with one or more robotic arms (Intervention-ROVs), representing until today the standard technology in that field [1]. The ROVs for assistance, which can be tele-operated for long periods, are usually controlled with a master-slave approach [2,3]. This kind of strategy has some limitations: the operator must be skilled with special type of training, underwater communication is often difficult and a significant delay in control can be present. If the operation has to be carried out at elevated depth, it is also necessary the presence of an ocean ship equipped with a heavy crane to lift the ROV when it emerges with important costs of the mission. All these problems, to which is added the presence of a long communication and power supply cable, restricting the possible movements of the vehicle and of its arm, have recently brought some researchers to consider what appears as the AUV natural evolution, i.e. the autonomous underwater vehicles equipped with manipulator arms, the I-AUVs [4,5].

The first attempt to realize an autonomous underwater vehicle equipped with a robotic arm dates back to the nineties and traces back to the development of ODIN and OTTER vehicles [1], respectively

at Hawaii University and Stanford Robotic Lab. Both vehicles are completely actuated and have a simple manipulator with one degree of freedom, assigned to automatic rescue operations of submerged objects. Immediately afterwards, at French Ifremer, VORTEX (Versatile and Open subsea Robot for Technical Experiments) was shown [6], an under-actuated vehicle with 5 degrees of freedom (DOFs), equipped with a 7 link arm of Mitsubishi PA-10 type, born as ROV and usable in an automatic mode too. Such vehicles represent no doubt a step forward for the I-AUV technology, even if they are exclusively used in controlled conditions and environments, particularly for the study of the hydrodynamic model, of the manipulator arm and for considerations concerning the controllability of the hybrid system. In 1997 the European project AMADEUS (Advanced Manipulation for Deep Underwater Sampling) was approved; the project involved many University partners, among whom University of Genova [7]. The project was focused on the coordinate control of two Ansaldo robotic arms at 7 DOFs mounted on a tele-operated ROV, in order to make an underwater cell able to pick samples and carry out complex manipulation operations on the seabed. In 2004 the development of this project led to the realization of SAUVIM (Semi-Autonomous Vehicle for Intervention Missions), Hawaii University, shown in Figure 1, the first underwater vehicle to try underwater manipulation operations on the open sea [8].

SAUVIM is equipped with a manipulator arm with 7 links, the same used during AMADEUS project, which is controlled in an uncoupled way as to the vehicle body. This one is moved according to 6 degrees of freedom by 8 propellers and has a strong weight difference compared to the arm, a characteristic making possible the decoupling control of the two elements. The vehicle is just semi-autonomous, although the

***Corresponding author:** Ridolfi A, Department of Industrial Engineering, Mechatronics and Dynamic Modelling Laboratory, University of Florence, Italy, Tel: +0039 055 2758764; E-mail: a.ridolfi@unifi.it

Received August 18, 2015; **Accepted** August 25, 2015; **Published** August 31, 2015

Citation: Ridolfi A, Conti R, Costanzi R, Fanelli F, Meli E (2015) A Dynamic Manipulation Strategy for an Intervention: Autonomous Underwater Vehicle. Adv Robot Autom 4: 132. doi: [10.4172/2168-9695.1000132](https://doi.org/10.4172/2168-9695.1000132)

Copyright: © 2015 Ridolfi A, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



Figure 1: SAUVIM vehicle, with Ansaldo robotic arm.



Figure 2: I-AUV in the framework of TRIDENT project.

supply takes place through batteries, because it is equipped with a cable allowing communication during the operation: the user specifies just instructions at high level, on the basis of the outputs of the vision sensor and the other on board sensors, while the robot runs in an autonomous way the low level orders, which are necessary to perform the mission. SAUVIM is till today used for the recovery of objects from the seabed or for other dexterous manipulation operations, as it can sink at a maximum depth of 4000 m. The vehicles, developed up to that moment, were large-sized and heavy, able to work at high depth and in ocean environments: A special interest was consequently given to the development and realization of smaller and lighter I-AUVs (with mass below 300 kg), used for intervention up to a depth of 500 m, among which ALIVE vehicle (Autonomous Light Intervention Vehicle) [9], born within the framework of the FP5 European project growth.

One important contribution to the development of the state of the art of the I-AUV is finally due to TRIDENT (Figure 2), an European project lasting for 3 years and started in 2010 [10]. The aim of TRIDENT was the development of new methodologies to complete manipulation assistances in non-structured underwater environments, through a cooperative team composed of an AUV equipped with a robotic arm at 7 DOFs and an ASC (Autonomous Surface Craft): the latter is an autonomous surface vehicle, whose aim is to replace, in the application near to the coast, the ship with crew necessary to the running of the AUV.

Very few examples of preliminary experience with autonomous free-floating grasping exist (rather than manipulation) but this was demonstrated in a harbour in the framework of the TRIDENT project

[11]. In March 2014 PANDORA project [12] has demonstrated free floating grasping and valve turning in tank. However in both cases the vehicle is in a hovering phase and not in “mobile navigation”. Anyway, to be realistic, autonomous underwater robotic manipulation with free-floating base is far from reaching an industrial product. This is particularly true in the framework of dynamic manipulation, where relevant vehicle velocities are required (in contrast with hovering manipulation).

As regards the control techniques of such vehicles, the problem is still open. Up to now decoupled controls for the AUV subsystem and the manipulator have been mostly studied [1,13], assuming the weight difference between the two elements is wide enough, and speeds are low, so that, this way, good control performances can be achieved. Opposed to arm-vehicle coupled control strategies [1,14], this set of techniques offer simpler hardware/software implementation and require less knowledge of the system’s parameters.

In this paper, a dynamic manipulation strategy for an Intervention Autonomous Underwater Vehicle is proposed; the whole system is composed of an AUV equipped with a 7 DOFs robotic arm and a three fingered gripper, with 2 DOFs for each finger. Using multibody modelling techniques [15], mathematical and simulation models for the system are derived. Interaction with the fluid is taken into account [16], including hydrostatic and hydrodynamics effects due to added mass, drag and lift forces and buoyancy effects. Hydrodynamic coefficients have been evaluated by means of CFD analysis, and the coupling between the fluid equations and the multibody model of the I-AUV has been efficiently performed through the software Matlab [17]. A suitable contact model between the gripper’s fingers and an object to be manipulated is derived.

The control strategy for the whole system is proposed. A decoupled strategy has been chosen, in view of its simplicity and robustness; in addition, exploiting the hand kinematics, the arm has been further decoupled from the gripper, and the two are controlled independently. A grasp planning strategy is presented, consisting of a visual pose estimation algorithm along with a contact point individuation algorithm. The control architecture has been validated by means of realistic simulations on a suitable test case using the software Matlab. The proposed techniques, after an additional testing phase, will be used in opportune hardware tests in the framework of existing research projects such as the Italian project SUONO (*Safe Underwater Operations iN Oceans*) [18] and the European project ARROWS [19,20], coordinated by the MDM Lab of the University of Florence, to obtain preliminary experimental results.

I-AUV Kinematics and Dynamics

System schematization

In Figure 3, in a schematic way, the geometrical model of the system is shown: the system is made up of the AUV, with equivalent dimensions $B \times L \times H$. The vehicle possesses 6 degrees of freedom and is equipped with a manipulator arm, which is assumed to be, for the considered application, a serial robot with 7 DOFs. The latter, composed of 2 links of cylindrical form is linked to the vehicle body by a joint which allows rotation along two orthogonal axes (shoulder roll and pitch angles) just next to the frontal face, located down below and at the center, to minimize the rolling disturbances and keeping at the same time the center of the mass of the vehicle under the buoyancy one. The junction between the two links (arm and forearm) possesses the same degrees of freedom of the shoulder. A spherical wrist is attached

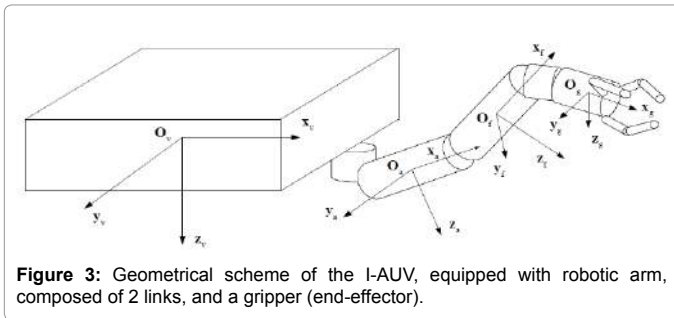


Figure 3: Geometrical scheme of the I-AUV, equipped with robotic arm, composed of 2 links, and a gripper (end-effector).

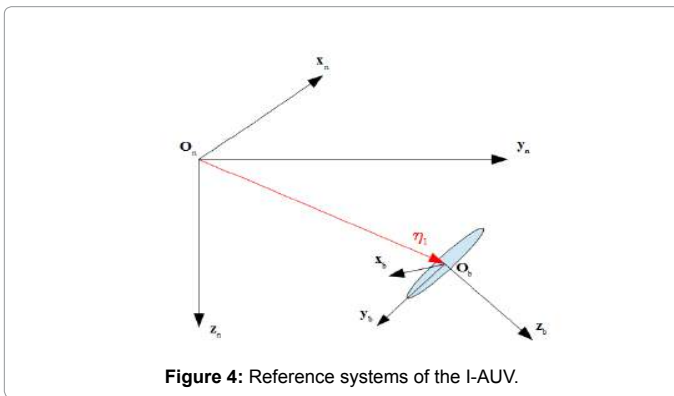


Figure 4: Reference systems of the I-AUV.

to the end of the forearm, for a total of seven degrees of freedom. On top of the wrist a 6-DOFs gripper is mounted: the latter has 3 fingers, each one composed of 2 phalanxes connected one to the other (and to the hand) by rotational joints. The center of the RPY spherical wrist is in the intersection of the rotation axes of the gripper as to the link of the forearm; besides, the reference frames are shown, linked to each rigid body and used to calculate the hydrodynamic terms.

The manipulator is similar to an anthropomorphic robot with first link collapsed in a single point, the intersection of the two rotation axes of the arm as the vehicle (shoulder) and the three bodies of the spherical wrist collapsed in the centre of the wrist itself, as well. To have a more realistic representation of the wrist, some finite lengths for these links and reasonable mass and inertia properties are assumed any way. The gripper, as it is composed of multiple mobile parts, represents a final branching of the kinematic chain, up to such a point of serial type, of the manipulator. It is assumed that all the DOFs of the system are controlled.

I-AUV model

The I-AUV model can be analysed dividing the study into two parts: the vehicle and the robotic arm. The geometrical and physical data regarding both the vehicle and the manipulator are taken from the technical literature considering the datasheets of commercial vehicles. As regards the gripper, in this research work, the connection with the arm has been simplified, assuming a rigid connection. The models are completely developed in Matlab Simulink® environment.

According to the SNAME notation [16], the kinematic model of the AUV is defined in terms of \$\eta\$ and \$v\$ vectors. \$\eta\$ represents the position (\$\eta_1\$) and the orientation (\$\eta_2\$) written in the fixed reference frame \$< n >\$; \$v\$ components are respectively the linear (\$v_1\$) and the angular (\$v_2\$) speeds described into the body reference frame \$< b >\$ (both the fixed and the body reference frames use the NED directions, Figure 4). The relations between \$\dot{\eta}\$ and \$v\$ can be written using the following expression:

$$\dot{\eta} = J_b^n(\eta_2)v, \quad (1)$$

$$\text{where } J_b^n(\eta_2) = \begin{bmatrix} J_b^n(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & T_b^n(\eta_2) \end{bmatrix}. \quad (2)$$

\$R_b^n(\eta_2)\$ is the rotation matrix between frame \$< n >\$ and frame \$< b >\$, and \$T_b^n(\eta_2)\$ is the transformation matrix between angular velocity and the time derivative of Euler angles (and its form depends on the particular choice of Euler angles).

