# Localisation and policy synthesis for underwater swarming autonomous vehicles with probabilistic guarantees about safe exploration and reachability requirements

EPSRC Pioneering research and skills

AIMS Autonomous Intelligent Machines & Systems

Ada Alevizaki
Centre for Doctoral Training in Autonomous Intelligent Machines and Systems
University of Oxford
antigoni.alevizaki@eng.ox.ac.uk

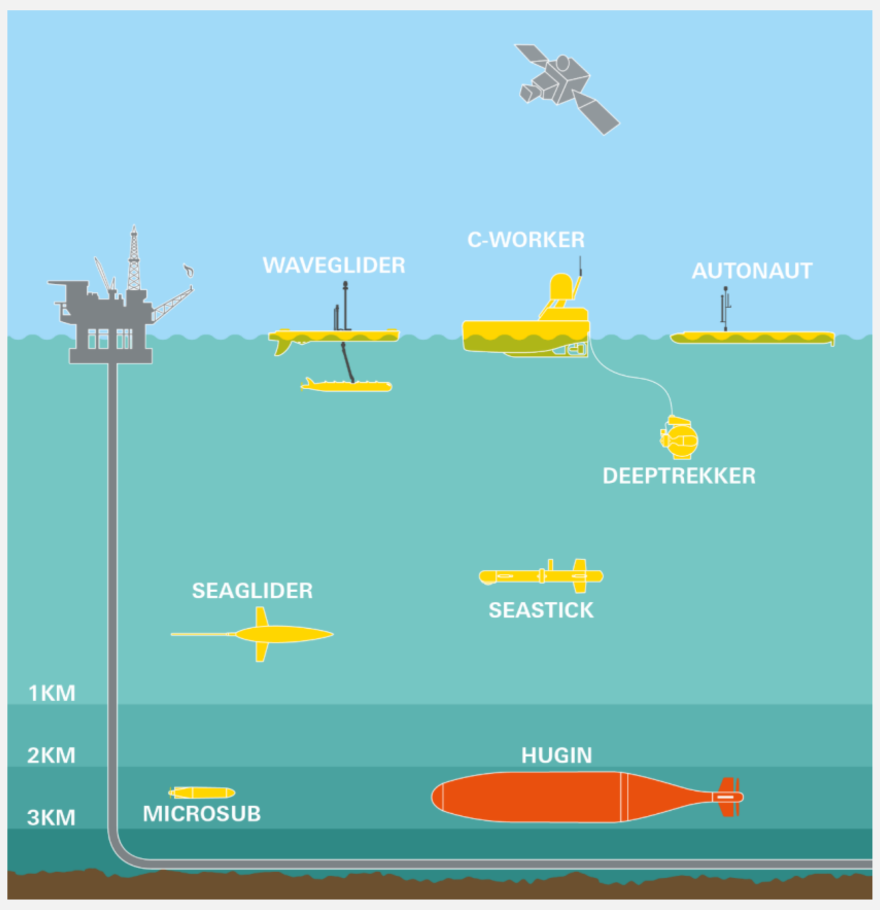
Supervision: Prof. Alessandro Abate, Prof. Niki Trigoni

UNIVERSITY OF OXFORD

## Overview

An AUV submerges into deep ocean to inspect underwater infrastructure.

**Task requires:** Proper decision-making and positional accuracy.




Underwater environment difficulties:
- No GPS.
- Signal attenuation.
- Multipath fading.
- Extreme and unpredictable currents.

**Goal**:
- Keep localisation error under control.
- Synthesise policy to safely explore environment and inspect pipeline.

**Suggested approach**

- 1 AUV (explorer) submerges, 3 AUVs (localisers) stationary on surface in an equilateral triangle formation (GPS enabled).
- Localisation via trilateration after exchange of acoustic links.
- LTL to express properties.
- Q-learning to synthesise policy.
- Observations (sensors, camera) to assign appropriate rewards.

**This differs from classic Q-learning!**

- No luxury to learn from mistakes (expensive equipment).
- Ability to update Q-values of not visited state (observations).
- No need to run multiple incidences to learn - continuous learning!

