Reinforcement Learning

Application to Underwater Acoustics

March 27, 2023

Thomas Aussaguès
Lucas Fourest
Selman Sezgin
Encadré par François-Xavier Socheleau
IMT Atlantique

Outline

Outline

Introduction to Underwater Acoustics (UWA)

The Reinforcement Learning Framework

Results & Discussion
- Scenario 1 & 2: Focusing on the SNR
- Scenario 3: Including power into the reward

Conclusion

The Underwater Acoustic Environment

- UW sensors are used in many applications: marine flore monitoring, anti-submarine warfare, fishing, seafloor exploration...
- Many difficulties:
  - Low **bitrates**: ≈ 10kbit/s/km
  - Transmitters and receivers are **unacessible**
  - **Perturbations**: attenuation, multipath (see figure 2), DOPPLER...

Figure 1: Nuclear submarine [1]

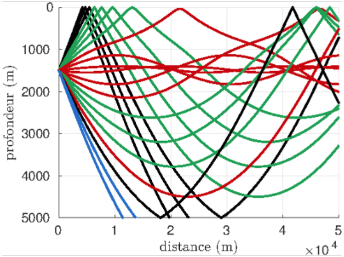

Figure 2: Multipath

The channel

- Figure 3: the channel impulse response $h(\tau)$ measured over the time t
- Gives details about the channel behavior
- No clear principal and secondary paths
- Time spreading

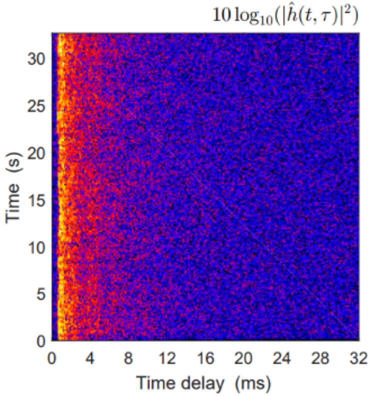

Figure 3: Channel impulse response over time (recorded in Norway)

Application to Underwater Sensors

- Objective: design a self **adaptive** transmitter...
- ...that can adapt itself to a time-varying channel to **preserve** the data link and **optimize** both **bitrate** and **resources** consumption
- **"Adapt"** : chooses a modulation among a set of available modulations

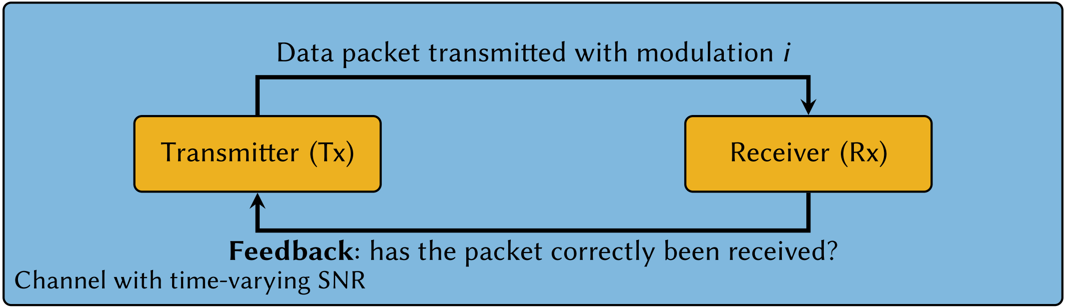

Figure 4: Transmitter and receiver

Outline

Introduction to Underwater Acoustics (UWA)

The Reinforcement Learning Framework

Results & Discussion

- Scenario 1 & 2: Focusing on the SNR
- Scenario 3: Including power into the reward

Conclusion

The principle [2, 3]

- Reinforcement Learning (**RL**) is a machine learning technique that involves training an agent to **make decisions** by **interacting** with an environment
- Receives **rewards or punishments** based on its actions $\longrightarrow$ learns to make better decisions to maximize an overall reward
- The process involves **several steps**:
  - 1. The agent observes the state of the environment and **chooses an action**
  - 2. It **interacts** with the environment and receives a **reward or punishment**
  - 3. The action strategy (*e.g* policy) is **adjusted** accordingly
- By using RL, an agent can learn to solve complex problems by **exploring** and **adjusting** its behavior based on the results obtained

The principle [2, 3]

