A novel approach to obstacle avoidance for an I-AUV: Preliminary Simulation Results*

Roberto Simoni [1], Pere Ridao Rodríguez[2], Patryk Cieślak[2], Dina Nagui Youakim[2]

***Abstract*— This paper presents a novel approach to obstacle avoidance approach for an I-AUV in a framework of set-based task-priority kinematic control algorithm. The approach is divided into two modes: Mode (1) navigation and inspection and Mode (2) intervention. For navigation we fully wrap the I-AUV with two safety spheres at the vehicle and one at the arm. For intervention we use more safety spheres with smaller sizes to fully wrap the I-AUV to allow more precise movements of the I-AUV near the intervention areas.The novel approach was implemented and simulated with the 8-DOF I-AUV GIRONA500 in a scenario for inspection and maintenance (valve turning) of a BOP (blowout preventer) structure used in oil and gas industry. The BOP structure was represented by an octomap and each occupied cell of the octomap was considered as an obstacle in our model.**

I. Introduction

A robot-manipulator is kinematically redundant if it possesses more DOFs than those required to execute a given task [1]. For instance, the I-AUV GIRONA 500 is an inherently redundant vehicle-manipulator because it has 8-DOFs. This redundancy can be explored in the context of task-priority to meet additional tasks apart from tracking the end-effector configuration.

In this paper we implemented a set-based task-priority algorithm and an obstacle avoidance task. We simulated the obstacle avoidance algorithm in a scenario with the 8-DOF I-AUV GIRONA 500.

II. Proposed Approach for Obstacle Avoidance of a I-AUV

For the purpose of obstacle avoidance we define virtual safety spheres to fully wrap the I-AUV and virtual tolerance spheres to fully wrap the obstacles which are represented by octomap. The proposed approach requires the definition of two operation modes for obstacle avoidance:

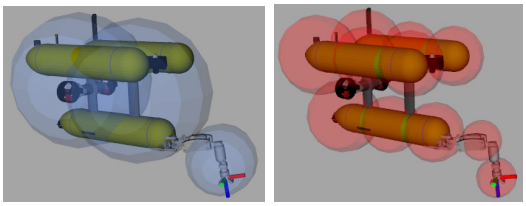
- Mode 1 - navigation and inspection: with two safety spheres to fully wrap the vehicle and one to fully wrap the arm (see the kinematic model in Fig. 1(a)); and
- Mode 2 - intervention: with nine safety spheres to fully wrap the vehicle and two to fully wrap the arm (see the kinematic model in Fig. 1(b)).

*The first author would like to tanks CAPES Foundation by the financial support. "Bolsista da CAPES - Proc. n 88881.156728/2017-01".

[1]Federal University of Santa Catarina, Rua Dona Francisca, 8300, Bloco U, CEP 89.219-600 Joinville/SC, Brazil. roberto.simoni@ufsc.br

[2]Centre d'Investigació en Robòtica Submarina, Girona, Spain.Computer Vision and Robotics Research Institute, Scientific and Technological Park of the University of Girona, CIRS Building C/ Pic de Peguera, CO 17071 Girona Catalonia, Spain pere@eia.udg.edu, patrik.cieslak@udg.edu, disaac@udg.edu

For each mode we defined a task and to enable and disable the mode we switch the tasks in the task hierarchy. In the proposed approach, the I-AUV follows a mission with several waypoints for "inspection" and for "intervention". Each waypoint has the position given by a vector and the orientation given by a quaternion. The resolution of the octomap is 0.03m (see Fig 2(a)). For Mode 1 - navigation and inspection - we access a coarser resolution of the octomap and for Mode 2 we access a very high resolution to allow more precise movements of the I-AUV near the structure to perform the intervention. For Mode 1, we access all the octomap with depth 11 and we define an obstacle for each occupied cell which will result in 200 obstacles (represented in our model by spheres) of size 0.96m to fully wrap the BOP structure (see Fig 2(b)). For Mode 2, we access the octomap depth 14 in a radius of 2m from the intervention waypoint which provide obstacles of size 0.12m, the number of obstacles to fully wrap the BOP structure depends on the position of the waypoint (see Figs 2(c) and 2(d)).