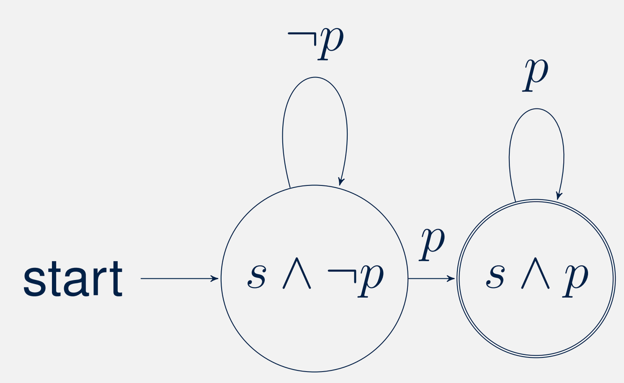
## Method

1. Model **environment** as a MDP $M = (S, \bar{s}, A, P, R)$:
   - $S$ is a set of states,
   - $\bar{s} \in S$ is an initial state,
   - $A$ is a set of actions,
   - $P$ is a partial probabilistic transition function,
   - $R = (R_S, R_A)$ is a reward structure.
2. Define **LTL property** of reachability + safety:

$$\Box s \wedge \Diamond p \wedge \Box(p \rightarrow \Box p). \quad (1)$$

**"ALWAYS keep safe AND eventually find pipe AND once pipe found, ALWAYS follow pipe"**

3. Create **LDBA** from LTL property:



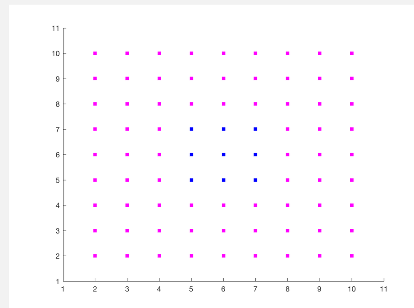

4. Create **product MDP** between $M$ and the automaton.
5. Initialise all voxels with **same** $Q(s, a)$ **values**.
6. Initial position $\bar{s}$ of explorer: **centre of equilateral triangle**, on surface.
7. Explorer submerges vertically until depth $d_{explore}$, **then** starts navigating.
8. **Localisation** using Time of Arrival (ToA) and trilateration:

$$d_i = cT_i, \quad i = \{1, 2, 3\}. \quad (2)$$

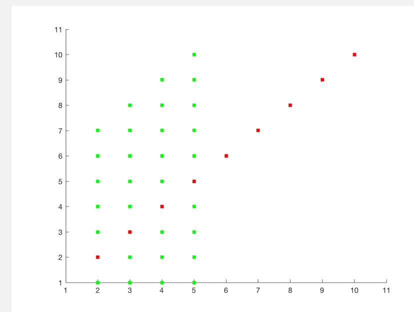
$$(x - x_1)^2 + (y - y_1)^2 + d^2 = d_1^2, \quad (3)$$
$$(x - x_2)^2 + (y - y_2)^2 + d^2 = d_2^2, \quad (4)$$
$$(x - x_3)^2 + (y - y_3)^2 + d^2 = d_3^2. \quad (5)$$

## Method (continued)

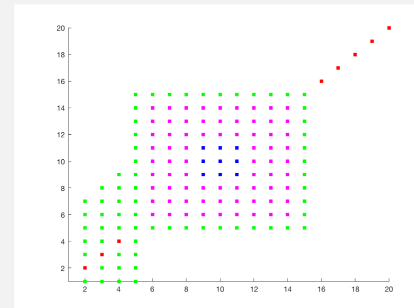
10. Assign **rewards to observed states** $Obs(s) \rightarrow$ prior knowledge!
    - Safe environment: no update.
    - Obstacle avoidance: $\forall s' \in Obs(s) \rightarrow Q(s', a_{dang}) \ll 0$.
    - Pipe following: $\forall s' \in Obs(s) \rightarrow Q(s', a_{pipe}) \gg 0$.
    - Combination $\rightarrow$ keep safe but do not lose pipe.



Obstacle avoidance.



Pipe following.



Combination.

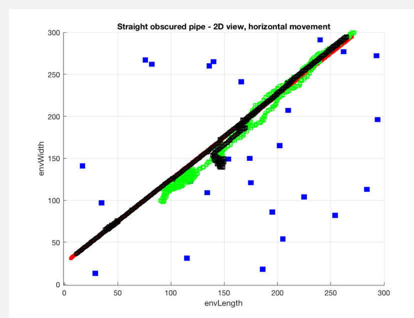
11. **Take action using LCRL** and **update** Q-values accordingly:

$$\begin{cases} Q(s,a) \leftarrow Q(s,a) + \mu \left[ R(s,a) - Q(s,a) + \gamma \max_{a' \in A_s} (Q(s',a')) \right], \\ Q(s'',a'') \leftarrow Q(s'',a''). \end{cases} \quad (6)$$

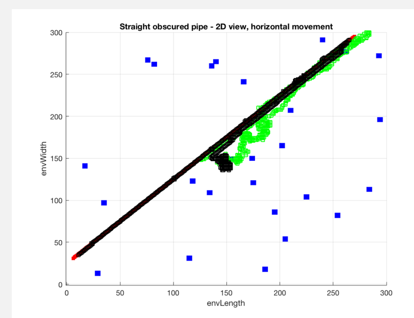
12. Repeat 8-11 until whole pipe inspected.