- At **each** time-step:
- 1. **States** $(s_k)_{1 \leq k \leq K}$ could be taken
- 2. **Actions** $(a_n)_{1 \leq n \leq N}$ could be chosen
- 3. An **action-value** function $Q(a_n)$ (eventually $Q(s_k, a_n)$) is **updated**
- 4. A **policy** $\pi(s_t)$ guides action-taking *NB : can also be Markovian* $\pi(a|s_t)$

$\sum$ $Q(a_n)$ is an **estimate** of $\mathbb{E}[R_t(a_n)]$

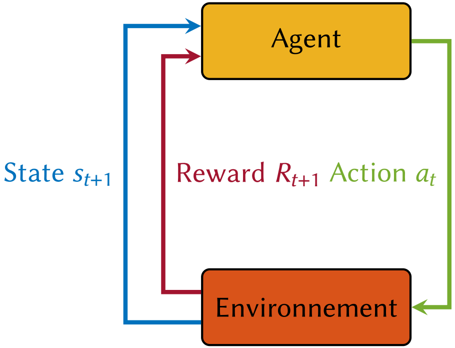

Figure 5: RL process

Q-learning [2, 3]

- Example of **a simple way** to update Q at time t
  - Choosing a_t leads to a reward R_t[1]
  - Update rule: Exponential Moving Average

$$
\begin{aligned}
Q_{t+1}(a_t) &= \overbrace{(1-\alpha)\,Q_t(a_t)}^{\text{Present \& past values}} + \overbrace{\alpha R_t}^{\text{Update}} \\
&= Q_t(a_t) + \alpha\,(R_t - Q_t(a_t))
\end{aligned}
$$

Σ Q might always be **incomplete** knowledge: π should ensure to keep a part of **exploration** in the process and not to always take the optimal action known so far: balance between **exploration** and **exploitation**

[1] The reward depends implicitly on the chosen action: $R_t = R_t(a_t)$

Policy implementation [2, 3]

- Policy π: How to **choose an action** at time t?
  - ϵ-greedy method

$$a_t = \begin{cases} \arg\max_a Q_t(a) \text{ with probability } (1-\epsilon) \\ \text{random action with probability } \epsilon \end{cases}$$

  - Upper Confidence Band (UCB) action selection

$$a_t = \overbrace{\arg\max_a Q_t(a)}^{\text{Greedy}} + \underbrace{c\sqrt{\frac{\ln t}{N_t(a)}}}_{\substack{\text{Measure of the uncertainty} \\ \text{in the estimate}}}$$

$N_t(a)$ being the number of times a has been explored at time t

Application to our problem [2, 3]

- In our case, we can model the **transmitters** as the **agents**
- The **actions** are the **modulation choice** to transmit the packets at each time-step
- Here, we do not consider several potential states for the transmitters (a single state)
- Depending on the modulation choice, the **bitrate** as well as the **transmission error probability** (Packet Error Rate) will be impacted
- The goal will be to **deploy RL** algorithms to find the **best modulations choices** in an **adaptative** way

Outline

Channel configuration

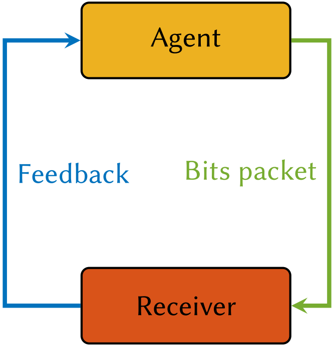

Figure 6: Communication between the receiver and the transmitter

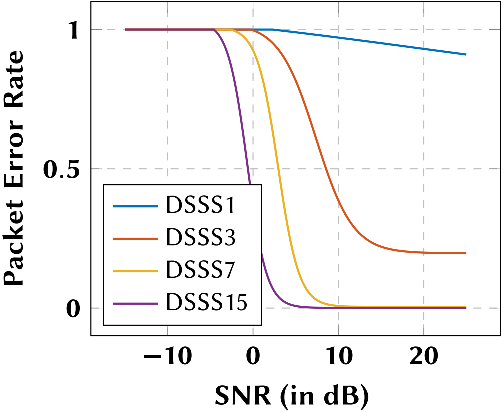

Figure 7: Packet Error Rate (PER) as a function of the SNR

Scenario 1

- Constant SNR
- No propgation delay
- Reward function:

$$R_t(a) = \begin{cases} D_a & \text{if the packet is transmitted, with a probability equal to } 1 - \mathrm{PER}(a) \\ 0 & \text{with a probability equal to } \mathrm{PER}(a) \end{cases}$$

where D_a is the bitrate related to modulation $a \in \{\mathrm{DSSS1}, \mathrm{DSSS3}, \mathrm{DSSS7}, \mathrm{DSSS15}\}$