Through Equation (1), the kinematic equations describing the AUV model are:

$$\begin{aligned} \dot{\eta} &= J_b^n v \Leftrightarrow v = (J_b^n)^{-1} \dot{\eta} \\ \ddot{\eta} &= J_b^n \dot{v} + \dot{J}_b^n v \Leftrightarrow \dot{v} = (J_b^n)^{-1} (\ddot{\eta} - \dot{J}_b^n (J_b^n)^{-1} \dot{\eta}). \end{aligned} \quad (3)$$

According to literature [16], the dynamic equation of the vehicle is defined as follows:

$$M_{RB} \dot{v} + C_{RB}(v)v = \tau H(v, v_c) + g(\eta) + \tau, \quad (4)$$

where \$M_{RB}\$ represents the mass matrix of the vehicle and \$C_{RB}(v)\$ is the Coriolis and centrifugal effect matrix. \$g(\eta)\$ and \$\tau\$ are respectively the contribution due to the gravity effects and the external forces, the latter composed of forces (\$\tau_1\$) and torques (\$\tau_2\$) applied to the vehicle as to the body reference frame \$< b >\$. These contributes are referred to the rigid body characteristics. Instead, the hydrodynamic effects \$\tau_H(v, v_c)\$ are partially decoupled from the dynamical equation in order to use the classical multibody techniques to solve the problem. In particular, the hydrodynamic effects and the buoyancy, are introduced into the model by means of generalized Lagrangian forces [21] applied to each body constituting the multibody system in order to increase the reality of the system. From the classic equation of motion for an underwater vehicle [16] and the absolute velocity \$v\$ written in the body.

Reference frame \$v = v_r + v_c\$ (where \$v_r\$ is the relative velocity and \$v_c\$ is the current velocity), the following expression for \$\tau_H(v, v_c)\$ can be extracted:

$$\begin{aligned} \tau_H &= -M_A \dot{v}_r + C_{RB}(v_r)v_c + C_{RB}(v_c)v_r + \\ &C_{RB}(v_c)v_c - C_A(v_r)v_r - D(v_r)v_r, \end{aligned} \quad (5)$$

where \$M_A\$ is the added mass matrix due to the fluid viscosity, \$C_A\$ is the Coriolis and centrifugal added effects, and \$D(v_r)\$ is the damping matrix.

The geometrical and physical characteristics of the vehicle are based on the literature and are defined in Table 1.

The I-AUV is provided of a robotic arm with 7 DOFs installed on the bow of the vehicle, in the middle of its lower part. For the kinematic model of the robotic arm (Figure 5), the joint coordinates \$q = [\theta_1 \theta_2 \dots \theta_7]^T\$ and the end-effector pose \$X = [x \ y \ z \ \phi \ \theta \ \psi]^T\$ are defined. According to the Denavit-Hartenberg approach, Table 2 collects the D-H parameters extracted for the arm. The main kinematic equations used to entirely describe the redundant manipulator are respectively, for the direct kinematics and for the differential kinematics:

Characteristic	Value
Degrees of Freedom	6
Length	0.8 m
Breadth	0.6 m
Height	0.4 m
Mass in air	100 kg

Table 1: I-AUV characteristics.

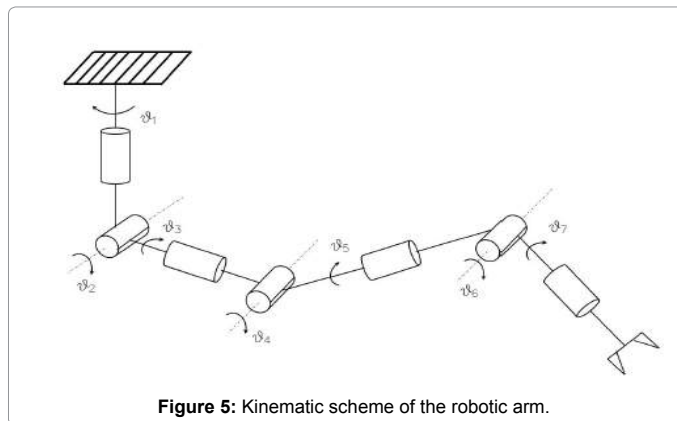


Figure 5: Kinematic scheme of the robotic arm.

Link	a_i [m]	α_i [rad]	D_i [m]	θ_i [rad]
1	0	$\pi/2$	-0.05	θ_1
2	0	$\pi/2$	0	$\theta_2 + \pi/2$
3	0	$-\pi/2$	0.15	θ_3
4	0	$\pi/2$	0	θ_4
5	0	$-\pi/2$	0.15	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	0.7	θ_7

Table 2: Denavit-Hartenberg parameters of the arm.

$$T_7^0 = T_7^0(q) \quad (6)$$

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J\dot{q} \quad (7)$$

where ${}^0\mathbb{R}^{4 \times 4}$ is the homogeneous transformation matrix between the base reference frame $\langle 0 \rangle$ fixed to the AUV and the end-effector reference frame $\langle 7 \rangle$ $q \in \mathbb{R}^{n \times 1}$ is the vector of the joint variables, \dot{p}_e is the time derivative of the end-effector position and \dot{q} is the time derivative of the joint coordinates q . According to the literature [22], the redundant DOFs are used in the Closed Loop Inverse Kinematics method to solve secondary tasks (i.e. the avoidance of the singularity or the minimization of the kinetic energy).

The dynamic model of the robotic arm is simulated through the multibody techniques described before, in which each rigid body is modelled as follows:

$$M_i^i \ddot{v}_i^i + C_i^i(v_i^i)v_i^i = \tau_H^i(v_i^i, v_{iC}^i) + g^i(\eta_i^i) + \tau_i^i \quad (8)$$

where M_i^i represents the mass matrix, $C_i^i(v_i^i)$ is the Coriolis and centrifugal effect matrix of the i th link. $g^i(\eta_i^i)$ and τ_i^i are respectively the contribution due to the gravity effects and the external forces applied to the link. The i th link characteristics define these contributes. As described before, the hydrodynamic effects $\tau_H^i(v_i^i, v_{iC}^i)$ are partially decoupled from the dynamical equation in order to use the classical multibody techniques to solve this problem; in Equation (9), all the matrices can be referred to the i th link characteristics:

$$\tau_H^i = -M_{Ai}^i v_r^i + C_i^i(v_r^i)v_c^i + C_i^i(v_c^i)v_r^i + C_i^i(v_c^i)v_c^i - C_{Ai}^i(v_r^i)v_r^i - D^i(v_r^i)v_r^i \quad (9)$$

where M_{Ai}^i is the added mass matrix due to the fluid viscosity, C_{Ai}^i is the Coriolis and centrifugal added effects, $D^i(v_r^i)$ is the damping matrix. Using this approach, it was possible to simulate the whole I-AUV model with the SimMechanics toolbox in Matlab Simulink

environment. The geometrical and physical characteristics of the robotic arm are taken from the technical literature; characteristic properties of the arm and forearm links are described in Table 3. The modelling of hydrodynamics and buoyancy effects $T_H(v, v_c)$ both of the vehicle and of the robotic arm is strongly necessary to reproduce in a proper way the I-AUV dynamical behaviour during a navigation phase or a manipulation task. In particular, these actions have been implemented in each body belonging to the I-AUV system (vehicle, links of the arm and gripper); the simulated effects are:

- Hydrostatic effects due to the added masses;
- Hydrodynamic effects due to the added masses;
- Drag and lift forces;
- Buoyancy effects.

For each body, these terms are described with respect to a reference frame system having its origin in the CG of the body, with axes parallel to the principal axes of inertia, vertical downhill z -axis, x -axis pointing towards the bow of the vehicle (when the robotic arm is stretched) and the y -axis accordingly in order to define a right-handed reference frame.

As regards the links of the robotic arm, three assumptions for the rigid bodies have been effectuated: they are of cylindrical shape, they are completely immersed and they are characterized by three levels of symmetry. Assuming that the arm does not move at high velocities, the added mass matrix M_A and the matrix of the centrifugal and Coriolis effects C_A are respectively described [16].

Another important aspect for the I-AUV modelling is the onboard sensor set. Here is reported, for sake of brevity, the list of the modeled sensors:

- GPS Adafruit module: working frequency 10 Hz, noise standard deviation 1.2 m;
- STS DTM Digital Pressure Transmitter (DPT): working frequency 10 Hz, noise standard deviation 0.3 m;
- Xsens MTI IMU: working frequency 100 Hz, noise standard deviation 0.1 deg;
- Teledyne Explorer Doppler Velocity Log (DVL): working frequency 10 Hz, noise standard deviation 0.1 cm/s [23]. All the sensor biases have been compensated through offline calibrations. The gyroscope bias is managed by a suitable filter [24].

Kinematic and dynamic model of the gripper

A 3D model of the gripper is shown in Figure 6. Each finger consists of two rotational joints connecting the hand to the first phalanx and the two phalanxes. A spherical tip is rigidly connected to the second phalanx. Using the Denavit-Hartenberg (DH) convention, the point where the first phalanx connects to the hand is the origin of reference frame 7 (for each finger), while the middle point of the finger is the origin of frame 8 and the end of the second phalanx is the origin of frame 9. Axis directions are chosen so that a positive value for DH parameter θ corresponds to finger's opening. The reference frame 7', visible in Figure 6, which is the frame attached to the end effector of the arm, is tied to the frame 7" (palm of the hand) by a constant transformation matrix; the same applies for the fingers' frames 7 with respect to the palm frame. As regards the gripper's dimensions, 0.025 m is the radius of the cylindrical hand, 0.1 m is its length, finger radius is 0.005 m, and the joints are modelled as cylindrical bodies with radius

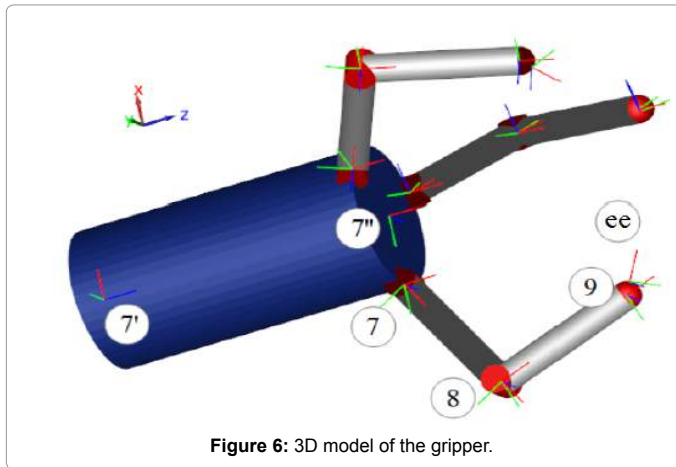


Figure 6: 3D model of the gripper.

Parameter	Value
Length	15 cm
Diameter	2.5 cm
Mass	2.315 kg
Inertia (body frame)	
I_{xx}	$7.23 \cdot 10^{-4} \text{ kgm}^2$
I_{yy}	0.0047 kgm^2
I_{zz}	0.0047 kgm^2

Table 3: Arm and forearm characteristics.

Parameter	Hand	Phalanx
Mass	1.5433 kg	0.0309 kg
I_{xx}	$4.82 \cdot 10^{-4} \text{ kgm}^2$	$3.87 \cdot 10^{-7} \text{ kgm}^2$
I_{yy}	0.0015 kgm^2	$6.62 \cdot 10^{-6} \text{ kgm}^2$
I_{zz}	0.0015 kgm^2	$6.62 \cdot 10^{-6} \text{ kgm}^2$

Table 4: Mass and inertia values of the gripper.

of 0.005 m and length of 0.01 m. Finally, a frame is attached to the end of each fingertips, rotated by 45° with respect to the axis of the second phalanx. These frames are the end-effector (ee) frames of each finger. Mass and inertia values of the hand and the fingers are shown in Table 4. The mass of the fingertip has been neglected.

Each finger is locally equivalent to a planar 2-DOFs manipulator, whose kinematics is shown in Figure 7; DH parameters (for one finger; they are the same for each) are reported in Table 5. The dynamic model of a finger in joint space is expressed by the well-known relation

$$B(q)\ddot{q} + n(q, \dot{q}) + J_f^T h_e = \tau_f \quad (10)$$

where B is the inertia matrix, n includes centrifugal, Coriolis and gravitational effects, J_f is the finger Jacobian matrix, $h_e \in \mathbb{R}^{6 \times 1}$ is the vector of forces/torques due to interaction with the environment and τ_f are the joint torques (i.e. the control inputs). As for the vehicle and the arm, multibody modelling techniques have been used for the gripper.

