Cooperative Assembly with Autonomous Mobile Manipulators

in an Underwater Scenario

Davide Torielli

Supervisors: G. Casalino[1], E. Simetti[1], P. J. Sanz[2], R. Marin[2]

[1]DIBRIS - Università degli studi di Genova, Italy
[2]IRSLab - Universitat Jaume I, Castellón de la Plana, Spain

Presentation of the Master Thesis for the degree of *Robotics Engineering*

30th August, 2019

Why Underwater Robots

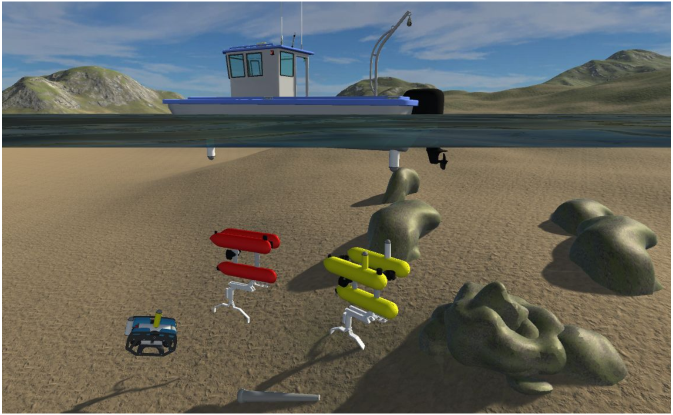
Because they can operate in difficult environments which are very important to reach.

Aquanaut, the Underwater Transformer, from *Houston Mechatronics*

- Pollution Monitoring *[SWARMs, 2018]*
- Mine disposal *[Lopes et al., 2017]*
- **Oil and gas industry** *[Diaz Ledezma et al., 2015]*
  pipe inspection, opening and closing valves, drilling, rope cutting ...

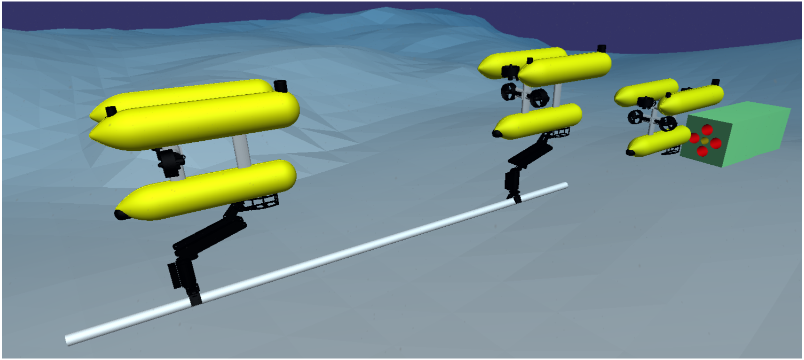
TWINBOT Project

TWIN roBOTs for cooperative underwater intervention missions



Simulated environment with the cooperative robots approaching a pipe

The Scenario

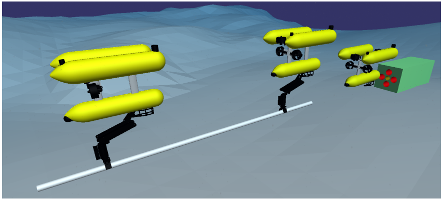


UWSim screenshot with the two Carrying robots and the Vision robot

The Scenario

The Peg-In-Hole Problem

It is performed by two autonomous mobile manipulators in an underwater scenario.



- One Vision robot to estimate the hole's pose
  - Detection & Tracking with computer vision algorithms
- Two Carrying robots which *cooperate* at kinematic level to transport and to insert the tool
  - Task Priority Inverse Kinematic (TPIK) for controlling the systems
  - Force-Torque Sensor to help the insertion phase

Task Priority Inverse Kinematic

During a Mission, there are different objectives (reaching a goal, avoiding joint limits, *reducing collisions between the peg and the hole*, ...).

