Reinforcement learning: The application to autonomous biomimetic underwater vehicles control

J Magalhães¹, B Damas¹ and V Lobo¹
¹ CINAV, Portuguese Navy Research Center, Almada, Portugal
E-mail: candeias.magalhaes@marinha.pt

Abstract. The Autonomous Biomimetic Vehicles have been increasing in popularity in the past few years. Controlling such type of vehicles is not trivial: due to its complex dynamics and kinematics, it is complex to analytically derive controllers that can efficiently perform a given task, such as reaching a given position target in a minimum time. In this paper we will evaluate the results of the implementation of a reinforcement algorithm in autonomous biomimetic underwater vehicles, providing a new way to control this type of vehicles in which the algorithm is in constant learning.

1. Introduction and motivation

Biomimetics is a field where various principles are applied to mimic biological processes, such as the humanoids robotics field where human-like robots are developed [1]. In the context of underwater vehicles the kinematics and dynamics of the vehicles try to emulate the motion of different sea creatures.

The study of autonomous Biomimetic Underwater Vehicles (BUV) have been increasing in the past few years in a military context due to its furtive locomotion capabilities: they have the ability to camouflage within the environment and a silent undulating propulsion that generates a very different acoustic signature when compared to conventional Unmanned Underwater Vehicles (UUV) [2].

This work is part of the international project "Swarm of Biomimetic Underwater Vehicles for Underwater Intelligence, Surveillance and Reconnaissance (ISR)" (SABUVIS), whose main objective is to use BUVs in missions for stealth data collection and surveillance, where Portugal, Poland and Germany and the countries part of the consortium. Where the Portugal collaborators are Escola Naval, LSTS and OceanScan.

Under this project two different vehicles are being built, one mimicking a fish (Figure 1) and the other replicating a seal. Both vehicles have the same basics characteristics, similar form, two
side fins and a tail, the major differences being their size and tail. The seal-like tail is composed by two smaller tails that mirror the movement of each other, trying to mimic a breaststroke or “frog” stroke style of swimming, while the fish-like tail is a simple tail sectioned in two parts to give more fluidity.

Figure 2. Seal-like vehicle during tests

Under this project the generated acoustic noise developed by the BUVs, as well as their autonomy capabilities, will be measured and compared to conventional UUVs.

These different mechanical structures that are inspired in biological systems present a much more complex kinematic structure that makes the task of controlling the motion of the vehicle a non trivial one. For a fish-like BUV [3], for example, it is not straightforward to develop controllers that actuate the fins and the tail of the vehicle in order to make it follow a desired trajectory in an efficient way [4][5].

Figure 3. Model of the seal-like tail, fish-like tail and fins models, respectively

The purpose of this work is to develop adaptive controllers for these kind of vehicles based on Reinforcement Learning (RL) techniques. Reinforcement learning is an Artificial Intelligence technique, also biologically inspired, that revolves around “trial and error” theory. In RL an agent is supposed to choose its actions in a way to maximize an external reward that it gets from its interaction with the environment: a positive reward is given when it fulfills certain conditions defined by the programmer, and a lower reward is obtained when that condition is not met. This reward is tightly linked to the desired behaviour for the agent and should be chosen carefully as the agent, or robot, will solely learn based on the rewards it gets. For instance, when learning to navigate from a position to other the agent should get higher value rewards as it gets closer to the desired position.

To learn to perform a given task the agent has an internal policy that maps states to actions. Given the history of visited states, associated performed actions and received rewards the agent continuously adapts its policy in order to maximize its expected accumulated reward in the long run, this way progressively starting to exploit the actions that get the higher positive rewards, until it reaches the best way to perform the task.
There is no need to worry about programming every action to perform in every possible state when using a RL scheme, as these actions are learned from trial and error; it is even possible to the algorithm to adapt to unexpected situations, because the agent will teach itself in a continuous process of adaptation. This is a major advantage of using RL schemes when compared to traditional controllers, where a good knowledge of the process to control is typically needed.

2. Reinforcement learning in underwater biomimetic vehicles control

To implement a RL controller in the BUVs the work will be divided in two different stages. In the first stage a simulator will be developed and the RL algorithms will be tested and applied in a simulated environment. As the BUV will learn how to control its movements from scratch, the learning phase will start in such a simulated environment to prevent actuator wear, collisions and other damage to the physical robot that would inevitable result from the initial exploratory random movements it would perform in this phase. Afterwards, when the simulated robot can already control its movement in a satisfactory way, the learned model will be transferred to the real vehicle, where it will undergo a second learning phase. Such adaptation is required to take into account the differences between the simulated and the real environment.

2.1. BUV Q-learning simulation

Q-Learning is a Reinforcement Learning algorithm that estimates, from its interaction with the environment, the utility of performing a given action in a particular state, given by \(Q(S, A)\) [6]. This policy can be learned from this interaction, without the need to have a model for this environment, using the update rule.

\[
Q(S,A) = Q(S,A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S,A)],
\]

where \(Q\) denotes the expected accumulated future reward obtained if action \(A\) is performed in state \(S\). \(S'\) is the observed next state after executing action \(A\) and \(R\) the corresponding reward. \(0 < \alpha \leq 1\) is a step-size parameter and \(0 < \gamma \leq 1\) denotes the discount-rate factor: the lower this value the lesser the importance given to distant future rewards, i.e, the more myopic the agent is regarding future rewards.