Contact model description

This section describes the algorithm that governs the contact between the spherical tips of the gripper's fingers and the object to be manipulated and between the object itself and the sea floor. The contact model has the following features [25,26]:

- Contact point: It is assumed that the contact point is a single,

Euclidean point;

- Hard finger contact: Tangential forces arise due to friction between the colliding surfaces;

- Generalizability: Even if the model has been created to govern the contact of the specific test case, its geometrical background is easily adaptable to different cases; thus, it is a complete three-dimensional model.

It is possible to divide the algorithm into two separate steps (Figure 8) at first, the distance between the contact surfaces is computed; if its value reaches zero, the algorithm determines the position in 3D space of the contact points and the slidings (i.e. the difference between the velocities of the contact points). Then, contact forces are computed, taking into account frictional forces. The following sections describe these steps in details, with particular reference to the considered test case.

Contact point computation: Let us consider the contact between a fingertip of the gripper and the lateral surface of the cylinder; by exploiting the geometrical background of the algorithm, one can easily transpose it to the case of contact between the base of the cylinder and the sea floor.

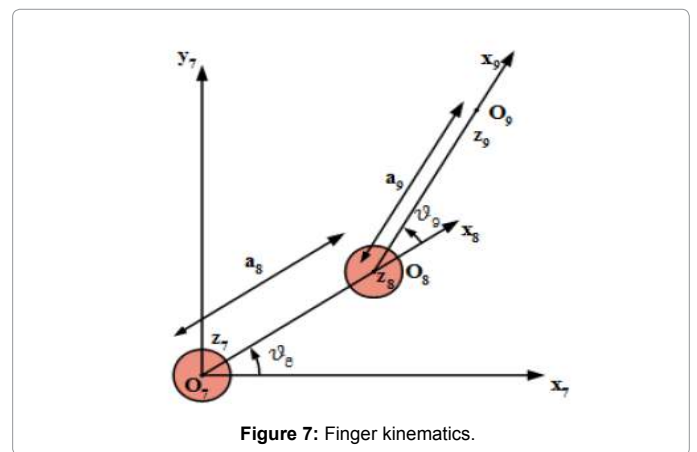


Figure 7: Finger kinematics.

Link	a_i [m]	α_i [rad]	d_i [m]	β_i [rad]
8	0.05	0	0	θ_8
9	0.05	0	0	θ_9

Table 5: DH parameters of the finger.

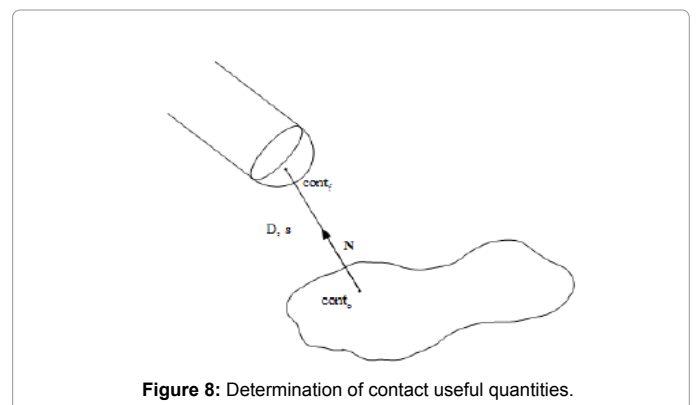


Figure 8: Determination of contact useful quantities.

Figure 9 shows the adopted notation; it is possible to recognize both the cylinder and the fingertip. O_9 represents the terminal point of the second phalanx of the finger, and its position can be determined using direct kinematics once the joint variables are known; in addition, the direction of the finger's major axis (x_9) is known. Let c_m and R_c denote the position and the orientation of the cylinder with respect to the inertial frame. Referring to Figure 9, the direction of the major axis of the cylinder is the third column of R_c . Contact points are, at first, computed in a local frame (the cylinder frame); they are then expressed in the inertial frame.

The first step consists of determining O_9 position in the local frame:

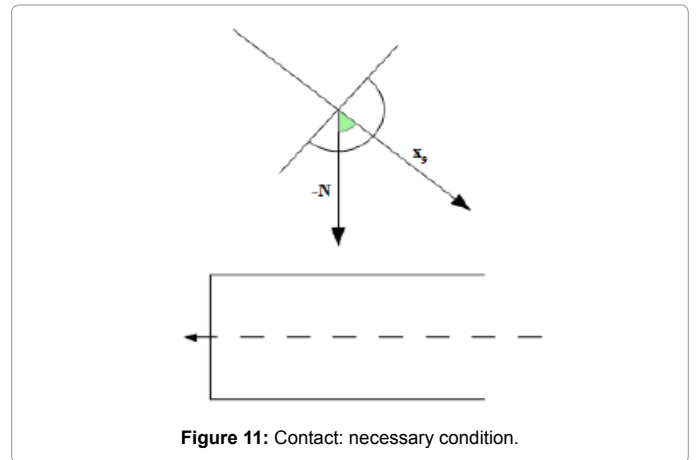
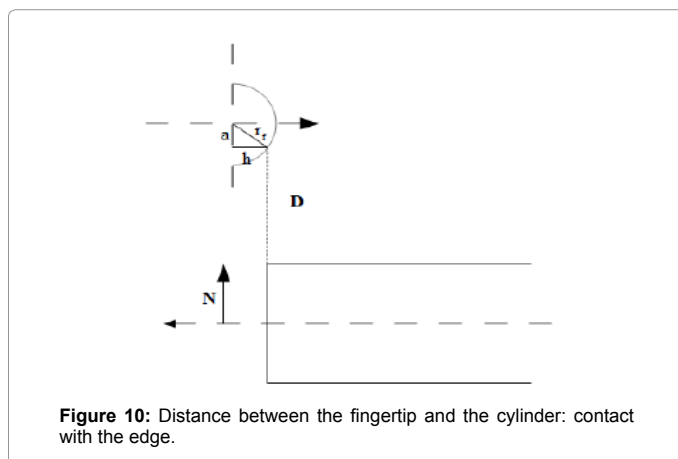
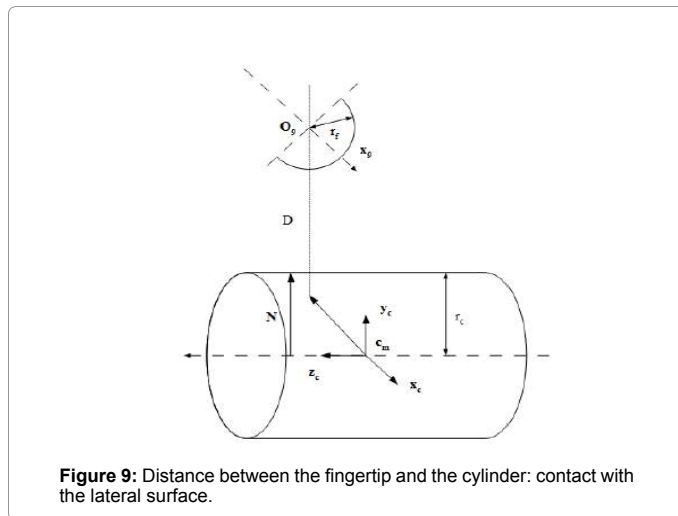
$$O_9^{loc} = R_c^T (O_9 - c_m). \quad (11)$$

If the absolute value of its z-coordinate does not exceed half the length of the cylinder, or this quantity plus the radius of the fingertip, then contact is possible. Depending on the case, this leads to an additional branching of the algorithm.

Contact with the lateral surface of the cylinder: Let us consider the first case, which is the one depicted in Figure 9. The distance between the two bodies is:

$$D = \|[O_9^{loc}(x) O_9^{loc}(y)]\| - r_c - r_f. \quad (12)$$

Contact with the edge of the cylinder: Particular care is needed if



$O_9^{loc}(z)$ lies beyond the bases of the cylinder, but within r_f (along z_c) from them. Figure 10 shows a lateral view of the described situation; x_9 and z_c are parallel, however this condition does not represent the general case. The value of a can be obtained as:

$$a = \sqrt{r_f^2 - h^2} \quad (13)$$

where h is the difference between $O_9^{loc}(z)$ and half the length of the cylinder. The distance is then given by:

$$D = \|[O_9^{loc}(x) O_9^{loc}(y)]\| - r_c - r_f + a. \quad (14)$$

The outward-pointing contact normal (expressed in the local frame) is, in both cases, the direction of vector $N^{loc} = [O_9^{loc}(x) O_9^{loc}(y) 0]^T$. The positions of the (coincident) contact points on the surface of the cylinder and on the fingertip, expressed in the cylinder frame, are given by:

$$cont_c^{loc} = [00 O_9^{loc}(z)]^T + r_c N^{loc} \quad (15)$$

$$cont_f^{loc} = cont_c^{loc} + D N^{loc}.$$

Then, the coordinates of the contact points and the normal vector can be expressed in the inertial frame:

$$cont_c = c_m + R_c cont_c^{loc} \quad (16)$$

$$cont_f = c_m + R_c cont_f^{loc}$$

$$N = R_c N^{loc}.$$

It is worth to mention another requirement to be fulfilled in order for contact occurrence. It is indeed necessary that the fingertip's surface and the cylinder can "see" each other. This amounts to verifying that the dot product between x_9 and $-N$ is not negative (i.e. the angle between the two vectors is less than 90 degrees), as visible in Figure 11.

Contact force computation: The knowledge of the coordinates of the contact points, the contact normal and the sliding allows the computation of the contact forces.

If the distance between the considered surfaces reaches zero, interaction occurs; then, the bodies exchange forces. Hard finger contact has been considered: three components of force are transmitted at contact; one is normal to the surface, while the others are frictional (tangential) forces.

Normal force follows a spring-damper model: let

$$s_N = S^T N \quad (17)$$

denote the absolute value of the component of the sliding vector along N ; then, normal contact force is given by

$$f_N = \pm (k_n D - c_n s_N) N \quad (18)$$

for $k_n > 0, c_n > 0$. The tangential force is composed of static and kinetic friction; the friction coefficient μ follows a law of the type shown in Figure 12:

$$\mu = \mu_k + (\mu_s - \mu_k) e^{-k \|s_t\|} \quad (19)$$

where $s_t = s - s_n N$ is the tangential sliding, $\mu_s > \mu_k$ are the static and kinetic friction coefficients and $k > 0$ is a tunable parameter. Tangential force is then:

$$f_t = -\mu \|f_N\| \frac{s_t}{\|s_t\|} \quad (20)$$

Finite slope can be assumed for small sliding values, to avoid chattering problems during simulations.

As a contact between two steel surfaces has been considered, the chosen contact parameters are reported in Table 6.

Camera modelling

During the grasp planning phase a pose estimation algorithm based on optical sensors (cameras) has been used; thus, it is necessary to give a mathematical model of such sensors. In this context, the basic pinhole model has been used: this model offers a good approximation of the behaviour of a real camera, maintaining a low level of complexity [27].

I-AUV Control Architecture

The purpose of control is to make sure that the system composed of the vehicle, the manipulator arm and the gripper can autonomously reach the object. The purpose of control is to make sure that the system composed of the vehicle, the manipulator arm and the gripper can autonomously reach the object to be manipulated and execute the planned task on it. Following a suitable reference trajectory, the control system is in charge of determining the correct inputs (i.e. the vehicle's thrusters forces and the arm and gripper actuating torques) that ensure

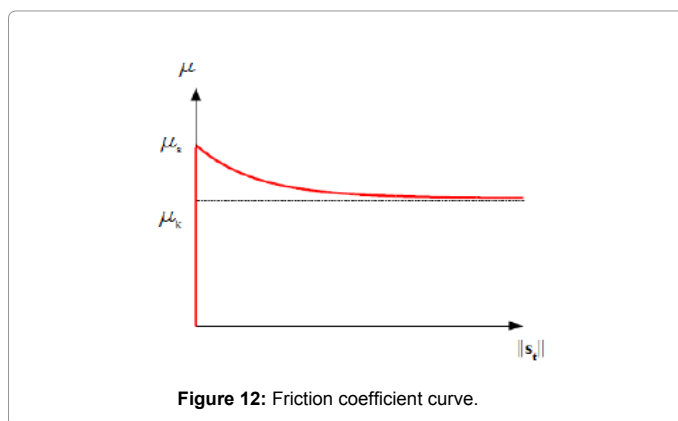


Figure 12: Friction coefficient curve.