The aim of the kinematic layer is to provide a system velocity reference vector $\bar{\dot{\boldsymbol{y}}} \in \mathbb{R}^{dof \times 1}$ which satisfies *at best* these objectives.

$$S_k \triangleq \left\{ \arg \underset{\dot{\boldsymbol{y}} \in S_{k-1}}{\mathrm{R}\text{-}\min} \left\| \boldsymbol{A}_k (\bar{\dot{\boldsymbol{x}}}_k - \boldsymbol{J}_k \dot{\boldsymbol{y}}) \right\|^2 \right\}, \; k = 1, 2, \ldots, N$$

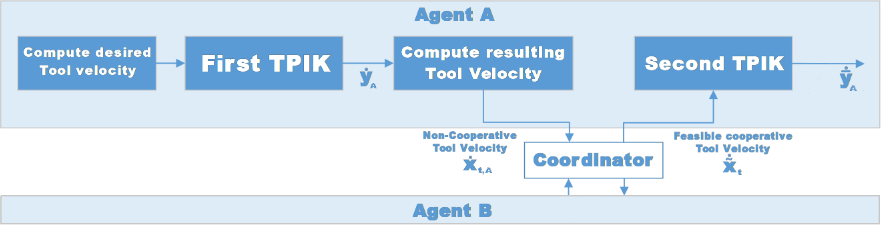
- $\boldsymbol{A}_k \in \mathbb{R}^{m_k \times m_k}$ diagonal matrix of activation functions
- $\bar{\dot{\boldsymbol{x}}}_k \in \mathbb{R}^{m_k \times 1}$ desired feedback reference rate
- $\boldsymbol{J}_k \in \mathbb{R}^{m_k \times dof}$ task-induced Jacobian (m_k is the k-th task dimension)
- $\boldsymbol{S}_{k-1}$ manifold of solutions of higher priority tasks
- R- min indicates “min” operator done with a series of regularizations *[Simetti and Casalino, 2016]*

Task Priority Inverse Kinematic

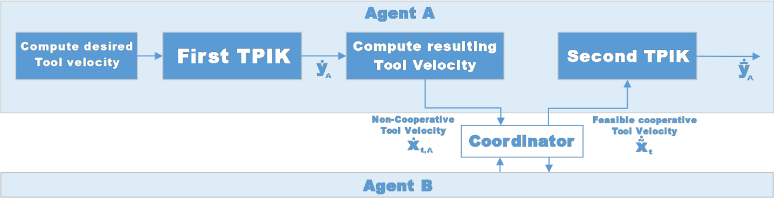
The cooperation

taken from *[Simetti and Casalino, 2017]* and *[Wanderlingh, 2018]*.

The two robots must generate an *identical* tool velocity $\dot{\boldsymbol{x}}_t$.



Each agent i runs a first TPIK as if it is alone. Then, it computes the *non-cooperative velocity* $\dot{\boldsymbol{x}}_{t,i} = \boldsymbol{J}_{t,i}\dot{\boldsymbol{y}}_i$ and sends it to the coordinator.



The coordinator computes the *non-feasible* cooperative Tool velocity $\dot{\hat{\boldsymbol{x}}}_t$:

$$\dot{\hat{\boldsymbol{x}}}_t = \frac{1}{\mu_A + \mu_B}(\mu_A \dot{\boldsymbol{x}}_{t,A} + \mu_B \dot{\boldsymbol{x}}_{t,B})$$

$$\mu_i = \mu_0 + \|\dot{\bar{\boldsymbol{x}}}_t - \dot{\boldsymbol{x}}_{t,i}\| \triangleq \mu_0 + \|\boldsymbol{e}_i\|, \quad i = A, B$$

Then it projects it into the space of the achievable object velocity $ker(\boldsymbol{J}_{t,A}\boldsymbol{J}^{\#}_{t,A} - \boldsymbol{J}_{t,B}\boldsymbol{J}^{\#}_{t,B})$, and provides $\dot{\tilde{\boldsymbol{x}}}_t$ to the robots.

