# Optimization of AUVs propulsion system for underwater infrastructures monitoring

*MOQESM'16 - Sea Tech Week*

Pablo Vega

PhD student at
**Institut de Recherche Dupuy de Lôme**
**CNRS FRE 3744**
PTR4 : Énergie et Systèmes - ENIB
Supervisor: O. Chocron    Director: M. Benbouzid

**Brest, 12th October 2016**

ÉCOLE NATIONALE D'INGÉNIEURS DE BREST

Institut de Recherche Dupuy de Lôme

université de bretagne occidentale

# Outline

# Motivation

- AUV can be trusted to carry out complex missions
- However, AUV development has not yet reached its full potential
  → their capabilities can be still greatly improved.
- One of these characteristic is maneuverability, a key factor to achieve full autonomy.
- An enhanced maneuverability along with the possibilities offered by current control methods and sensor technology could increase greatly the complexity and reliability of AUV missions

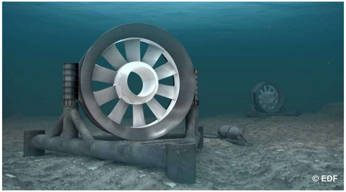



**Fig. 1.** Marine turbine. EDF

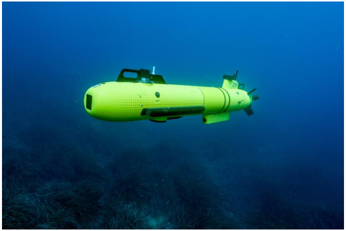


**Fig. 2.** A9-EAUV ECA group

Institut de Recherche Dupuy de Lôme

# Our approach

- To give AUVs the capability to reconfigure their propulsion.
- Adapt the propulsive topology and control parameters to the task dynamically.
- Use of vectorial thruster to achieve propulsion reconfiguration.
- Use of model-based controller to cancel out nonlinearities.
- A task-based design optimized propulsion topology and controller increases maneuverability, leading to enhanced AUV autonomy.

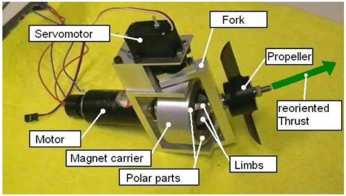



**Fig. 3**. RMCT prototype. IRDL

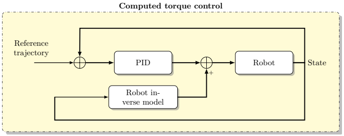
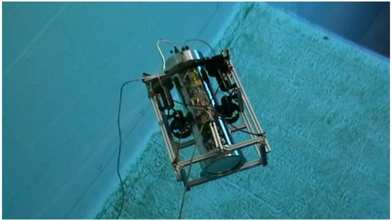



**Fig. 4**. Model based controller

In order to implement our propulsion optimization approach we need to develop the following elements:

- A dynamic model (solid and hydro) of an AUV. Based on prototype developed in IRDL: the RSM robot
- A control method allowing us to follow a task trajectory.
- An optimization method in order to find an optimal propulsive configuration (topology and control parameters) for a given mission.



RSM robot (IRDL-ENIB) in Ifremer bassin

The vectors describing the movement of the AUV in 6 DOF are:

### Position and orientation in $R_0$

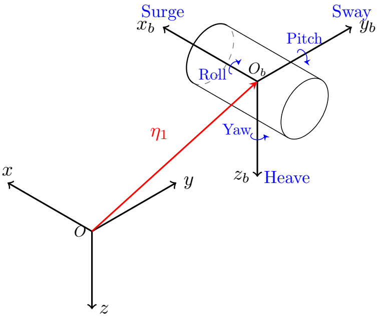
$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \quad \eta_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \eta_2 = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

### Linear and angular velocity in $R_b$

$$\nu = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \quad \nu_1 = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \nu_2 = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

### Efforts in $R_b$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad \tau_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \tau_2 = \begin{bmatrix} K \\ M \\ N \end{bmatrix}$$




