Ship Course-keeping in Waves using Sample-efficient Reinforcement Learning

Keeping a ship on course during its voyage in seas and oceans is important to make the journey as quick and efficient as possible. Though conventional controllers are often seen as the popular solution to this problem, reinforcement learning is emerging as a smart alternative with the recent rise of machine learning. In contrast to the circumstantial tuning of conventional controllers, reinforcement learning learns the dynamics of the ocean and its waves, allowing it to be applied to any sea state without further intervention.

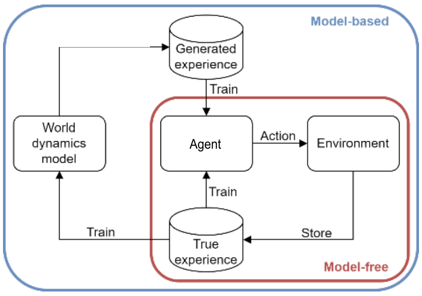

Figure 2: The relation between model-free RL and model-based RL.

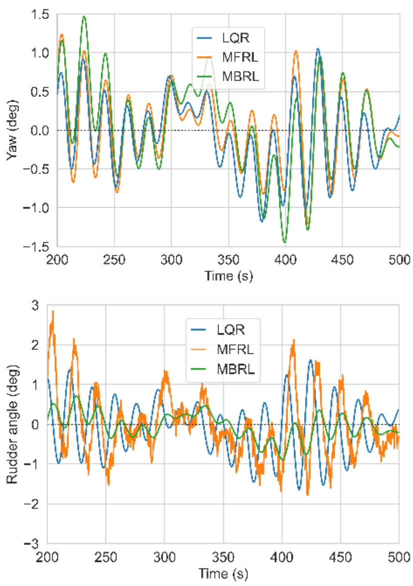

Figure 4: The yaw and rudder angle of the ship between 200s and 500s of a simulation.

Reinforcement learning (RL) has two main components: the agent and the environment. The agent takes actions based on its observation of the environment. The environment changes according to its dynamics and the action taken by the agent. The agent receives feedback in the form of a reward by which it learns which actions to take when the environment is observed to be in a certain state (Figure 1). By maximising the cumulative reward, the agent can discover optimal strategies in complex, dynamic environments.

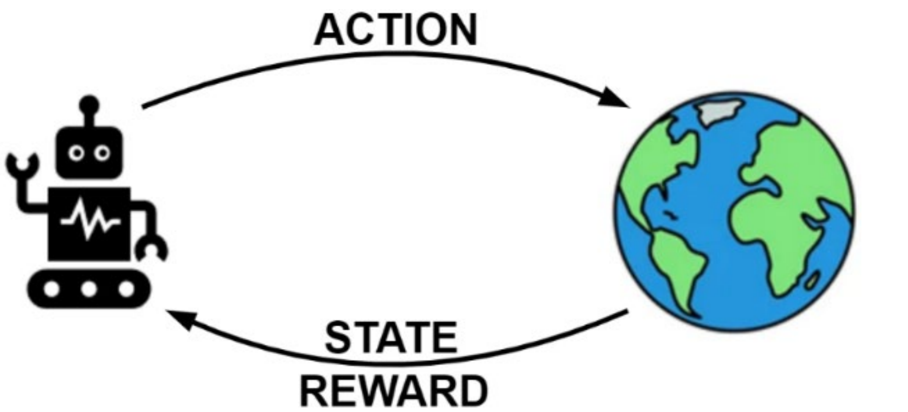

Figure 1: The agent-environment interaction cycle in reinforcement learning.

The aim of this project was to develop a RL agent that can learn to keep a ship on course in waves. These waves in te environment provide an uncertainty in the future state after an action is performed. Hence, the agent does not only have to learn which action to take in different situations, but also how different the environment might be afterwards. To achieve this, the agent was trained in multiple simulations with differing significant wave heights, wave periods, and wave direction (β).