The two robots run a second TPIK, with the task of following the cooperative Tool velocity $\dot{\tilde{\boldsymbol{x}}}_t$ at the highest priority.

The Force-Torque Sensor

The simulator UWSim provides force-torque sensor data caused by collisions on the peg. It is used for:

- Adding arm and vehicle movements caused by peg collisions:

$$\dot{\boldsymbol{y}}_\delta \triangleq \begin{bmatrix} \dot{\boldsymbol{q}}_\delta \\ \boldsymbol{v}_\delta \\ \boldsymbol{w}_\delta \end{bmatrix} = \begin{bmatrix} k_q \\ k_v \\ k_w \end{bmatrix} \left[\left({}^{lin}\boldsymbol{J}_t\right)^T \boldsymbol{f} + \left({}^{ang}\boldsymbol{J}_t\right)^T \boldsymbol{m} \right] \qquad 0 < k_q, k_v, k_w < 1$$

  formula from *[Siciliano, Luigi Villani, and Oriolo, 2009]*.

- Changing the goal frame towards the peg is driven to by the control

- Adding a new objective in the TPIK list

The Force-Torque Objective

Aim: to nullify the forces and the torques caused by the collisions between the peg and the inner walls of the hole.

The **feedback reference rate** is:

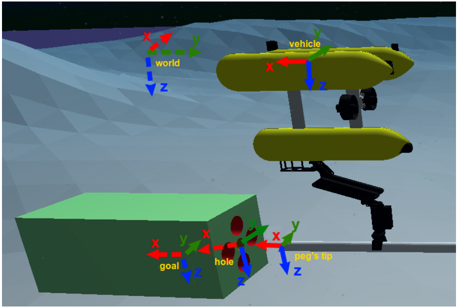
$$\dot{\bar{\boldsymbol{x}}}_{ft} \triangleq \begin{bmatrix} \dot{\bar{x}}_f \\ \dot{\bar{x}}_m \end{bmatrix} \triangleq \begin{bmatrix} \gamma_f \\ \gamma_m \end{bmatrix} \begin{bmatrix} 0 - \|\boldsymbol{f}\| \\ 0 - \|\boldsymbol{m}\| \end{bmatrix} \qquad 0 < \gamma_f < 1, \quad 0 < \gamma_m < 1$$

In the TPIK hierarchy, this objective is put before the reaching goal one.

The activation function $\boldsymbol{A}_{ft}$ is used to smooth the behaviour but also to deactivate the task when the norm is near to zero.

Assumptions

- Except the collisions, no dynamics is simulated (e.g. $\dot{\boldsymbol{y}}$ is the real velocity)
- No problems caused by the peg collisions with the external hole's surface
- Peg firmly grasped by both robots



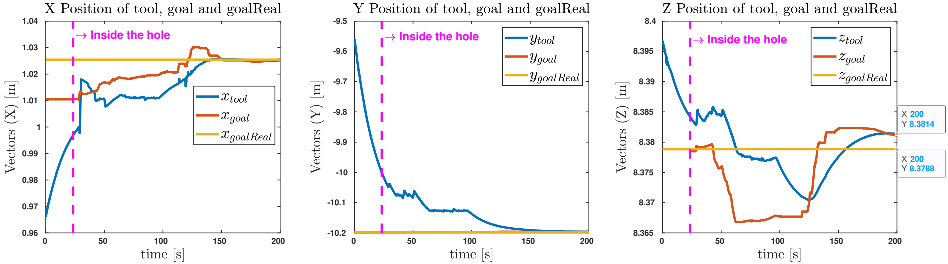
Main frames involved

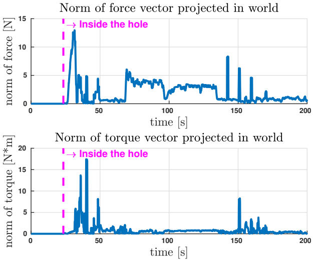
- There is a common reference frame known to the Carrying robots and to the Vision one
- Experiment details:
  - Peg: length 6m; diameter 0.1m
  - Hole: diameter 0.14m
  - Goal is to drive the peg's tip 0.2m inside the hole