Coordinate frames describing the AUV

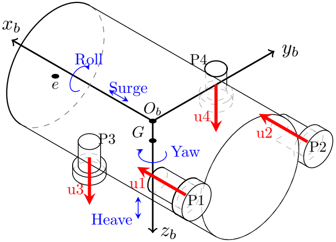
The equation describing the underwater robot solid and hydro dynamics is the following (Fossen):

$$\mathbf{M}\dot{\nu} + \mathbf{C}\nu + \mathbf{D}\nu + \mathbf{G} = \boldsymbol{\tau} = \mathbf{B}\,\mathbf{u}_p$$

with

- $\mathbf{M} = \mathbf{M_{RB}} + \mathbf{M_A}$, mass and inertia matrices
- $\mathbf{C} = \mathbf{C_{RB}} + \mathbf{C_A}$, coupling matrices
- $\mathbf{D}$ = Damping matrix
- $\mathbf{G}$ = Gravity and buoyancy
- $\mathbf{B}$ = Thruster configuration matrix
- $\mathbf{u}_p$ = Thrusters forces (thrust)

$$\boldsymbol{\tau} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -P_{3y} & -P_{4y} \\ P_{1z} & P_{2z} & P_{3x} & P_{4x} \\ -P_{1y} & -P_{2y} & 0 & 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}}_{\mathbf{u}_p}$$

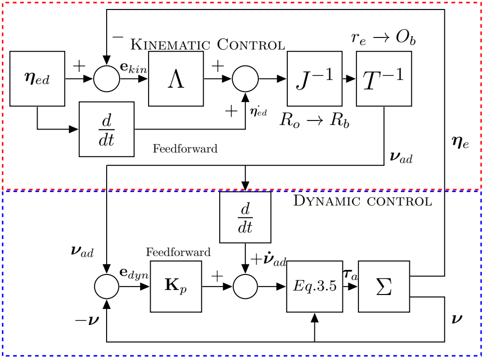


RSM robot propulsive topology

- **B** is the matrix including the position and orientation of thrusters.
- To implement the dynamic reconfiguration we modify this matrix.
- Each thruster is dynamically modeled as an electromechanical device.

### Principle

- The idea is to algebraically transform nonlinear systems into (fully or partly) linear ones in order to apply linear control methods.
- It creates a control input able to cancel out the nonlinear effects (inertia, coupling, drag, gravitational forces and buoyancy)



Two control laws in cascade:

## Kinematic law: Velocity

- derived from the robot inverse kinematic model.
- generates the velocity reference to follow the desired trajectory.

$$\boldsymbol{\nu}_{ad} = \mathbf{T}^{-1}\left\{\mathbf{J}^{-1}\left[\dot{\boldsymbol{\eta}}_{ed} + \boldsymbol{\Lambda}\,\mathbf{e}_{kin}\right]\right\}$$

with $\mathbf{T}$ a velocity transport matrix (rigid body kinematics)

## Dynamic law: Acceleration

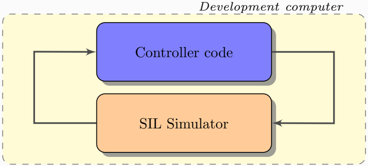
- derived from the robot complete dynamic model.
- generates the control input necessary to follow the reference velocity.

$$\boldsymbol{\tau}_a = \mathbf{M}[\dot{\boldsymbol{\nu}}_{ad} + \mathbf{K_p}\mathbf{e}_{dyn}] + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{G}$$

## Software-in-the-Loop (SIL)

Control algorithm coded in the simulator (EAUVIVE) as if will be coded in the robot processor: same source code, but using the PC CPU (programmed under ROS/C++ and MATLAB).

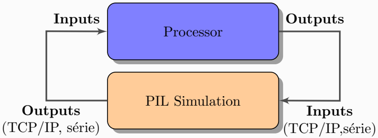
Allows to test the code in order to find errors in the control algorithm.




## Processor-in-the-Loop (PIL)

The control algorithm is compiled for and executed in the robot CPU. The simulation runs on the PC (ROS/C++ and MATLAB).

Allows to detect errors related to the processor (compilation, memory, optimization, performance, latency)



# Dynamic reconfiguration

## Principle

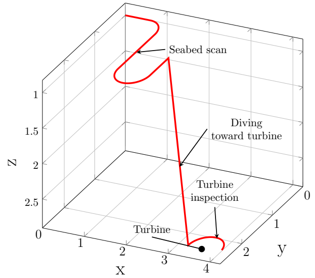
For a mission consisting on multiple tasks:

- One topology $\Rightarrow$ hardly multitasking
- Completely modify the topology $\Rightarrow$ not technically possible (for the moment)
- Modify actuators orientation + control parameters $\Rightarrow$ possible in the near future ✓(using vectorial thrusters like RMCT)

$$\underbrace{\begin{bmatrix} X \\ Y \\ Z \\ M \\ N \\ K \end{bmatrix}}_{\tau_a} = \underbrace{\begin{bmatrix} \cdots & c\theta_i c\psi_i & \cdots \\ \cdots & c\theta_i s\psi_i & \cdots \\ \cdots & -s\theta_i & \cdots \\ \cdots & -s\theta_i P_{iy} - c\theta_i s\psi_i P_{iz} & \cdots \\ \cdots & c\theta_i c\psi_i P_{iz} + s\theta_i P_{ix} & \cdots \\ \cdots & c\theta_i s\psi_i P_{ix} - c\theta_i c\psi_i P_{iy} & \cdots \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} \vdots \\ u_i \\ \vdots \end{bmatrix}}_{\mathbf{u}_p}$$