Parameter	Value
Elastic coefficient	$kn=10000 \text{ kg/s}^2$
Damping coefficient	$cn=100 \text{ kg/s}$
Static friction coefficient	$\mu_s=0.5$
Kinetic friction coefficient	$\mu_k=0.3$
Friction curve parameter	$k=5$

Table 6: Contact model parameters.

the success of the specified manipulation task.

Before describing the control architecture in details, a brief discussion about sensors and actuators is due. To successfully perform an underwater autonomous manipulation task, many different sensors are required [28]. It is indeed of extreme importance the knowledge of an accurate estimation of the vehicle's and the object's pose; in addition, to obtain satisfactory performances for the control system, a precise measure of the joints variables of the arm and the gripper is required. To this purpose, several different sensors can be used: regarding vehicle's localization, common choices of sensors include IMU platforms, depth sensors, DVLs and acoustic modems; encoders and force sensors are used to ensure good performances of the manipulation systems; as regards the estimation of the object's pose, the underwater environment imposes strong limitations on sensor choice: in this context, optical sensors (cameras) have been considered. Sensors are, at the moment, assumed to be ideal; an accurate modelling will be dealt with once real sensors will be available.

Concerning actuation, it has been assumed that every DOF of the system is controlled; in particular, a six-DOF control is applied to the AUV. Actuating forces and torques and their variations have been limited by means of saturations and rate limiters, to simulate the presence of a real motor.

I-AUVs' control techniques can be divided into two categories [1]. A first set simultaneously controls the vehicle and the manipulation system [14], subjugating the first to the latter: in order to execute the task, the AUV moves to ease the arm and gripper operations. The second set of techniques makes use of a decoupled approach [13]: the arm manipulates the object while the vehicle tracks its own reference trajectory; the effects of one subsystem on the other are considered as disturbances.

Here, in view of its simplicity and robustness with respect to unknown parameters of the system, the second approach has been chosen. Moreover, exploiting the kinematics of the gripper, the control of the fingers has been further decoupled from the control of the arm: once the desired pose for the fingertips is known, it is indeed possible to univocally determine, from purely geometrical considerations, the desired pose of the end effector of the arm. This way, it is possible to control each component of the manipulation system (arm and fingers) independently.

A block diagram of the control architecture is shown in Figure 13. Filled lines represents physical interactions, while dashed lines stand

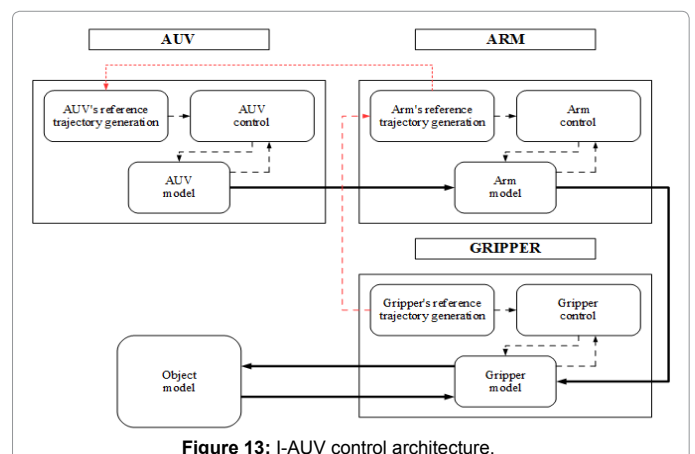


Figure 13: I-AUV control architecture.

for functional dependence. Every component of the system has their own controller and trajectory planning blocks; the latter, however, are not independent one from another: it is indeed required that the global trajectory permits the manipulation of the object. In particular, the arm reference is determined in view of the aforementioned geometrical decoupling algorithm. The dotted line connecting the vehicle's and the manipulator's trajectory planning blocks indicates that, even if the two references are virtually independent because of the adopted decoupled strategy, the AUV's trajectory must allow the arm to reach the object.

Figure 14 gives a better understanding of what has just been presented: the control techniques chosen for each component are shown. All the controllers are independent and they do not share data. The whole control system follows a "backward strategy": during the manipulation phase, the contact points on the object are computed by the grasp planning algorithms; then, a suitable continuous (smooth) trajectory that takes the fingertips on such points is generated. The positional control of the fingers is in charge of following this trajectory closely. The geometrical decoupling algorithm, which will be analysed in details in the following, allows the computation of the reference trajectory of the arm, which is kinematically controlled. Finally, an admissible reference trajectory for the AUV is generated, and a SISO PID control is applied to each DOF of the vehicle.

Gripper's reference trajectory generation and control

The contact points on the object surface constitute the desired (final) values of the reference trajectory; the desired orientation is chosen so that the approach axis of a fingertip frame points inwards the object (i.e. it is directed in the opposite direction of the contact normal).

The reference trajectory is generated as convex combination of initial and final pose [29]. The strategy can be seen as a particular case of point-to-point motion [30]. Let us consider position trajectory generation first: let x denote the generic (scalar) position variable; its desired trajectory is then chosen as

$$x(t) = \lambda(t)x_i + (1 - \lambda(t))x_d \quad (21)$$

where x_i and x_d represent the initial and the desired (final) value of x and $\lambda(t)$ is a parameter that continuously (and with continuous derivatives) varies from 1 to 0. This ensures a smooth transition between the initial and the desired values. The advantage of this method is that the continuity of the trajectory is maintained even if the desired value x_d varies with time: changes of the final value are automatically taken into account.

For orientation trajectory generation, it is possible to resort to an axis-angle parametrization and follow the same guidelines used for position trajectory generation: let R_i and R_d denote the initial and the desired orientation matrices, and let $\theta_p, r_p, \theta_d, r_d$ be their axis-angle parameters; it is possible to determine the desired orientation trajectory as a time-varying matrix.

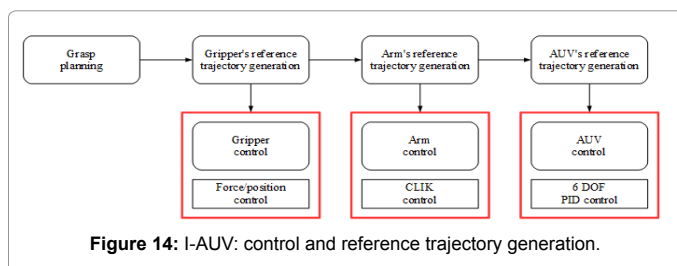


Figure 14: I-AUV: control and reference trajectory generation.

$$R(\theta(t), r(t)) = \begin{bmatrix} r_x^2(1-c_\theta) + c_\theta r_x r_y(1-c_\theta) - r_z s_\theta r_x r_z(1-c_\theta) + r_y s_\theta \\ r_x r_y(1-c_\theta) + r_x s_\theta r_y^2(1-c_\theta) + c_\theta r_y r_z(1-c_\theta) - r_x s_\theta \\ r_x r_z(1-c_\theta) - r_y s_\theta r_y r_z(1-c_\theta) + r_x s_\theta r_z^2(1-c_\theta) + c_\theta \end{bmatrix} \quad (22)$$

where c_θ and s_θ stand for $\cos(\theta)$ and $\sin(\theta)$ and

$$\begin{aligned} \theta(t) &= \lambda(t)\theta_i + (1 - \lambda(t))\theta_d \\ r(t) &= \lambda(t)r_i + (1 - \lambda(t))r_d \end{aligned} \quad (23)$$

Regarding control, each finger is independently controlled with a parallel force/position regulator: a PI action on the force error is inserted inside an inverse dynamics position control. Referring to (10), the control input is:

$$u_f = By + n + J^T h_e \quad (24)$$

where

$$y = J^{-1} M_{df}^{-1} (M_{df} \ddot{X}_d + K_{df} \dot{\tilde{X}} + K_{pf} \tilde{X} + X_f - M_{df} \dot{J} \dot{q}) \quad (25)$$

M_{df} , K_{df} and K_{pf} are gain matrices, \tilde{X} denote the position error (i.e. the difference between the desired position vector X_d , obtained from the grasp planning algorithm, and the effective value), and X_f is the object must be limited in order not to damage the object. Considering a PI controller, C_f is for given gain matrices K_{ff} , K_{if} . The

$$X_f = C_f \tilde{f} \quad (26)$$

Where C_f is the control action on the force error $\tilde{f} = f_d - f_e$. The force applied to the object must be limited in order not to damage the object. Considering a PI controller, C_f is

$$C_f = K_{ff} + K_{if} \int d\tau \quad (27)$$

for given gain matrices K_{ff} , K_{if} reader can refer to [31] for more details.

Arm's reference trajectory generation and control

During the navigation phase (i.e. while the I-AUV is patrolling, searching for the object to be manipulated) the arm is kept at rest position: without loss of generality, the arm is stretched, while the wrist leans forward to avoid wrist singularities. It is assumed that a camera is mounted on the palm of the gripper (eye-in-hand configuration). As soon as the object enters in the field of view of the camera, the POSIT algorithm [32] is executed to obtain an estimation of the pose of the object itself. Then, the arm's reference trajectory is changed so as to align the camera focal axis with the line connecting the estimation of the position of the center of mass of the object to the major axis of the gripper; this ensures that the object is kept inside the field of view of the camera all the time. During the manipulation phase, the arm's reference trajectory is obtained by means of a geometrical algorithm that determines the pose of the end effector of the arm from the knowledge of the fingertips pose. Hence, this algorithm can be used for the arm's reference trajectory generation: once the desired pose of the fingers (obtained from the grasp planning algorithms) is given, one can easily obtain the reference trajectory (position and orientation) for the end effector of the arm. The solution is unique, and the solving process is composed of just a few steps; this results in a very lightweight and efficient algorithm, hereafter explained in details.

Maintaining the DH convention, once the desired pose for the fingertips is known, the desired position of the origin of the frames 8 and 9 for each finger can be found: the position of the origin of frame 8 is obtained "going back" along the desired approach direction

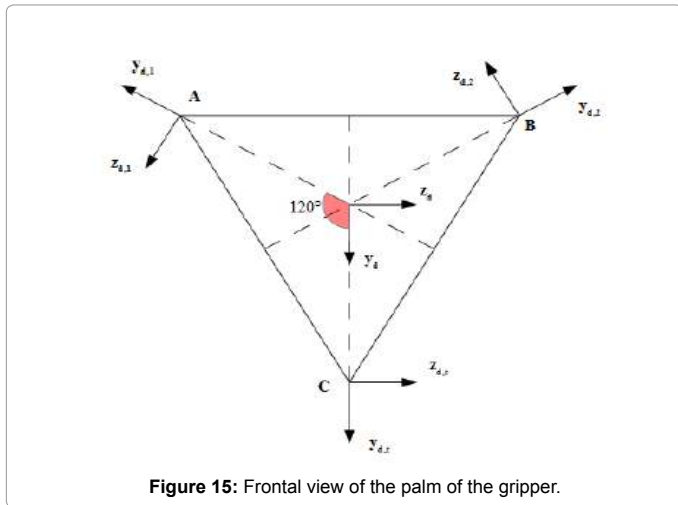


Figure 15: Frontal view of the palm of the gripper.

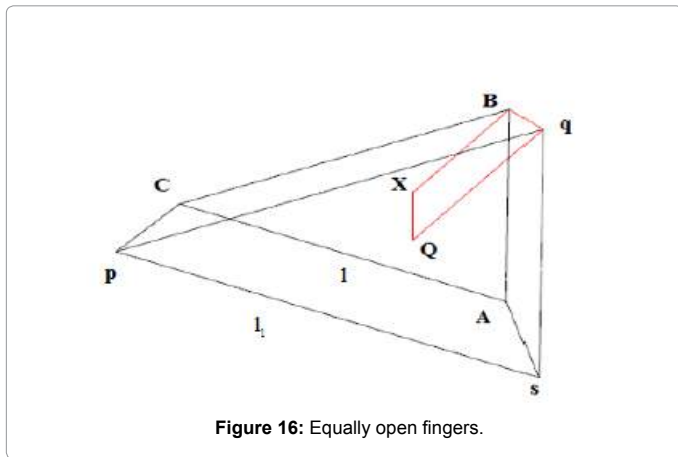


Figure 16: Equally open fingers.

of frame 9, whose pose is tied to the one of the fingertip frames by a rigid (constant) transform. Note that only the position of the origin of frame 8 is known. The vertices A, B and C of the triangle shown in Figure 15 represent the points on the palm of the gripper where the fingers connect to the hand (which are the origins of the frames 7); the proposed view is frontal.