Results

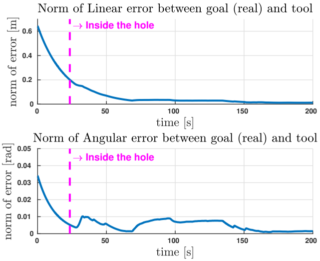
Experiment with error on hole's pose of 0.015 m on x axis

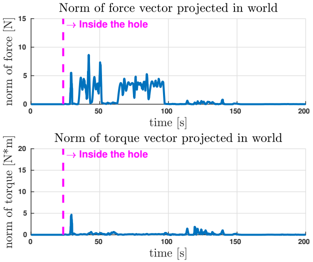
Position of the tool and of the changing goal
divided in the three components



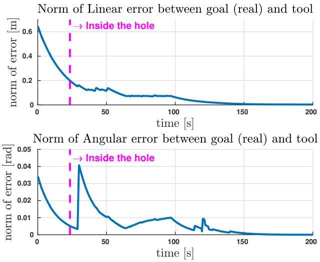


Without Change Goal nor FT objective





With Change Goal and FT objective



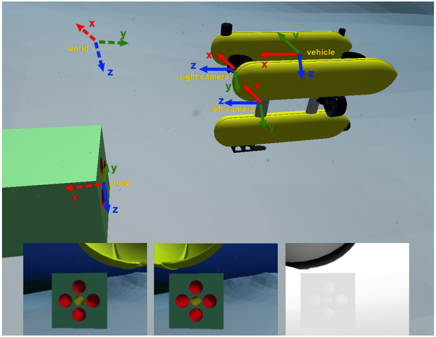
Vision part

Hole's pose estimation

Two steps: *Detection* and *Tracking*

Detecting the hole in the scene is necessary to initialize the Tracking algorithm.

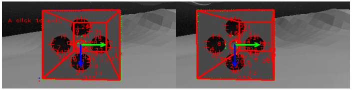
- Find Square *[Suzuki and Be, 1985]*
- Template Matching



Principal frames involved and cameras

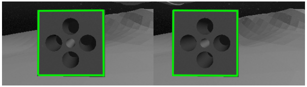
Tracking is done to compute in real-time the pose, even while moving. It is a model-based Tracking performed with the library ViSP *[ViSP, 2019]*.

- Mono Camera
- Stereo Camera
- Stereo-depth Camera



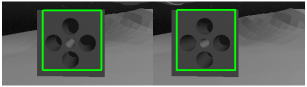
Left and Right Tracking (Stereo)

Find Square



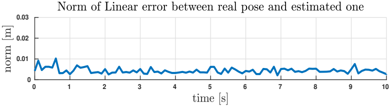
Left and Right Detection Result

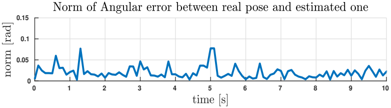
Template Matching

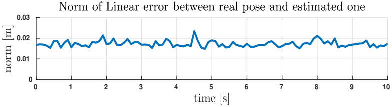


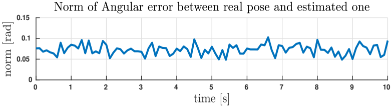
Left and Right Detection Result

Mono Camera (left one)

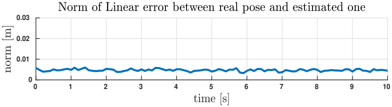


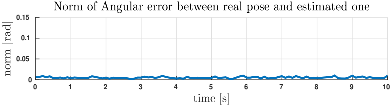


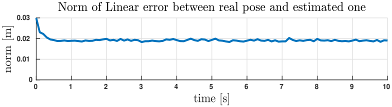


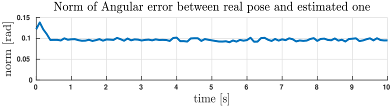


Stereo Cameras









Video of the final experiment

The video is also visible here `https://streamable.com/kvoxq`.