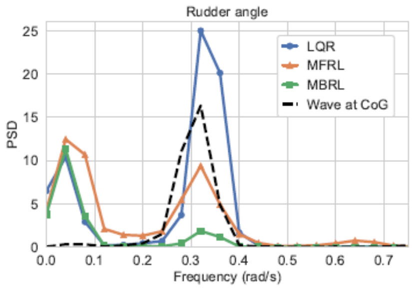

Figure 5: The power spectral density (PSD) of the rudder motion of each controller. The dashed line indicates the PSD of the encountered wave at the ship's centre of gravity (COG).

Future research

While this project has been concluded, more questions and possible research directions have emerged. How would different RL methods, such as offline or hierarchical RL, change the behaviour of the agent? How can the learned dynamics model be improved? What if conventional controllers were to be combined with RL? Questions that we aim to answer at MARIN in the future.

For more information contact MARIN:
Justin Greep
T + 31 317 49 36 06
E j.greep@marin.nl

Additionally, we investigated the speed at which the agent is trained. Model-free RL (MFRL) is straightforward to implement and train but it can require many simulations for learning a control strategy. MFRL learns a strategy by directly interacting with an environment; it does not require the model of the environment nor makes it an attempt to learn the model. On the other hand, Model-based RL (MBRL) learns the model of the environment to predict the future states. MBRL is more complex to implement and train but it is more sample-efficient as it can leverage the learned model to generate hypothetical experiences, reducing the number of actual interactions with the environment needed. Figure 2 gives an overview of the relation between MFRL and MBRL.

We conducted a large number of simulations with different sea states to evaluate the performance of the RL agents. The yaw error, i.e. how far off-course the ship is, and the rudder usage were used as criteria for the agent's performance and compared to those of a linear-quadratic regulator (LQR). Figure 3 gives an overview of the average performance and Figure 4 shows a fragment of the agent's behaviour.

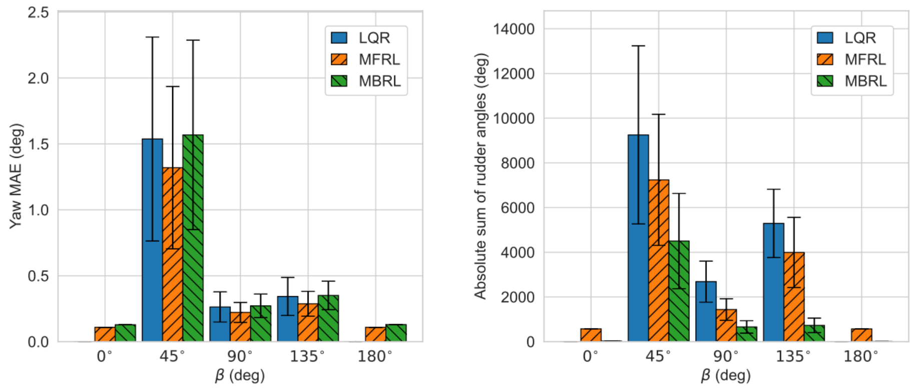

Figure 3: The mean absolute yaw error (left) and the total rudder usage (right) of the LQR and RL agents. For each wave direction (β), the performance of the three controllers is averaged over multiple sea states with varying significant wave heights and peak periods.

The MBRL agent has reduced its rudder usage significantly without hurting its ability to stay on course. This can be attributed to the fact that the MBRL agent learned to ignore the first-order wave disturbances and to focus only on the low-frequency yaw deviations (see Figure 5).

Using the MBRL agent, the required training time of the agent was reduced from nearly 540K steps to just 11K steps compared to the model-free counterpart. This is a significant outcome as it makes applying RL in real-life applications more feasible.

To conclude, this project has shown that RL can be a competent controller for ship course-keeping in waves. A particular bottleneck of training RL agents - the large number of agent-environment interactions - was mitigated by employing a model-based RL agent, which learns a smart control strategy in an efficient manner.