Axes $z_{d,i}$ for $i = t, 1, 2$ are the (desired) joint axis directions, whose positive direction is chosen so that a positive value for the DH parameter θ corresponds to finger's opening. Axes $x_{d,i}$ are pointing outwards the palm, and axes $y_{d,i}$ are chosen so as to form a right-handed coordinate system. x_d, y_d and z_d denote the palm reference frame. Its orientation is chosen to be coincidental with the thumb frame; however, due to symmetry, this choice is not unique.

The first step of the algorithm consists in determining the desired orientation of the palm: z_d is already known (it has the same direction of $z_{d,t}$); x_d is given by the cross product of $z_{d,t}$ and $z_{d,2}$; finally, y_d is given by the cross product of z_d and x_d .

Let us now assume that the joint angles of frame 7 (θ_8) are the same for all the fingers, as shown in Figure 16, where p, q and s represent the origins of frames 8 and X is the centroid of the triangle \overline{ABC} . Note that \overline{ABC} and \overline{sqp} are equilateral and lie on parallel planes. Hence, the trapezoid whose vertices are B, q, Q and X,

where Q is the centroid of \overline{sqp} , is rectangular. The length of the base of such trapezoid are known (they are the segments which connect a vertex of an equilateral triangle to its centroid); the value $\|B - q\|$ is known too (length of a finger).

These observations prove to be useful in the general case (Figure 17): regardless of the value of the joint variable, the component of $(q - p)$ along z_d equals half the length of the side of the equilateral triangle obtained if all the fingers were opened at the same angle of the considered one. The centroid (Q) of such triangle can be found moving $a = b/\cos(30^\circ)$ along $-y_{d,2}$, starting from q.

The vector that connects this point to X is parallel to x_d ; its absolute value can be found as:

$$h = \sqrt{r^2 - g^2} \tag{28}$$

Where g is the absolute value of the difference between a and the radius of the hand. Relation (28) is valid both for the open hand ($a > r_{hand}$) and for the closed hand ($a < r_{hand}$). Finally, X can be found as

$$X = Q - hX_d \tag{29}$$

The palm frame, whose pose is given by X, $R_p = [x_d \ y_d \ z_d]$, is tied to the frame attached to the end effector of the arm by a rigid transform; its pose is then determined resorting to the inverse transform. The workflow of the algorithm is shown in Figure 18.

To overcome the problem of obtaining exact values for the hydrodynamic coefficients of the arm (whereof only an estimation is available), a kinematic control has been preferred to a dynamic control strategy. Hence, a Closed Loop Inverse Kinematic Control (CLIK) has been chosen [30,33] exploiting the arm's redundancy to keep the manipulator far from singularities. The following inversion algorithm has been used:

$$\dot{q} = J^*(q) \begin{bmatrix} \dot{P}_d + K_p e_p \\ \omega_d + K_o e_o \end{bmatrix} + N_f \dot{q}_0 \tag{30}$$

where \dot{P}_d and ω_d are the desired linear and angular velocity of the end

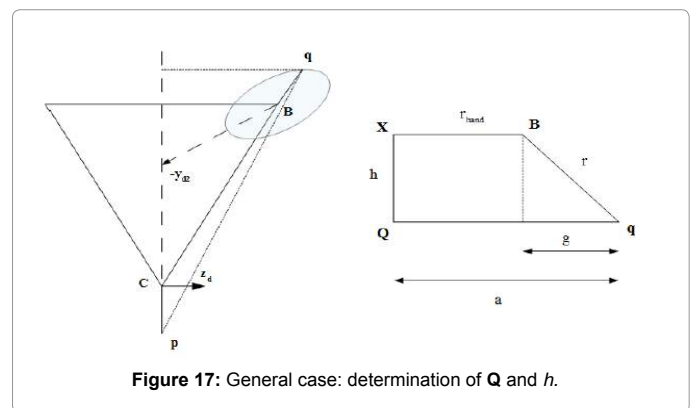


Figure 17: General case: determination of Q and h.

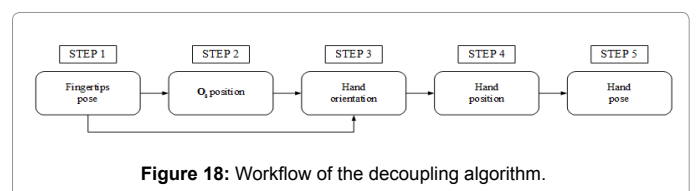


Figure 18: Workflow of the decoupling algorithm.

effector of the arm, e_p and e_o denote position and orientation errors, and K_p and K_o are definite positive gain matrices which ensure convergence of the pose error. J^* is the damped least square inverse of the manipulator Jacobian, given by:

$$J^* = J^T (JJ^T + k^2 I)^{-1} \quad (31)$$

where $k^2 I$ is the damping factor that renders the inversion better conditioned from a numerical point of view. \dot{q}_0 are additional joint velocities used to accomplish the secondary task of staying far from singularities, being N_J a projector in the null space of J . Orientation error is expressed by means of unit quaternions [30]: this is the most efficient choice in terms of computational load. Note that the desired values for linear and angular velocity of the end effector can be obtained from the desired position and orientation given by the geometrical decoupling algorithm, while effective pose can be computed using the direct kinematics of the arm.

Joint velocities \dot{q} , once integrated, constitute the reference values for seven PID controllers, each one used to control a single DOF of the arm. Joint velocities \dot{q} , once integrated, constitute the reference values for seven PID controllers, each one used to control a single DOF of the arm. The 7×1 control vector (i.e. arm joints torques) is then:

$$u_{arm} = K_{pa} \dot{q}(t) + K_{da} \ddot{q}(t) + K_{ia} \int_0^t \dot{q}(\tau) d\tau \quad (32)$$

being \dot{q} the difference between the q computed by the inversion algorithm and actual values, and K_{pa} , K_{da} , K_{ia} are suitable gain matrices.

AUV's reference trajectory generation and control

As the definition of dynamic manipulation implies, the AUV never stops during the execution of the task; furthermore, the vehicle must constantly keep the object inside the arm and gripper's workspace in order for manipulation to take place. It is indeed the vehicle's motion which determines the time when the manipulation phase starts: in fact, this step begins when a threshold value for the distance between the gripper and the object is obtained. This amounts to reaching the surface of a hemisphere centered at the object's center of mass and whose radius is the given threshold value. The AUV is controlled by means of a decoupled PID strategy: six PID controllers have been used, one for each degree of freedom of the vehicle [16]. The control law is:

$$u_{AUV}^{th} = H^+ u_{AUV} \quad (33)$$

where

$$u_{AUV} = J_b^n \tau_{PID} + g(\eta) \quad (34)$$

being τ_{PID} a 6×1 vector of PID action on the pose error e_n (i.e. the difference between the desired value η_d given by the vehicle's reference trajectory generator, and η), defined as follows:

$$\tau_{PID} = K_{pv} e_n(t) + K_{dv} \dot{e}_n + K_{iv} \int_0^t e_n(\tau) d(\tau) \quad (35)$$

for appropriate gain matrices K_{pv} , K_{dv} and K_{iv} . J_b^n is the AUV Jacobian matrix, $g(\eta)$ a term of gravity compensation, and H^+ is a generalized pseudo-inverse of the propeller matrix H [1], which maps the vector of thruster forces into the vector of forces/torques acting on the vehicle:

$$\tau = HS. \quad (36)$$

In the considered case, $S = [S_1 S_2 S_3 S_4 S_5 S_6]^T$; hence, the propeller matrix is square and it has the form:

$$H = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \\ P_{m1} \times V_1 & P_{m2} \times V_2 & P_{m3} \times V_3 & P_{m4} \times V_4 & P_{m5} \times V_5 & P_{m6} \times V_6 \end{bmatrix} \quad (37)$$

Where P_{mi} for $i = 1$ to 6 is the propeller position vector in body reference frame and v_i for $i = 1$ to 6 is the unitary vector of the thrust directions.

Grasp Planning

In the context of dynamic manipulation, the arm must manipulate an object while the AUV keeps moving; hence, it is of fundamental importance the knowledge of the exact pose of the object to be manipulated in 3D space, and which points on its surface may be "good" contact points. From a practical point of view, these operations require adequate sensors, either optical (cameras) or acoustic (e.g. multibeam) that give in real time (or at a sufficiently high frequency) an estimation of the shape and of the pose of the object and an algorithm whose purpose is the computation of contact points that can guarantee the success of the manipulation.

In this context the shape of the object is supposed to be known, and an algorithm that computes an estimation of the pose of the object in 3D space using cameras has been considered. Subsequently, the performances of a grasp planning algorithm have been investigated: this algorithm computes the location of good contact points by minimizing (or maximizing) suitable grasp quality indexes. Hence, starting from a single image of the object obtained by a camera, the position of contact points which guarantee the quality of the grasp is computed.

This section is organized as follows: first, the pose estimation algorithm is introduced. Next, the aforementioned grasp quality indexes are considered and the contact points calculation algorithm is presented.

Pose estimation algorithm

POSIT (*Pose from Orthography and Scaling with Iterations*) algorithm gives the pose of the object to be manipulated by means of images acquired using cameras; indeed, a single image obtained from a single camera is used [32]. The shape of the object is supposed to be known; this way, the pose of adequate points on its surface in a reference frame attached to the object (object points) can be computed. In addition, the perfect correspondence between these points and the points seen on the camera screen (image points) is required.

The algorithm is composed of two steps: the first step (POS) computes an approximate pose from the coordinates of the image points (at least four non-coplanar points are required) by using linear algebra; this amounts to assuming that the image points are given by the SOPs (*Scaled Orthographic Projections*) of the object points. SOPs are indeed an approximation of the true perspective projections (PP), obtained assuming that the distance between the object points along the focal axis of the camera can be neglected. The second step of the algorithm (IT), corrects the first pose estimation by recursively applying POS algorithm to the points computed at the previous iteration.

Software implementation of the POSIT algorithm is rather easy, and it usually converges to accurate pose measurements in a few iterations; these features make it suitable for use in difficult environments. The block scheme in Figure 19 shows the workflow of the algorithm.

Contact points individuation

For manipulation to be successful an intelligent grasp planning strategy is needed; a necessary requirement is a choice of the contact

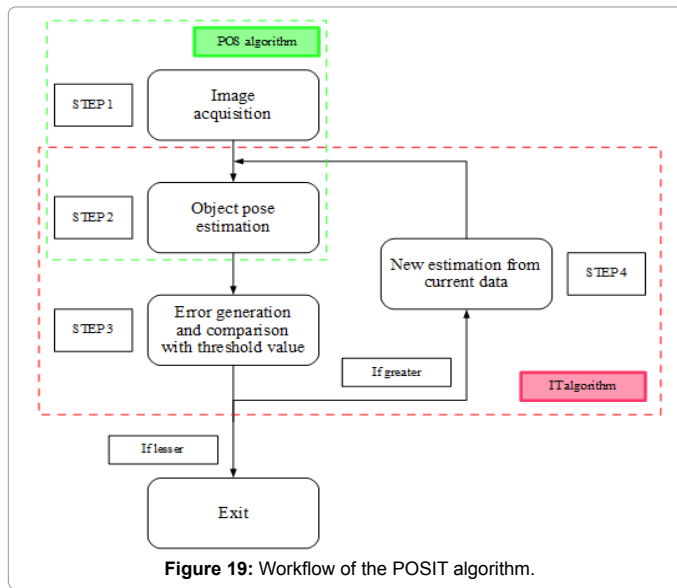


Figure 19: Workflow of the POSIT algorithm.

points that ensure the quality of the grasp. Such quality can be evaluated by means of numerous desirable properties that the grasp may possess. Among these properties, the most important are *form closure* and *force closure*. The first indicates the capacity of the hand to completely immobilize the object, while the latter is the ability to resist any external load applied to the object by arbitrarily “squeezing” the object itself. A lot of sources regarding grasp analysis can be found in literature, and alternative definitions of form/force closure can be found too, e.g. [22,34,35]. In this context, the considered grasp planning algorithm aims at optimizing the contact points’ choice with respect to adequate quality indexes in order to maximize the ability to resist external disturbances. The minimization (or maximization) of these indexes is bound to achieving the afore-mentioned force closure property.