We do this using genetic algorithms:

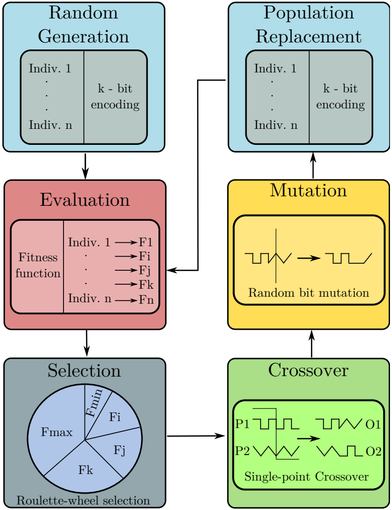
- Evaluation of solutions : minimizing tracking error and energy consumption
- Optimizing design parameters:
  - Thrusters orientation ($\theta_i$, $\psi_i$)
  - Control parameters ($\Lambda$, $K_p$ and tracking point position)




Inspection mission

# Dynamic reconfiguration

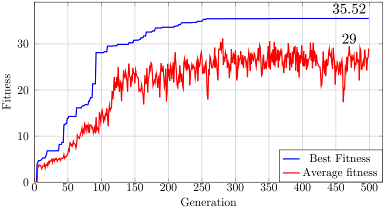
## Genetic algorithm optimization



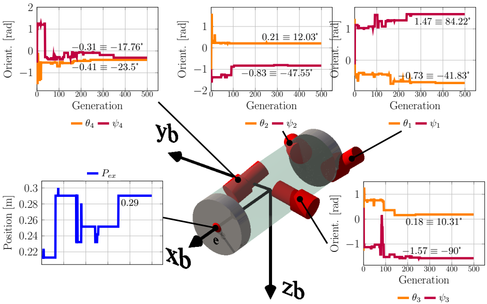

Genetic algorithm

**Table 1**. Evolutionary parameters for dynamic configuration.

| Parameter and symbol | Numerical value |
|---|---|
| Population size, $\mu$ | 40 |
| Number of generations, $g$ | max 500 |
| Number of parameters, $n_p$ | 25 |
| Number of coding bits (thrust. or.), $k_{pos}$ | 5 |
| Number of coding bits (thrust. pos.), $k_{or}$ | 7 |
| Number of coding bits (gains), $k_{gains}$ | 8 |
| Number of coding bits (*e.* pos.), $k_e$ | 5 |
| Size of genotype, $s_{gen}$ | 147 |
| Crossover probability, $pc$ | 0.5 |
| Mutation probability, $pm$ | $5.74 \times 10^{-3}$ |