(a) (b)

Fig. 1. Safety spheres to fully wrap the I-AUV for: (a) Mode 1 and (b) Mode 2.

For obstacle avoidance we implement a task in the framework of a task-priority algorithm and this task will be enabled when one of the safety spheres at the I-AUV (see Fig. 1) reaches the border of an obstacle (see Fig 2).

III. Mathematical Background

For a general robotic system with n-DOFs in a given configuration $\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^T$ it is possible to express a tasks σ_i in the operational space through forward kinematics

$$\sigma_i(t) = \sigma_i(\boldsymbol{q}(t)) \tag{1}$$

The velocities in the operational space for the task σ_i are given by

$$\dot{\boldsymbol{x}}_E = J(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2}$$

where $J(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix and $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ are the joint velocities.

The inverse solution to the differential kinematics mapping is given by

$$\dot{\boldsymbol{q}} = J^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}}_E, \tag{3}$$

where $J^{\dagger}(\boldsymbol{q})$ is the Moore-Penrose pseudoinverse of the end-effector Jacobian matrix.

For a redundant robotic system, the inverse solution of the differential kinematics mapping which takes in account the null-space projector is given by

$$\dot{\boldsymbol{q}} = J^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}}_E + (I_n - J^{\dagger}(\boldsymbol{q})J(\boldsymbol{q}))\dot{\boldsymbol{q}}_{null} \tag{4}$$

where $\dot{\boldsymbol{q}}_{null} \in \mathbb{R}^n$ represents an arbitrary joint velocity and $\dot{\boldsymbol{x}}_E = \sigma_i + K_i\tilde{\sigma_i}$ represents the desired cartesian velocity with $\tilde{\sigma_i} = \sigma_{i,d} - \sigma_i$ being the task error and K_i a diagonal gain matrix.

It is well known the operator $P = I_n - J^{\dagger}J$ projects $\dot{\boldsymbol{q}}_{null}$ in the null space of the Jacobian matrix. In simple words, the velocities $\dot{\boldsymbol{q}}_{null}$ projected in the null space $N = I_n - J^{\dagger}J$ do not affect the cartesian velocities $\dot{\boldsymbol{x}}_E$ to complete the task. In this sense, additional tasks can be handled in a framework of task-priority such as, joint limits, obstacle avoidance, minimization of joint torques, control of the orientation of directional sensors, and maximization of arm manipulability, maximization of different dexterity measures, and so on [1]–[3].

A. Task-priority algorithm

In agreement with Siciliano and Slotine [4], the algorithmic general solution for N tasks in a task-priority framework is given by

$$\dot{\boldsymbol{q}}_i = \dot{\boldsymbol{q}}_{i-1} + \bar{J}_i^{\dagger}(\boldsymbol{q})\left(\dot{x}_i - J_i(\boldsymbol{q})\dot{\boldsymbol{q}}_{i-1}\right), \qquad i = 1, \ldots, N \tag{5}$$

where $\bar{J}_i(\boldsymbol{q}) = J_i(\boldsymbol{q})P_{A,i-1}(\boldsymbol{q})$ and $P_{A,i}(\boldsymbol{q}) = I - J_{A,i}^{\dagger}(\boldsymbol{q})J_{A,i}(\boldsymbol{q})$. The algorithm initialization requires $\dot{\boldsymbol{q}}_0 = 0_{n\times 1}$ and $P_0 = I_{n\times n}$ [2].

The task-priority algorithm was implemented in the context of set-based tasks. In this sense, a task can enabled or disabled depending on the task value. If the range of valid values for the task σ is $D = [\sigma_{min}, \sigma_{max}]$ the task will be enabled on the border of D to prevent a violation of the set D. For obstacle avoidance tasks, we consider the radius r_i of the i-th obstacle to define the set-based $D_i = [r_i, \infty)$. The task-priority algorithm predict the next value of $\boldsymbol{q}$, *i.e.* $\boldsymbol{q}_{i+1}$ predicted, and if any violation is detected for the predicted $\boldsymbol{q}$ the task will be enabled and the task-priority algorithm will be recomputed to avoid such violation in the next iteration.