This section analyses the considered quality indexes and presents the algorithm [36-42].

Grasp quality indexes

Shape of the grasp polygon: For planar grasps, the term *grasp polygon* indicates the polygon whose vertices are the contact points on the surface of the object; the property of the grasp to resist external wrenches applied is improved if contact points are distributed in an uniform way. Hence, the associated grasp index is a function of the difference between the internal angles of the grasp polygon and the angles of the regular polygon with the same number of vertices:

$$Q_1 = \frac{1}{\theta_{\max}} \sum_{i=1}^m |\theta_i - \bar{\theta}| \quad (38)$$

where m is the number of contact points, $|\theta_i - \bar{\theta}|$ is the modulus of the difference between the internal angle at vertex i and the internal angle of the associated regular polygon, and θ_{\max} is the sum of the internal angles when the polygon has the most ill conditioned shape (it degenerates into a line and the internal angles are either 0 or π radians). Index Q_1 equals zero when the grasp polygon is regular. In particular, in the case of a three-fingered contacts it coincides with an equilateral triangle.

Grasp polygon area: For planar grasps, the larger the area of the grasp polygon, the larger the external load that the same contact forces

can resist. The second quality index is expressed as follows:

where G_p indicates the grasp polygon. For 3D cases, one may consider the volume of the convex hull of the contact points.

$$Q_2 = Area(G_p) \quad (39)$$

Where (G_p) indicates the grasp polygon. For 3D cases, one may consider the volume of the convex hull of the contact points.

Distance between the centroid of the grasp polygon and the centre of mass of the object: Gravity and inertial effects on the contact forces are minimized if the centre of mass of the object belongs to the plane where the grasp polygon lies (the grasp plane). Thus, the third quality index is

$$Q_3 = \|c_m - c_m^p\| \quad (40)$$

where c_m is the position vector of the centre of mass and c_m^p is its projection on the grasp plane.

In addition to the previous indexes, the computation of good contact points is simplified if they are coplanar; in the considered case, since there are three contact points (and they are not aligned), they necessarily belong to the unique plane which contains them all. Furthermore, it is important to notice that the presented indexes are not the only available options to evaluate grasp quality: it is possible to include other indexes that account for the hand kinematics or other relevant factors of the considered case, such as keeping hand joints far from mechanical stops or avoiding to grasp the object in some undesirable areas.

Grasp planning algorithm

Let the shape and the position of the centre of mass object be known; in particular, let its surface be discretized with a sufficient number of points. The working principle of the algorithm is the following: once initial candidates for the contact points have been chosen, at each iteration a virtual force vector f_v^i is computed (for each contact point) that “pushes” those points toward a better grasping configuration. The tangential component of f_v^i is considered, and new candidates are chosen in its direction; the search ends when the norm of the virtual force falls below a threshold value, or after the maximum number of iterations allowed. A block scheme of the complete workflow is depicted in Figure 20.

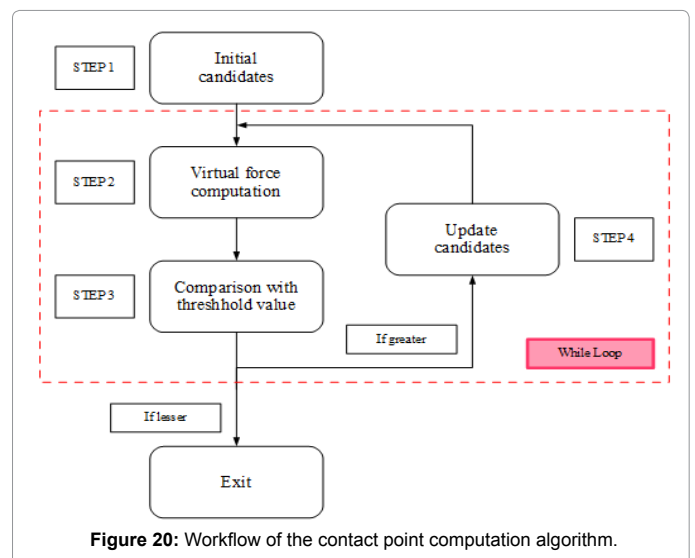


Figure 20: Workflow of the contact point computation algorithm.

Before analysing in detail the terms that compose the aforementioned virtual force, it is important to notice that, generally speaking, during the execution of the algorithm the candidates do not lie on the same plane; hence, referring to Figure 21, for the computation of f_v^i their projections (indicated by the apex P) on the plane Π which minimize the distance from all the contact points are considered. In this case, since there are only three contact points, Π is the unique plane which contains them all; points coincide with their projections, and apex P has been omitted for the ease of reading.

Virtual force f_v^i is composed of the following terms:

$$f_v^i = f_{cm} + f_a^i + f_e^i + f_\beta^i \quad (41)$$

Where the first three contributes are tied to the previously introduced quality indexes. In particular, the effect of each term is the following:

$f_{cm} = k_{cm}(c_m - c_m^p)$, $k_{cm} > 0$ is the same for all the contact points; the term tries to shorten the distance between the centre of mass of the object and plane Π ; $f_a^i = k_a \frac{p_i - c_m^p}{\|p_i - c_m^p\|}$, $k_a > 0$ is the virtual force in charge of enlarging the grasp polygon area by increasing the distance between the i -th contact point and c_m^p ;

$f_e^i = k_e(\theta_i - 2\pi/m)t_i$, $k_e > 0$ aims at producing an equilateral grasp configuration; θ_i is the angle between $(p_i - c_m^p)$ and $(p_j - c_m^p)$ with $j = i + 1$ for $i = 1, \dots, m$ and $j = 1$ for $i = m$, while t_i is the vector, lying on Π , normal to $(c_m^p - p_i)$ and pointing toward p_j .

To better understand the effect of f_e^i , refer to Figure 22, where a three

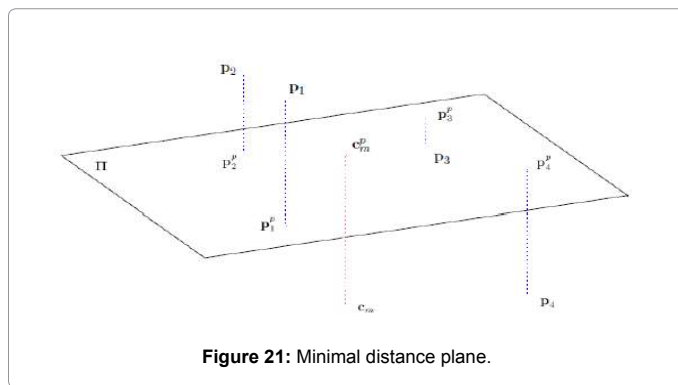


Figure 21: Minimal distance plane.

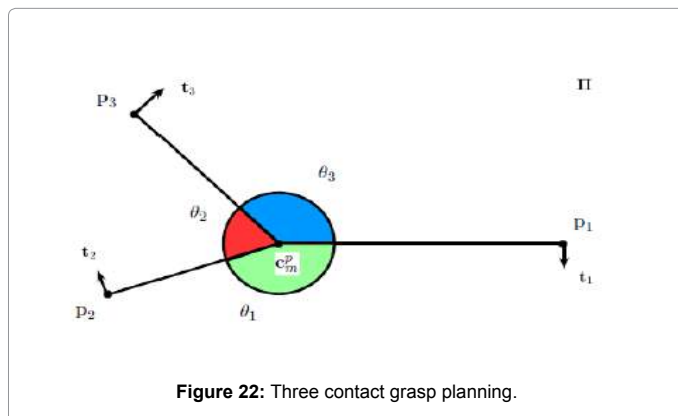


Figure 22: Three contact grasp planning.

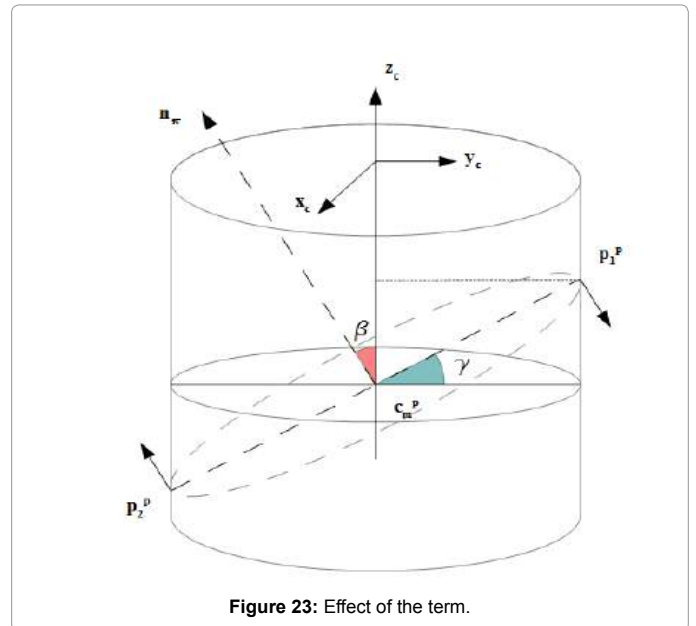


Figure 23: Effect of the term.

contact points grasp configuration is depicted. Figure 22 shows a top view of the plane Π . In case of equilateral grasp configuration, internal angles are all of the same amplitude (60°); hence, $\theta_1 = \theta_2 = \theta_3 = 120^\circ$. For the i -th contact point p_i the (signed) difference between θ_i and the reference value $2\pi/m = 120^\circ$ is computed; if, for example, the result is positive the effect of f_e^i is such to move p_i on Π in order to reduce the amplitude of θ_i toward the reference value. If, on the contrary, the difference is negative, the force will act along $-t_i$ as to increase the amplitude of θ_i .

f_β^i : this term is not directly bound to any of the quality indexes (38), (39) or (40); however, its effect is to produce a better grasp configuration. For instance (Figure 23), let us consider the case of grasping a cylindrical object by means of two contacts. Let $x_c y_c z_c$ be a suitable reference frame attached to the object, such that its axes are aligned with the symmetry axes of the object itself. f_β^i is in charge of aligning the normal n_n to the grasp plane with one of the axes of the object reference frame, in order to exploit existing symmetries in the choice of the contact points. Its expression is as follows:

where β is the angle between n_n and z_c ; if the positive direction of n_n is chosen as the one that gives the maximum dot product between n_n and z_c (so that β is always $< 90^\circ$), then the direction of f_β^i is chosen as follows: if the projection of p_i^p on z_c is greater than the the projection of c_m^p on the same axis, then f_β^i lies on $-n_n$. For the case depicted in Figure 23, the term f_β^i reduces the amplitude of the angle γ . k_{cm} , k_a , k_e and k_β constitute the relative weight associated which each term.

Let n_i be the normal to the object surface at point p_i ; once f_v^i has been computed, its tangential component is given by:

$$f_{v,t}^i = f_v^i - (f_v^i \cdot n_i)n_i \quad (42)$$

If its magnitude is lower than a threshold value α_j , the i -th contact point is fixed at p_i ; otherwise a new candidate p_i^0 is selected as the closest point to p_i in the direction of $f_{v,t}^i$. As a consequence, a careful tuning of the threshold value α_j is required: indeed, it represents the trade-off between optimality of the solution and execution time.