## Results

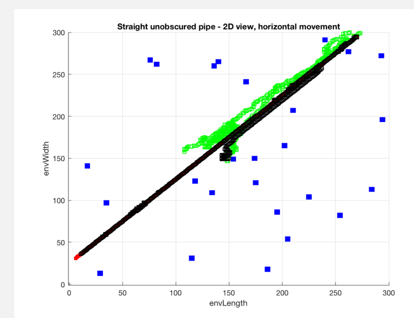
1. Effect of length $l$ of equilateral triangle (case: straight unobscured pipe).



$l = 500$

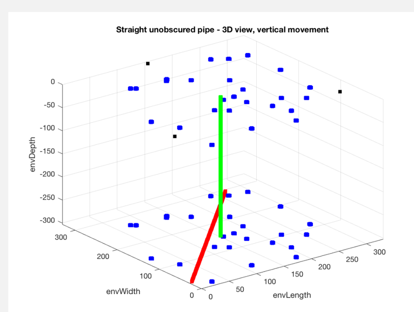


$l = 1000$

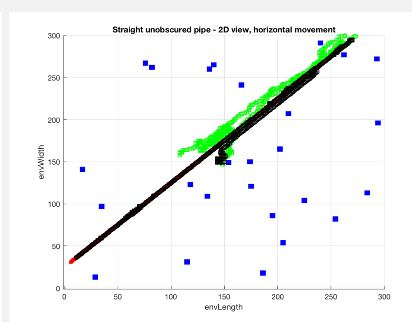


$l = 1500$

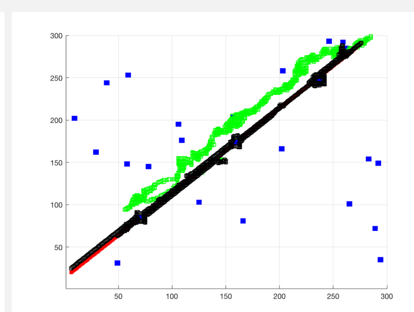
2. Localisation and navigation: test cases.



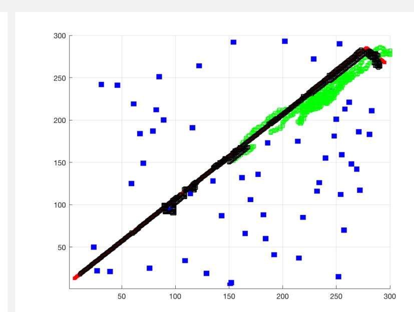
Straight unobscured pipe, vertical movement.



Straight unobscured pipe, horizontal movement.



Straight obscured pipe, horizontal movement.



Straight angular pipe, horizontal movement.

**Localisation very accurate in vertical, but fails in horizontal movement. Synthesised navigation policy very effective under difficult scenarios.**

## Future work

- Sophisticated localisation techniques, e.g. SLAM/Kalman filter.
- Account for energy constraints.
- Less risk-averse localisation and navigation.
- Multiagent set-up.

## Acknowledgements

This project was offered by BP in the context of mini projects carried out by students of the Centre for Doctoral Training in Autonomous Intelligent Machines and Systems at the University of Oxford. We would very much like to thank the BP representatives, Mr. Joe Little, Senior Technology Innovation Manager, Mr. Richard Bailey, Senior Process Engineer and Mr. Peter Collinson, Global Environmental Response Expert, for their time and guidance.

## References

[1] BP. How a new robot fleet is monitoring the underwater world, 2016. [Online; accesed 18-May-2018].
[2] C Baier and Joost P. Katoen. *Principles of Model Checking*. MIT Press, 5 2008.
[3] Mohammadhosein Hasanbeig, Alessandro Abate, and Daniel Kroening. Logically-correct reinforcement learning. *CoRR*, abs/1801.08099, 2018.
[4] Tuna Gurkan and Gungor V. Cagri. A survey on deployment techniques, localization algorithms, and research challenges for underwater acoustic sensor networks. *International Journal of Communication Systems*, 30(17):e3350, 2017. e3350 IJCS-16-0556.R1.
[5] Sarfraz Nawaz, Muzammil Hussain, Simon Watson, Niki Trigoni, and Peter N. Green. An underwater robotic network for monitoring nuclear waste storage pools. In Stephen Hailes, Sabrina Sicari, and George Roussos, editors, *Sensor Systems and Software*, pages 236–255, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
[6] Guangjie Han, Deokjai Choi, and Wontaek Lim. A novel reference node selection algorithm based on trilateration for indoor sensor networks. *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, pages 1003–1008, 2007.