Average cumulative reward

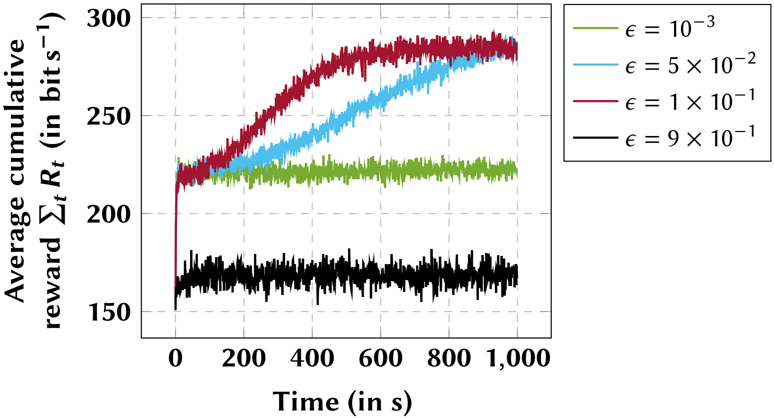

Figure 8: Average cumulative reward over 2000 independent agents

Scenario 2

- Variable SNR : $\mathrm{SNR}(t) = \mathrm{SNR}_0 + g(t)$ where

$$\begin{cases} g(t) = \phi g(t-1) + \varepsilon(t) \quad \text{(AR(1) model)} \\ \varepsilon(t) \sim \mathcal{N}(0, \sigma^2) \end{cases}$$

  with SNR_0 the mean SNR and $\phi > 0$ the channel coherence time.

- Still no propagation delay
- Same reward function as before

Fluctuations of the Packer Error Rates

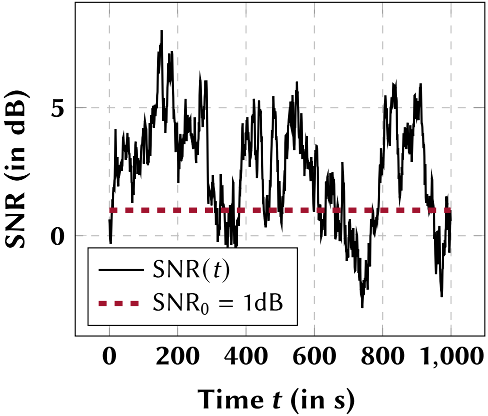

Figure 9: SNR trajectory ($\text{SNR}_0 = 1\text{dB}$)

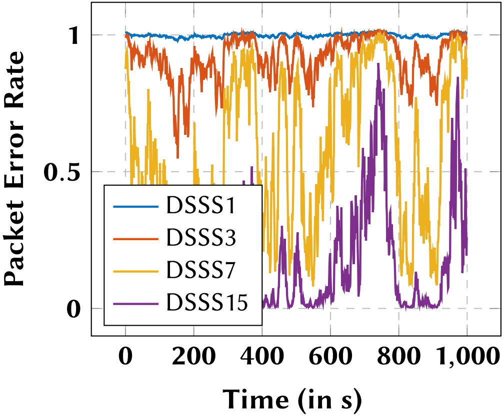

Figure 10: Packer Error Rate as a function of the time

Average reward

- Optimal value for α : $\approx 10^{-2}$ or $\approx 10^{-1}$
- If α is too low, the agent can not track the SNR fluctuations
- ϵ-greedy methods perform better than the UCB