Conclusions

- It has been presented a Control Architecture for the stated problem.
- Computer Vision algorithms have been explored.
- A force-torque sensor has been exploited at kinematic level.
- Anyway, the problem is wide and largely unexplored, so further work is necessary in different direction.
  - Dynamics (e.g. with FreeFloatingGazebo *[Kermorgant, 2014]*)
  - More complex peg-in-hole missions
  - Relaxing Vision assumptions
  - Communication and different assembly problems *[TWINBOT, 2019]*

Thank you for the attention!

References I

Diaz Ledezma, Fernando et al. (2015). "A Market Survey of Offshore Underwater Robotic Inspection Technologies for the Oil and Gas Industry". In: DOI: 10.2118/177989-MS.

Kermorgant, Olivier (2014). "A dynamic simulator for underwater vehicle-manipulators". In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots Simpar*. Bergamo, Italy: Springer. URL: https://hal.inria.fr/hal-01065812.

Lopes, Luís et al. (2017). "UNEXMIN: developing an autonomous underwater explorer for flooded mines". In: *Energy Procedia* 125, pp. 41–49. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2017.08.051. URL: http://www.sciencedirect.com/science/article/pii/S1876610217335336.

Siciliano, Bruno, Lorenzo Sciavicco and Luigi Villani, and Giuseppe Oriolo (2009). "Robotics: Modelling, Planning and Control". In: Springer-Verlag London, pp. 147–151. DOI: 10.1007/978-1-84628-642-1.

Simetti, E. and G. Casalino (2016). "A Novel Practical Technique to Integrate Inequality Control Objectives and Task Transitions in Priority Based Control". In: *Journal of Intelligent & Robotic Systems* 84. DOI: 10.1007/s10846-016-0368-6.

References II

Simetti, E. and G. Casalino (2017). "Manipulation and Transportation With Cooperative Underwater Vehicle Manipulator Systems". In: *IEEE Journal of Oceanic Engineering* 42.4, pp. 782–799. ISSN: 0364-9059. DOI: 10.1109/JOE.2016.2618182.

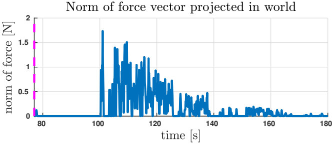
Suzuki, Satoshi and Keiichi A. Be (1985). "Topological structural analysis of digitized binary images by border following". In: *Computer Vision, Graphics, and Image Processing* 30.1, pp. 32–46. ISSN: 0734-189X. DOI: https://doi.org/10.1016/0734-189X(85)90016-7. URL: http://www.sciencedirect.com/science/article/pii/0734189X85900167.

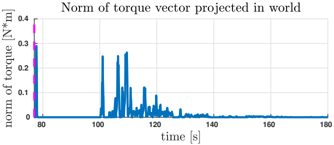
SWARMs (2018). *Swarms website*. URL: http://www.swarms.eu/ (visited on 2019).

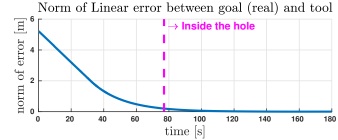
TWINBOT (2019). *Twinbot website*. URL: http://www.irs.uji.es/twinbot/twinbot.html (visited on 2019).

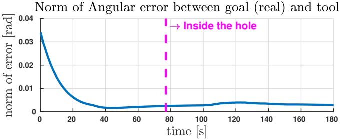
ViSP (2019). *ViSP website*. URL: https://visp.inria.fr/robot-interface/ (visited on 2019).