The algorithm has been implemented for a spherical and a

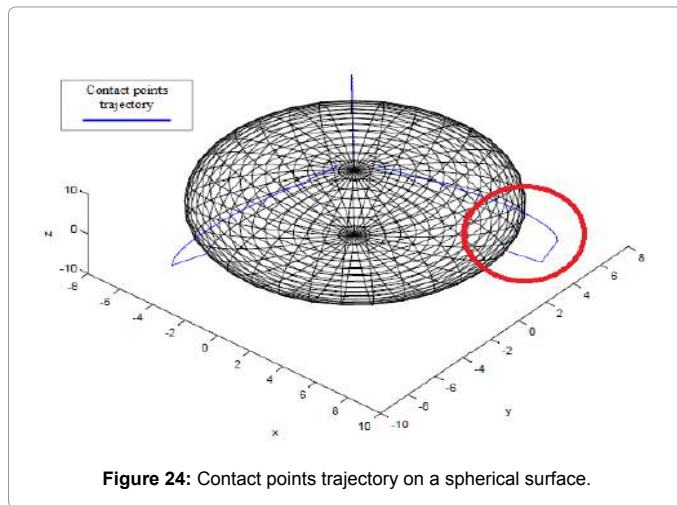


Figure 24: Contact points trajectory on a spherical surface.

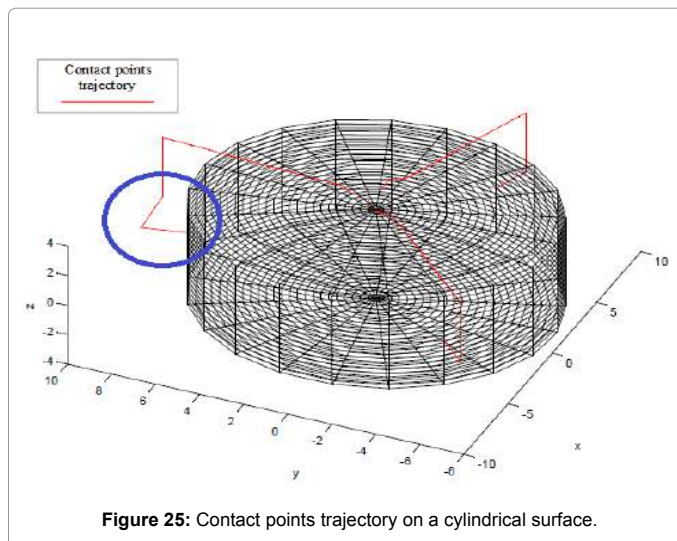


Figure 25: Contact points trajectory on a cylindrical surface.

cylindrical surface using Matlab[®]. A three-fingered grasp has been considered in both cases. Initial candidates have been chosen as not to form an equilateral grasp configuration; however, as it is shown in Figure 24 for the sphere and in Figure 25 for the cylinder, the algorithm moves the candidates horizontally in order to reach an equilateral configuration. For all the contact points the magnitude of the virtual force $f_{v,i}^f$ at the end of the execution is zero.

During the execution the contact points are kept at a fixed distance from the surface of the object, as a safety measure aimed at avoiding undesirable contacts when the output of the algorithm is used as the reference value for the positional controller of the gripper's fingers. This distance is then reduced to zero at the last iteration, once the final contact points have been chosen.

In the context of manipulation, if the shape of the object is known, the algorithm can be executed a priori: the contact points are computed in a local reference frame, and then rotated and translated once the pose of the object is obtained in a suitable reference frame. However, in the general case of unknown object, the presented algorithm is suitable for parallel use together with a shape reconstruction algorithm: the candidate contact points are computed at each iteration, assuming that the real shape of the object is the one obtained from the shape

reconstruction algorithm. Hence, the number of points to be used to discretize the surface of the object must be carefully chosen: the greater the number of points, the greater the chance of obtaining an optimal solution. This, however, affects the execution time of the algorithm. In the presented examples 1091 points have been used for the sphere and 1325 for the cylinder; generally speaking, a thousand points offer a good trade off between accuracy of the solution and computational speed.

Finally, it is advised to use, together with the proposed ones, an algorithm that computes the optimal contact force for each finger on the basis of the object properties and the required task.

Simulation Results

Task description and technical details

The behaviour of the whole I-AUV system has been simulated during the execution of a dynamic manipulation task, whose scheme is presented in Figure 26. The task consists of grasping a cylindrical object lying on the sea floor, and it is composed of the following steps:

- The I-AUV, starting from rest, accelerates for 5 seconds until steady-state speed is reached;
- As soon as the cylinder enters the field of view of a camera mounted on the palm of the gripper (eye-in-hand configuration), the reference trajectory is changed so as to align the camera focal axis (which corresponds to the gripper major axis) with the line connecting the palm of the gripper to the position of the center of mass of the object (estimated by the POSIT algorithm); this way, the cylinder is kept inside the field of view of the camera all the time;
- When a threshold distance between the gripper and the object is reached, the next step takes place: the hand is positioned over the object, aligned with its major axis, maintaining a safety distance in order to avoid undesired collisions;
- The gap is then closed and the fingers grasp the cylinder. The applied force is limited not to damage the object;
- The cylinder is lifted and carried as the arm reaches a final "rest" configuration (i.e. the cylinder carried vertically under the bow of the vehicle).

It is worth to note that the AUV never stops nor decelerates during the execution of the task, which is a fundamental requirement for a dynamic manipulation task. For the ease of reading, reported data (unless stated otherwise) are expressed in an inertial reference frame whose x axis is aligned with the forward direction of the vehicle, z axis is directed toward the sea surface, and y axis forms a right-handed coordinate system. Without loss of generality, the origin of this frame is chosen coincident with the initial position of the center of mass of the AUV. The object lies on the sea floor at about -0.6 m (initial depth). Due to its dimensions, hydrodynamic effects have been neglected (gravity has been used instead).

The described task has been simulated using Matlab Simulink[®] software. Solver parameters are visible in the Table 7. A fixed step

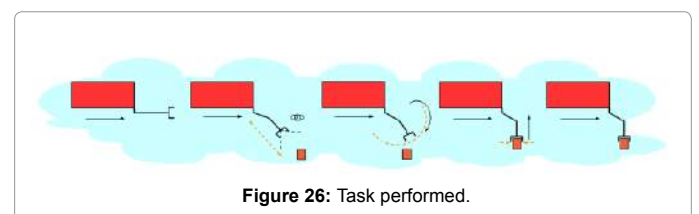


Figure 26: Task performed.

Parameter	Value
Integrator	ode5 (Dormand-Price)
Step	Fixed step
Step size	10^{-4} s

Table 7: Solver parameters.

Test case parameters			Cylinder parameters	
AUV speed	Acceleration time	Acceleration profile	Radius	2.5 cm
0.1 m/s	5 s	Trapezoidal	Length	6 cm
0.2 m/s	5 s	Trapezoidal	Density	7860 kg/m ³
0.25 m/s	5 s	Trapezoidal	Mass	926 g

Table 8: Simulation parameters.

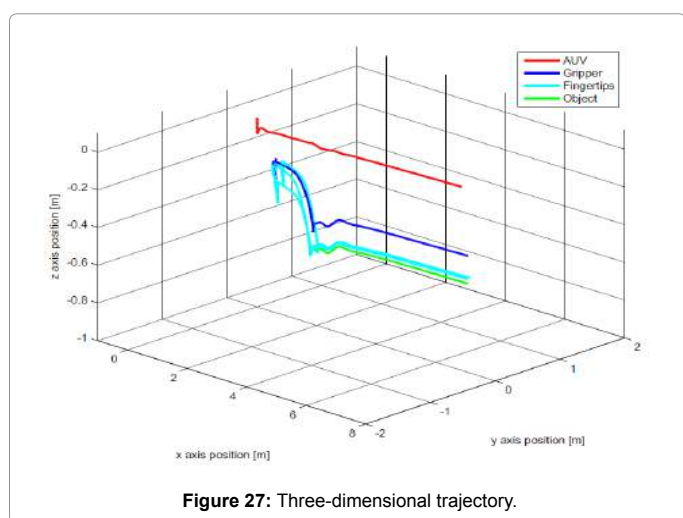


Figure 27: Three-dimensional trajectory.

solver has been chosen to increase affinity with real hardware.

Results

Three simulations have been carried out, at different vehicle's speeds: 0.1 m/s, 0.2 m/s and 0.25 m/s; this makes possible the analysis of the effect of increasing speed on the performances of the control system. Table 8 summarizes the main parameters of the simulations and the cylinder properties. Figure 27 reports the three dimensional trajectory of the vehicle, the gripper, the fingertips and the object obtained during the third simulation.

Figure 28 shows the behaviour of the AUV during the three simulations: the error along the direction of forward motion (the x axis) is very small, reaching a maximum values of 2 cm for the third simulation. Its value increases with the speed of the vehicle; this is because the AUV always accelerates for 5 s, thus higher steady-state speed equals higher acceleration values, which lower the performance of the PID controller. Y-motion, roll and yaw angular errors are neglectable. Initial errors on the z-axis motion and on the pitch angle, which are not affected by speed, are due to the total buoyancy of the system: while the vehicle has positive buoyancy (1%), the arm and the gripper tend to sink the I-AUV; in addition, since the manipulation system is mounted centrally on the front side of the AUV (to minimize the disturbance on the roll angle), it has the effect of leaning the vehicle forward. However, these errors are kept small and rejected in time.

Arm position errors, visible in Figure 29, are expressed in a local coordinate system whose origin coincides with the shoulder of the manipulator. Aside from the initial error (due to gravity), Figure 29 shows that the error is kept small during manipulation. As for the vehicle, it increases with the speed the I-AUV maintains while executing the task. Vertical lines in Figure 29 denote the initial time of each step of the manipulation operation: positioning over the object, grasping, lifting and finally moving the arm to the rest position while carrying the cylinder.

Fingertips position errors are very similar to the arms ones; for instance, Figure 30 shows the position error obtained for one of the fingers during the three simulations.

Figure 31 shows the position of the center of mass of the cylinder on each axis. Plots have been divided according to x, y and z coordinates. It is clearly visible the time when the object is grabbed and then lifted (motion on the x and z axis, respectively), while transversal motion (on

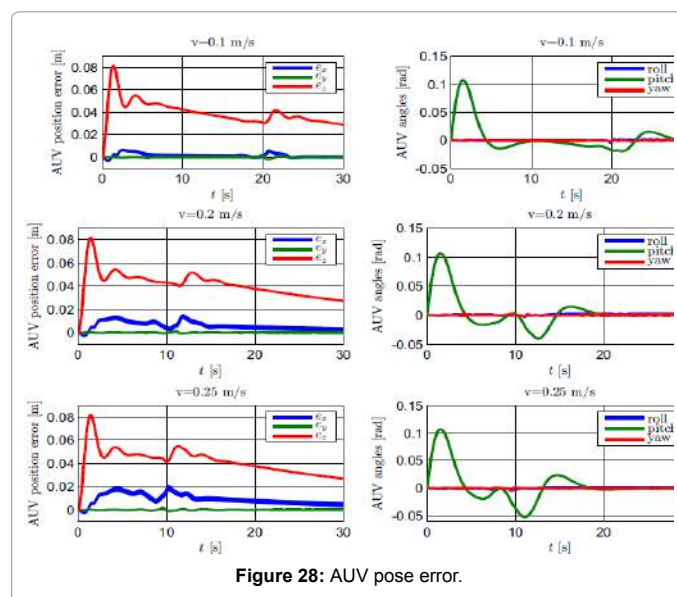


Figure 28: AUV pose error.

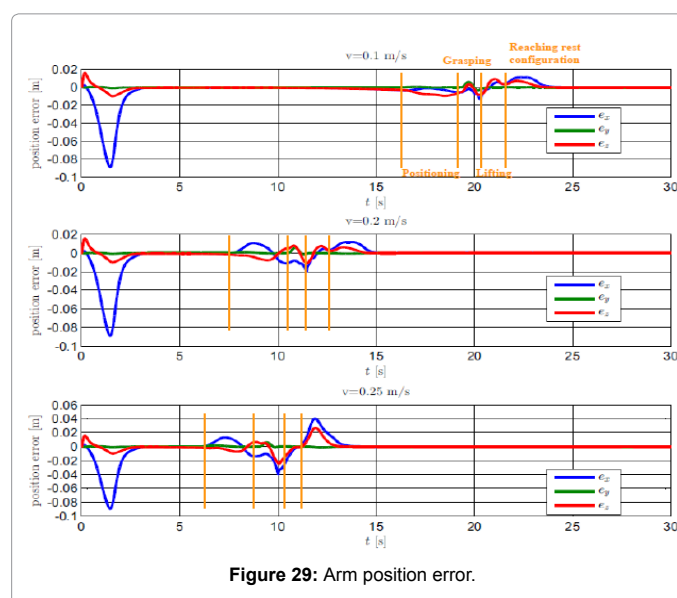


Figure 29: Arm position error.