B. Obstacle avoidance task

The goal of the obstacle avoidance task is to not allow the intersection between the safety and tolerance spheres. To avoid the intersection between the i-th tolerance sphere with radius ρ at the position $\boldsymbol{p}_O \in \mathbb{R}^3$ and the j-th safety sphere with radius ϱ at the vehicle (or arm) at the position $\boldsymbol{p}_S \in \mathbb{R}^3$ we implemented a scalar task which is given by the distance between them

$$\sigma_{OA}^{ij} = \sqrt{(\boldsymbol{p}_O - \boldsymbol{p}_S)^T(\boldsymbol{p}_O - \boldsymbol{p}_S)} \in \mathbb{R} \tag{6}$$

The task error will be given by the distance between the i-th tolerance sphere and the j-th safety sphere, *i.e.*,

$$\widetilde{\sigma}_{OA}^{ij} = k[\sigma_{OA}^{ij} - (\rho + \varrho)] \tag{7}$$

where k is a scalar gain.

The Jacobian will be a single-entry row matrix given by

$$J_{OA}^{ij} = -\frac{(\boldsymbol{p}_O - \boldsymbol{p}_S)^T}{\sigma_{OA}} J_{pos}^{j}(\boldsymbol{q}) \tag{8}$$

where J_{pos}^{j} is the position jacobian of the j-th safety sphere.

In the general case, where we have multiple tolerance spheres at the border of multiple safefy spheres, we use a vector notation to describe

$$\boldsymbol{\sigma}_{OA} = \begin{bmatrix} \sigma_{OA}^{11} \\ \sigma_{OA}^{12} \\ \vdots \\ \sigma_{OA}^{ij} \\ \vdots \\ \sigma_{OA}^{m(n-1)} \\ \sigma_{OA}^{mn} \end{bmatrix} \tag{9}$$

The error will be given by

$$\widetilde{\boldsymbol{\sigma}}_{OA} = \begin{bmatrix} \widetilde{\sigma}_{OA}^{11} \\ \widetilde{\sigma}_{OA}^{12} \\ \vdots \\ \widetilde{\sigma}_{OA}^{ij} \\ \vdots \\ \widetilde{\sigma}_{OA}^{m(n-1)} \\ \widetilde{\sigma}_{OA}^{mn} \end{bmatrix} \tag{10}$$

The Jacobian will be given by

$$J_{OA} = \begin{bmatrix} J_{OA}^{11} \\ J_{OA}^{12} \\ \vdots \\ J_{OA}^{ij} \\ \vdots \\ J_{OA}^{m(n-1)} \\ J_{OA}^{mn} \end{bmatrix} \tag{11}$$

The repulsive velocity in the joint space is given by Equation 5

$$\dot{\boldsymbol{q}}_i = \dot{\boldsymbol{q}}_{i-1} + \bar{J}_{OA}^{\dagger}(\boldsymbol{q})\left(\widetilde{\boldsymbol{\sigma}}_{OA} - J_{OA}(\boldsymbol{q})\dot{\boldsymbol{q}}_{i-1}\right) \tag{12}$$

1) Random noise task: It may happen that only with the obstacle avoidance task the I-AUV get trapped by several obstacles and in this case it may be interesting to use a random noise task to try to take it away from the local minima and pass by the obstacles. The random noise task will be enabled when the obstacle avoidance task is enabled for a while and distance traveled by the end-effector is less than a desired value which means the vehicle is trapped.

To run a random noise task with the algorithm from Equation 5 we need to generate a random task error which will be given by the scalar

$$\widetilde{\sigma}_{RN} = rand(), rand_{min} \leq rand() \leq rand_{max} \quad (13)$$

and the Jacobian which will be a single-entry row matrix given by

$$J_{RN} = \frac{(\eta)^T}{|\eta|} J^{ee}_{pos}(\boldsymbol{q}) \quad (14)$$

where η is a random vector and J^{ee}_{pos} is the position jacobian of the end-effector.