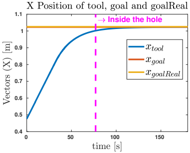
Wanderlingh, Francesco (2018). "Cooperative Robotic Manipulation for the Smart Factory". PhD thesis. Università degli Studi di Genova.

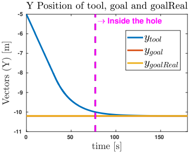

Norm of force vector projected in world
norm of force [N]
time [s]

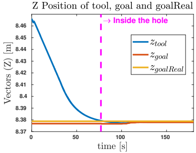

Norm of torque vector projected in world
norm of torque [N*m]
time [s]

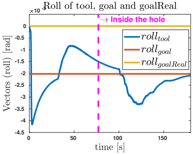

Norm of Linear error between goal (real) and tool
→ Inside the hole
norm of error [m]
time [s]

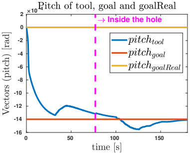

Norm of Angular error between goal (real) and tool
→ Inside the hole
norm of error [rad]
time [s]

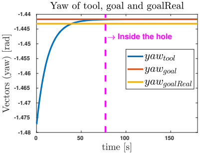

X Position of tool, goal and goalReal
→ Inside the hole
x_{tool}
x_{goal}
$x_{goalReal}$
Vectors (X) [m]
time [s]

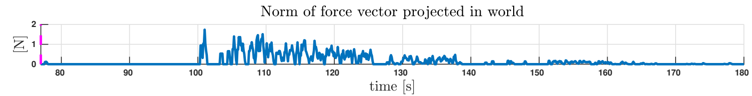

Y Position of tool, goal and goalReal
→ Inside the hole
y_{tool}
y_{goal}
$y_{goalReal}$
Vectors (Y) [m]
time [s]

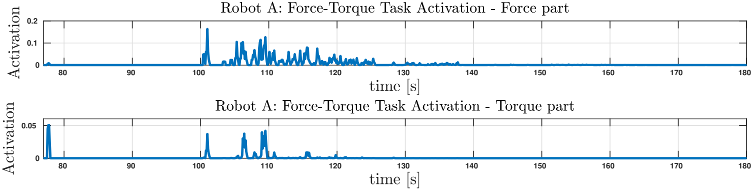

Z Position of tool, goal and goalReal
→ Inside the hole
z_{tool}
z_{goal}
$z_{goalReal}$
Vectors (Z) [m]
time [s]

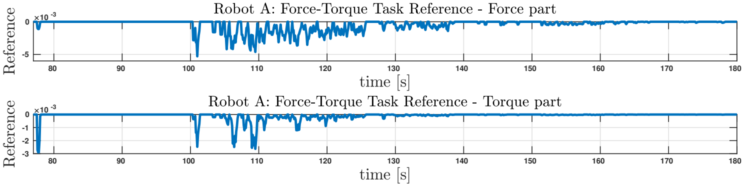

Roll of tool, goal and goalReal
→ Inside the hole
$roll_{tool}$
$roll_{goal}$
$roll_{goalReal}$
Vectors (roll) [rad]
time [s]


Pitch of tool, goal and goalReal
→ Inside the hole
$pitch_{tool}$
$pitch_{goal}$
$pitch_{goalReal}$
Vectors (pitch) [rad]
time [s]


Yaw of tool, goal and goalReal
→ Inside the hole
yaw_{tool}
yaw_{goal}
$yaw_{goalReal}$
Vectors (yaw) [rad]
time [s]


Norm of force vector projected in world
[N]
time [s]
Norm of torque vector projected in world
[N*m]
time [s]


Robot A: Force-Torque Task Activation - Force part
Activation
time [s]
Robot A: Force-Torque Task Activation - Torque part
Activation
time [s]


Robot A: Force-Torque Task Reference - Force part
Reference
time [s]
Robot A: Force-Torque Task Reference - Torque part
Reference
time [s]