The vehicle in training is a fish-like composed by two lateral fins, one on each side, and one tail. Each motor is actuated by a sinusoidal signal with variable frequency (F) [mHz] and deflection (K), the average value in which the tail or fins cycle around, where \(F\) and \(K\) denote the tail variables, \(F1\) and \(K1\) the left fin and \(F2\) and \(K2\) for right fin. Together these 6 controlled variables define the action vector [7].

The tabular nature of the vanilla Q-Learning algorithm requires a discretized set of actions and states. With respect to actions we define \(A\) as the variables whose columns contain the discretized action values for each component of the action vector

\[
A = \begin{bmatrix}
3000 & 500 & 3000 & 500 & 3000 & 500 \\
2250 & 250 & 2250 & 250 & 2250 & 250 \\
1500 & 0 & 1500 & 0 & 1500 & 0 \\
750 & -250 & 750 & -250 & 750 & -250 \\
0 & -500 & 0 & -500 & 0 & -500
\end{bmatrix},
\]
where each column represents $F$, $K$, $F_1$, $K_1$, $F_2$ and $K_2$ respectively and where we set $F = F_1 = F_2$, in steps of 750, and $K = K_1 = K_2$, in steps of 250. This way, we have a set of $5^6$ different actions from where to choose at each iteration.

We consider the state to be the position in each axis, in meters, and the vehicle angle in the xy plane (in radians). The state variables $S$, that contain that discrete values, is then defined as

$$
S = \begin{bmatrix}
1 & 1 & 1 & 0.6 \\
2 & 2 & 2 & 1.2 \\
\vdots & \vdots & \vdots & \vdots \\
9 & 9 & 9 & 5.4 \\
10 & 10 & 10 & 6
\end{bmatrix},
$$

where each column contains the possible values for $x$, $y$, $z$, in steps of 1, and $\psi$, in steps of 0.6, respectively. This makes $10^4$ distinct states.

The objective of this simulation is to reach a goal point and stay there. To fulfill it, the reward function is defined as

$$
R = -\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2},
$$

so that the reward will be higher the closest it gets to the goal.

The reward had to be made negative due to $Q$, that is initialized with zeros, selecting the action with the maximum value to the present and future state. By making it negative, it is guaranteed that each action is selected at least one time for each state, in the infinite number of episodes.

The simulation is divided by episodes. Each episode starts the vehicle in a random position with a 4 meter distance to goal and with the vehicle pointed at it, each episode it has 200 iterations, each one representing 0.1 seconds, and it end when 20 seconds have passed, starting a new episode.

Figure 4. Graphics of early results of the Q-learning algorithm
The graphics above show one of the trajectories of the vehicle, in a 3D dimension and 2D x0y axis, its altitude during the episode and its accumulative reward in the past episodes. The red star being the starting position and the green star the goal.

It is noted that the accumulative reward has already started to converge, which means that is starting to reach a satisfactory solution, but it is taking a long time to do it.

2.2. LSTS
The Laboratório de Sistemas e Tecnologia Subaquática (LSTS) is an interdisciplinary research laboratory part of Faculdade de Engenharia da Universidade do Porto (FEUP) that specializes on the design, construction and operation of unmanned underwater, surface and air vehicles and on the development of tools and technologies for the deployment of networked vehicle systems.

LSTS created the software toolchain Neptus-IMC-DUNE that will be present in the project SABUVIS. DUNE is the system for vehicle on-board software, it has a C++ programming environment and it is responsible for navigation, code for control and access to sensors and actuators; IMC is the communication protocol; Neptus is a command, control, communications and intelligence framework for operations with vehicles, systems and human operators.

![Figure 5. LSTS toolchain, Neptus-IMC-DUNE](image)

In the project SABUVIS, LSTS is responsible for the integration of software and sensors on the vehicle.

3. Conclusions and future work
The convergence of the Q-learning algorithm shows us that we are headed in the right direction with these results, we can already see that the simulation is trying to reach the goal as soon as possible, also it is already starting to learn to turn back and try to return to the goal to obtain a better reward. There are still some problems like needing an adjust in the state variables to decrease the convergence time and the main problem being the discretization of the values. A different approach may resort to function approximation or algorithms based on policy improvement \( \text{e.g., } PI^2 \) or POWER, in order to circumvent the discretization of the state and action space. We also expect to implement the algorithm in the physical BUV, using the software toolchain that runs on the vehicle developed by FEUP.
References

[1] BAR-COHEN, Yoseph, "Biomometrics: Biological Inspired Technologies", California, Jet Propulsion Laboratory (JPL), California Institute of Technology, 2006

[2] LISTEWNIK, K., "Sound Silencing Problem of Underwater Vehicles", in Solid State Phenomena, Vol. 196, Trans Tech Publications, 2013, pp. 212-219

[3] LOW, K. H., "Modelling and parametric study of modular undulating fin rays for fish robots", in Mechanism and Machine Theory, Vol. 44, 2009, pp. 615–632

[4] COLGATE, J. E. and LYNCH, K. M., "Mechanics and Control of Swimming: A Review", in IEEE Journal of Oceanic Engineering, Vol. 29, No. 3, 2004, pp. 660-673

[5] MALEC, M. et al., "Analysis of Parameters of Traveling Wave Impact on the Speed of Biomimetic Underwater Vehicle", in Solid State Phenomena, Vol. 210, Trans Tech Publications, 2014, pp. 273-279

[6] BARTO, Andrew G. and SUTTON, Richard S., Reinforcement Learning: An introduction, 1st edition, Cambridge, MIT, 1998

[7] SZYMAK Piotr, "Mathematical model of underwater vehicle with undulating propulsion", Poland, Polish Naval Academy, Institute of Electrical Engineering and Automation