The random velocity in the joint space is given by Equation 5

$$\dot{\boldsymbol{q}}_i = \dot{\boldsymbol{q}}_{i-1} + \bar{J}^{\dagger}_{RN}(\boldsymbol{q}) \left(\widetilde{\sigma}_{RN} - J_{RN}(\boldsymbol{q})\dot{\boldsymbol{q}}_{i-1}\right) \quad (15)$$

IV. Simulation

Figure 2 shows the simulation of the proposed approach for inspection and intervention in a BOP structure used in oil and gas industry.

For the simulation, we proposed a mission with several waypoints for inspection and for intervention in a BOP structure used in oil and gas industry as shown in Figure 2. The BOP structure was modeled by an octomap which can be accessed at different resolutions [5]. The intervention consists of turning a valve on a panel inside of the BOP structure.

V. Conclusions

In this paper we presented a novel approach for obstacle avoidance based on two operation modes: Mode 1 - navigation and inspection and Mode 2 - intervention. The proposed approach was implemented in the context of a task-priority algorithm and as a preliminary result we present the simulation in a BOP structure used in oil and gas industry.

References

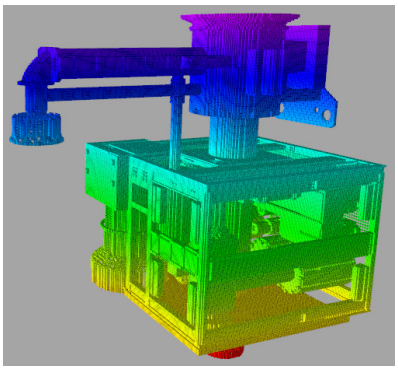
[1] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[2] P. Cieslak, P. Ridao, and M. Giergiel, "Autonomous underwater panel operation by girona500 uvms: A practical approach to autonomous underwater manipulation," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 529–536, IEEE, 2015.

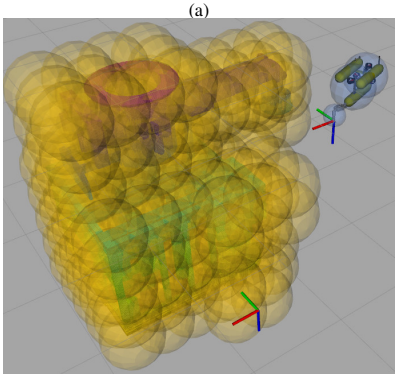
[3] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, p. 16, 2016.

[4] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," *Advanced Robotics*, pp. 1211–1216, 1991.

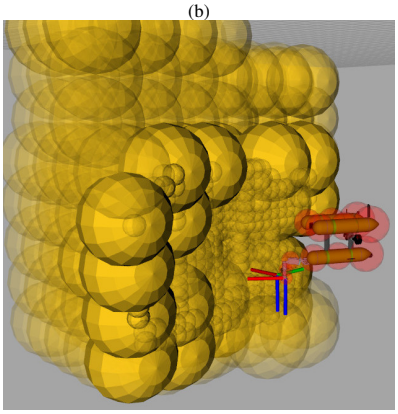
[5] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.



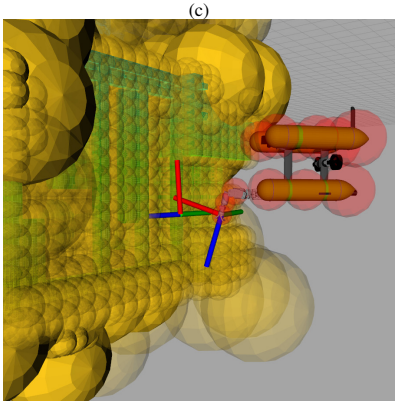
(a)



(b)



(c)



(d)

Fig. 2. (a) BOP represented by the octomap. (b) Obstacles model for Mode 1 with a coarser resolution of the octomap. (c) Obstacles model for Mode 2 with a high resolution of the octomap in a distance of 2m from the intervention waypoint. (d) Mode 2 closer view.