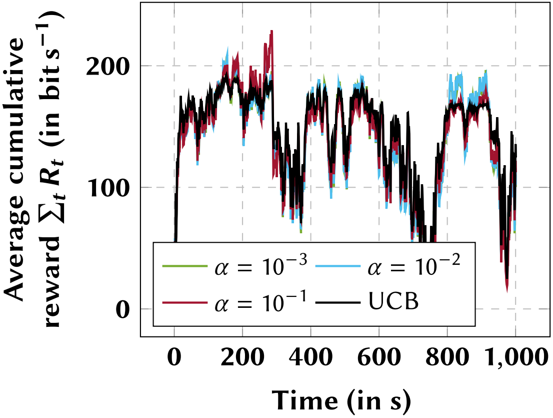

Figure 11: Average reward for different values of learning rate α

Proportions of the chosen modulations

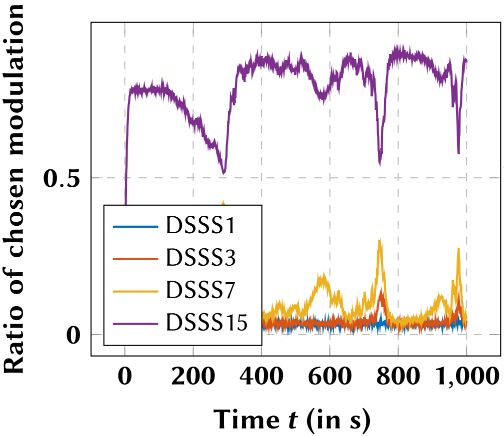

Figure 12: Modulations chosen among 2000 agents

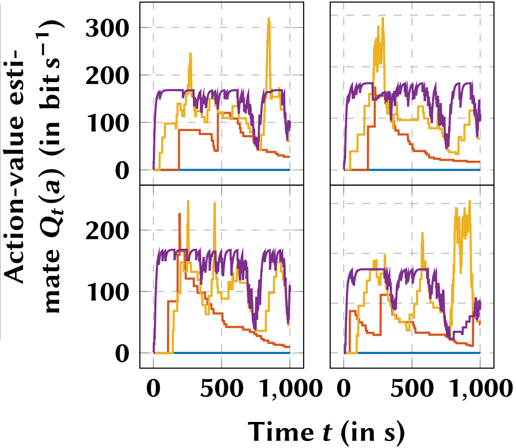

Figure 13: Estimated action values of different agents

Outline

Introduction to Underwater Acoustics (UWA)

The Reinforcement Learning Framework

Results & Discussion

- Scenario 1 & 2: Focusing on the SNR
- Scenario 3: Including power into the reward

Conclusion

Designing new modulations

- In the foregoing slides: only optimization of the bitrate, regardless of the power consumption...
- How to take it into account?
- Let's add new modulations!
- From DSSS7, two additional modulations are added by translating the PER curve by ±3dB
- Necessity of a new reward

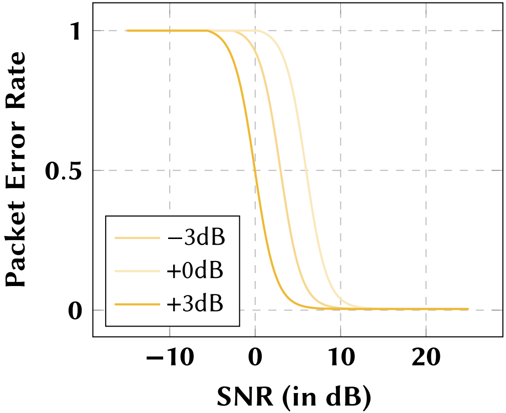

Figure 14: PER curves for the new DSSS7 modulations

Designing a new reward

- The reward should be designed such that it takes both bitrate and power into account
- Idea: compute the bitrate value per energy unit ⇔ "How many bits can we send with one J?"

$$\overbrace{\mathcal{E}_a}^{\text{Energy required to send 1 bit}} = \frac{\overbrace{P_a \times T}^{\text{Energy}}}{\underbrace{D_a \times T}_{\text{Number of bits}}} = \frac{P_a}{D_a}$$

$$\underbrace{R_t(a)}_{\text{Reward obtained by choosing action } a} = \begin{cases} \frac{D_a}{\mathcal{E}_a} = \frac{D_a^2}{P_a} & \text{if the packet is transmitted} \\ -\frac{D_a^2}{P_a} & \text{else} \end{cases}$$

Conclusion

Thank you for your attention

References

[1] Wikipédia, “Classe le triomphant — wikipédia, l’encyclopédie libre,” 2022. [Online; accessed 21-March-2023].

[2] R. Sutton and A. Barto, *Reinforcement Learning, second edition: An Introduction.* Adaptive Computation and Machine Learning series, MIT Press, 2018.

[3] A. Pottier, F.-X. Socheleau, and C. Laot, “Quality-of-Service Satisfaction Games for Noncooperative Underwater Acoustic Communications,” *IEEE Access*, p. ., Apr. 2018.