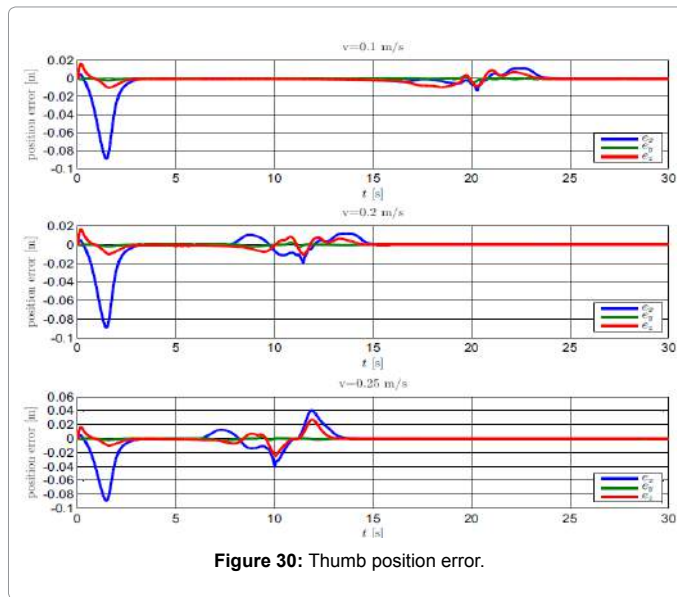


Figure 30: Thumb position error.

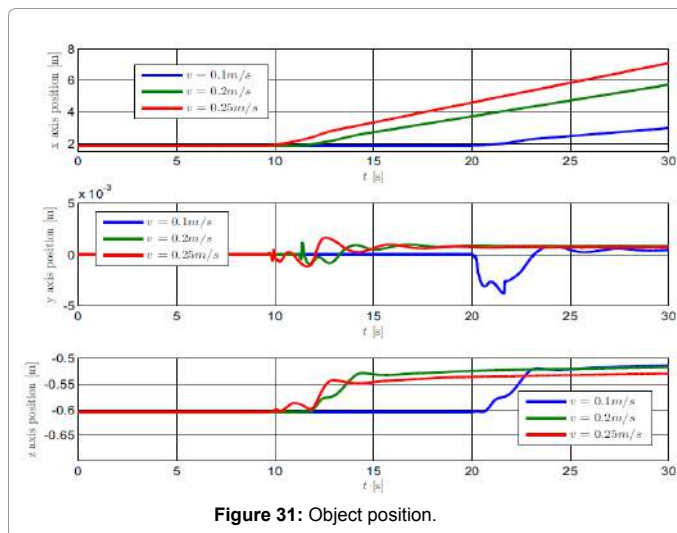


Figure 31: Object position.

the y axis) can be neglected. Vehicle's speed has little to no effect on the behaviour of the cylinder.

Conclusions and Further Developments

In this paper, the description of a control strategy for an AUV specifically thought for underwater mobile manipulation, usually called I-AUV (Intervention-Autonomous Underwater Vehicle), has been presented. In particular, dynamic manipulation has been considered; such operations need strict planning and control specifications to be successful, both for the navigation and the manipulation phases. The developed multibody model considers all the DOFs of the I-AUV, taking into account the inter-action with the fluid: classic hydrodynamic equations are coupled with the multibody approach to obtain a realistic model. At the same time, contact modelling makes possible to accurately reproduce the interaction between the gripper and the object to be manipulated. The vehicle's control maintains the AUV along the desired trajectory, modifying it, if necessary, to ease grasping and manipulation operations. A suitable grasp planning strategy has been adopted to choose good contact points on the object

surface, which are then used as reference for the position control of the manipulation system, composed of the arm and the gripper. The control of these two components has been further decoupled through an appositely developed geometrical decoupling algorithm.

The preliminary simulation results of the proposed decoupled control strategy have been satisfactory. Results obtained show that the adopted control strategy allows the execution of a dynamic manipulation task at different speeds. Steady-state speed affects the performance of the AUV and the arm controllers; further investigations are required to determine the highest speed value which makes the correct execution of the task possible. Moreover, further developments are scheduled in order to reach a more realistic simulation, e.g. the robustness of the control system, taking into account imprecision in grasping positions or the presence of disturbances from the environment, will represent for sure a next step of the proposed activity, also in view of the preliminary testing phase within the framework of the Italian project SUONO [18]: further efforts will be addressed to the acquisition process of information about the object to be grasped (e.g. acoustic or optical feedbacks) and to autonomy calculations.

Acknowledgements

This work has been partially supported by the European ARROWS project, that has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 308724.

References

1. Antonelli G (2006) Underwater robots, springer tracts in advanced robotics. Springer, (2ndedn), Heidelberg.
2. Hildebrandt M, Christensen L, Kerdels J (2009) Realtime Motion Compensation for ROV-based Tele-operated Underwater Manipulators. IEEE Robotics and Automation Magazine 8: 1-6.
3. Christensen L, Kampmann P (2009) Hardware ROV Simulation Facility for the Evaluation of novel Underwater Manipulation Techniques. IEEE Robotics and Automation Magazine 1: 1-8.
4. Ridao P, Carreras M, Ribas D, Sanz PJ, Oliver G (2014) Intervention AUVs: The Next Challenge. World IFAC, Cape Town, SA.
5. Eielsen AA (2012) Autonomous underwater robotics: State of the Art and Future Challenges, PhD Trial Lecture, NTNU, Trondheim.
6. Rigaud V, Coste-Maniere E, Perrier M, Peuch A, Simon D (1993) VORTEX: versatile and open subsea robot for technical experiments. OCEANS'93 1: 198-203.
7. Lane DM, Davies JBC, Casalino G (1997) AMADEUS: Advanced MANipulation for DEep Underwater Sampling. IEEE Robotics and Automation Magazine 4: 34-45.
8. Marani G, Choi SK, Yuh J (2009) Underwater autonomous manipulation for intervention missions AUVs. Ocean Engineering 36: 15-23.
9. <http://www.cybernetix.fr/www.cybernetix.fr%209>
10. Sanz PJ, Ridao P, Oliver G (2010) TRIDENT: a Framework for Autonomous Underwater Intervention Missions with Dexterous Manipulation Capabilities. 7th Symposium on Intelligent Autonomous Vehicles, Lecce, Italy.
11. <http://www.irs.uji.es/trident/outreach.html>
12. PANDORA FP7 project <http://persistentautonomy.com/>
13. Kim J, Chung WK, Yuh J (2003) Dynamic analysis and two-time scale control for under water vehicle manipulator systems. International Conference on Intelligent Robots and Systems 1: 571-582.
14. McLain TW, Rock SM, Lee MJ (1996) Coordinated control of an underwater robotic system. IEEE 3: 213-232.
15. Wittenburg J (2008) Dynamics of Multibody Systems, Springer, Berlin.
16. Fossen TI (1994) Guidance and control of ocean vehicles, (1stedn), John Wiley and Sons, Chichester UK.

17. <http://in.mathworks.com/>
18. SUONO project <http://www.mdm.unifi.it/CMpro-v-p-108.html>
19. ARROWS project www.arrowsproject.eu
20. Allotta B, Costanzi R, Ridolfi A (2015) The ARROWS project: Adapting and developing robotics technologies for underwater archaeology. IFAC Workshop on Navigation Guidance and Control of Underwater Vehicles, Girona, Spain.
21. Courant R, Hilbert D (1989) *Methods of Mathematical Physics*. Wiley 1: 53-7164.
22. Siciliano B, Khatib O (2008) *Handbook of Robotics*. Springer Handbooks, Napoli and Stanford.
23. Conti R, Meli E, Ridolfi A, Allotta B (2015) An innovative decentralized strategy for I-AUVs cooperative manipulation tasks. *Robotics and Autonomous Systems* 72: 261-276.
24. Allotta B, Costanzi R, Fanelli F, Monni N, Ridolfi A (2015) Single axis FOG aided attitude estimation algorithm for mobile robots. *Mechatronics*.
25. Falomi S, Malvezzi M, Meli E (2011) Multibody modeling of railway vehicles: Innovative algorithms for the detection of wheelrail contact points. *Wear* 271: 453-461.
26. Wriggers P (2006) *Computational contact mechanics*. (2nd edn), John Wiley and Sons.
27. Hartley RI, Zisserman A (2004) *Multiple View Geometry in Computer Vision*. Cambridge University Press.
28. Allotta B, Costanzi R, Meli E, Pugi L, Ridolfi A, et al. (2014) Cooperative localization of a team of AUVs by a tetrahedral configuration. *Robotics and Autonomous Systems* 62: 1228-1237.
29. Lay DC (2006) *Linear algebra and its applications*. Addison-Wesley.
30. Sciavicco L, Siciliano B (2000) *Robotica industriale Modellistica e controllo di manipolatori*. McGraw-Hill Libri Italia, (2nd edn), Milano.
31. Chiaverini S, Siciliano B, Villani L (1994) Force/Position Regulation of Compliant Robot Manipulators. *IEEE Trans on Automatic Control* 39: 647-652.
32. De Menthon Daniel F, Davis Larry S (1995) Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision* 15: 123-141.
33. Chiacchio P, Chiaverini S, Villani L, Siciliano B (1991) Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research* 27: 481-504.
34. Murray RM, Li Z (1994) *A Mathematical Introduction to Robotic Manipulation*, CRC Press.
35. Bicchi A (1995) On the closure properties of robotic grasping. *The International Journal of Robotics Research* 14: 1-25.
36. Lippiello V, Ruggiero F, Siciliano B, Villani L (2013) Visual Grasp Planning for Unknown Objects Using a Multifingered Robotic Hand. *IEEE/ASME Trans on Mechatronics* 18: 1050-1059.
37. Ponce J, Faverjon B (1991) On Computing Three-Finger Force Closure-Grasps of Polygonal Objects. *IEEE* 2: 1018-1023.
38. Mirtich B, Canny J (1994) Easily Computable Optimum Grasps in 2-D and 3-D.
39. Ding D, Liu Y, Wang S (2001) Computation of 3-D Form-Closure Grasps. *IEEE Trans on Robotics and Automation* 17: 515-522.
40. Chinellato E, Fisher RB, Morales A, Del Pobil AP (2007) Ranking Planar Grasp Configurations For A Three-Finger Hand. *IEEE International Conference on Robotics and Automation* 1: 1133-1138.
41. Kim B, Oh S, Yi B, Suh IH (2001) Optimal Grasping Based on Non-Dimensionalized Performance Indices. *IEEE/RSJ International Conference on Intelligent Robots and Systems* 2: 949-956.
42. Suárez R, Roa M, Cornella J (2006) Grasp quality measures.

Citation: Ridolfi A, Conti R, Costanzi R, Fanelli F, Meli E (2015) A Dynamic Manipulation Strategy for an Intervention: Autonomous Underwater Vehicle. *Adv Robot Autom* 4: 132. doi: [10.4172/2168-9695.1000132](https://doi.org/10.4172/2168-9695.1000132)

OMICS International: Publication Benefits & Features

Unique features:

- Increased global visibility of articles through worldwide distribution and indexing
- Showcasing recent research output in a timely and updated manner
- Special issues on the current trends of scientific research

Special features:

- 700 Open Access Journals
- 50,000 editorial team
- Rapid review process
- Quality and quick editorial, review and publication processing
- Indexing at PubMed (partial), Scopus, EBSCO, Index Copernicus and Google Scholar etc
- Sharing Option: Social Networking Enabled
- Authors, Reviewers and Editors rewarded with online Scientific Credits
- Better discount for your subsequent articles

Submit your manuscript at: <http://www.omicsonline.org/submit>