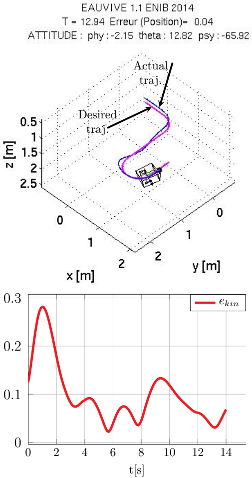

Fitness evolution



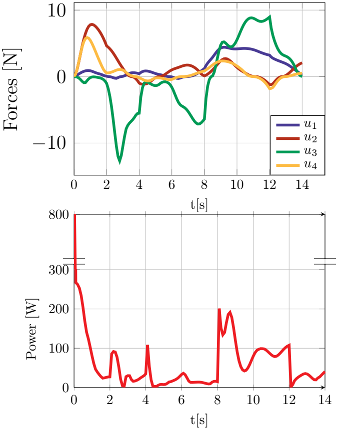

Best propulsive configuration

Final $\Lambda = 1$

Final $K_p = 1.5$



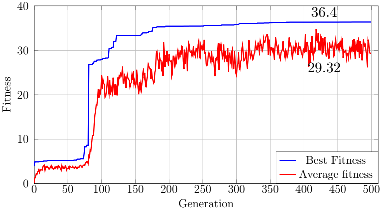

Trajectory tracking



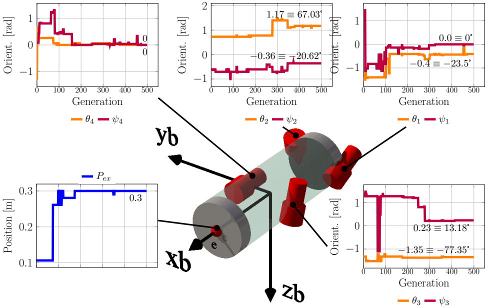

Actuation - power

# Dynamic reconfiguration

## Results - Diving




Fitness evolution

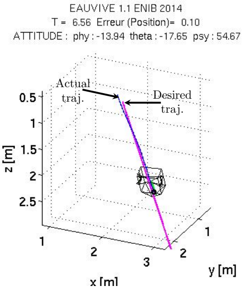


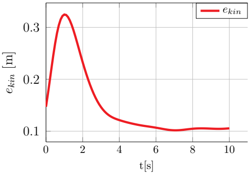

Best propulsive configuration

Final $\Lambda = 0.5$

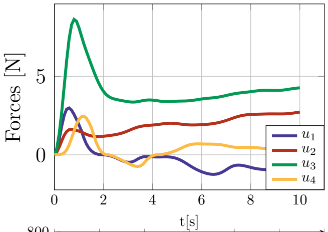
Final $K_p = 2.75$

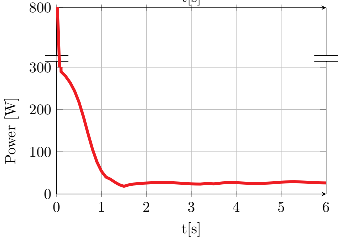
# Dynamic reconfiguration
## Results - Diving




Trajectory tracking

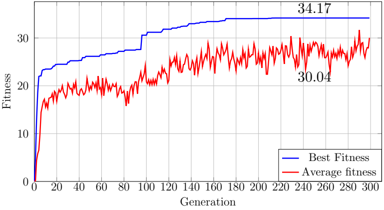


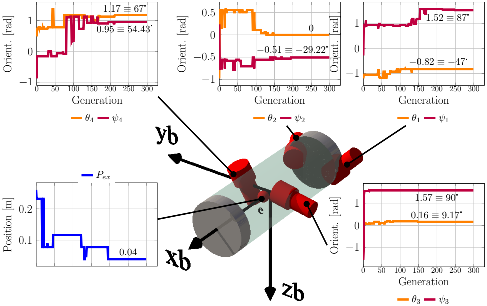

Actuation - power

# Dynamic reconfiguration

## Results - Tomography




Fitness evolution



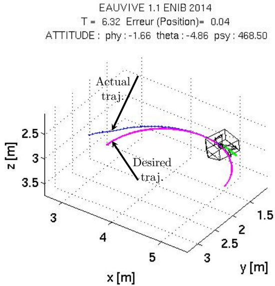

Best propulsive configuration

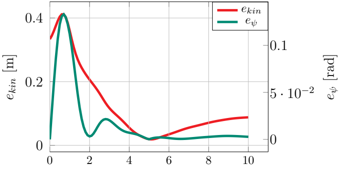
Final $\Lambda = 0.25$

Final $K_p = 9.25$

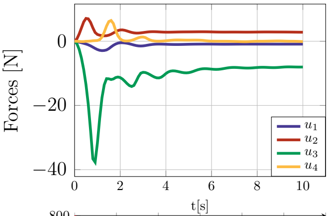
# Dynamic reconfiguration

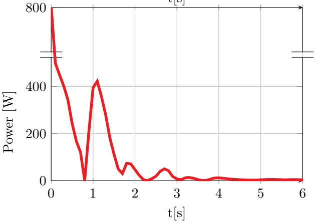
## Results - Tomography




Trajectory tracking




Actuation - power

# Conclusions and perspectives

- Genetic algorithms allow to find an optimal propulsive configuration for a given task (in this work it includes thruster orientation, tracking point position and control gains).
- A robot using this technique could be optimally adapted to a mission including any number of tasks or sub-tasks. For instance seabed scanning could be split into a straight line and sharp turns.
- Current advances in reconfigurable thruster technology can make this technique applicable in the near future (such as RMCT work in IRDL-ENIB).
- The application of the genetic algorithm is susceptible of being improved (use of different GA techniques, improvement in the fitness functions...)
- Other control methods can be included as well in the optimization (robust, adaptive).
- Global optimization can be achieved adding other design parameters.

Thank you for your attention!

Any Questions?