Maamoun *et al*. *Complex Eng Syst* 2022;2:13
DOI: 10.20517/ces.2022.28

Complex Engineering Systems

Research Article

Open Access

Reinforcement learning-based control for offshore crane load-landing operations

Khaled Said Ahmed Maamoun[1], Hamid Reza Karimi[2]

[1]Department of Automation and Control, Politecnico di Milano, Milan 20133, Italy.
[2]Department of Mechanical Engineering, Politecnico di Milano, Milan 20156, Italy.

Correspondence to: Khaled Said Ahmed Maamoun, Automation and control, Politecnico di Milano, 32, Piazza Leonardo Da Vinci, Milan 20133, Italy. E-mail: khaledsaid.maamoun@mail.polimi.it;

How to cite this article: Maamoun KSA, Karimi HR. Reinforcement learning-based control for offshore crane load-landing operations. *Complex Eng Syst* 2022;2:13. http://dx.doi.org/10.20517/ces.2022.28

Received: 23 Jul 2022 **First Decision:** 16 Aug 2022 **Revised:** 23 Aug 2022 **Accepted:** 30 Aug 2022 **Published:** 31 Aug 2022

Academic Editor: Chang-Hua Lien **Copy Editor:** Fanglin Lan **Production Editor:** Fanglin Lan

Abstract

Offshore crane operations are frequently carried out under adverse weather conditions. While offshore cranes attempt to finish the load-landing or -lifting operation, the impact between the loads and the vessels is critical, as it can cause serious injuries and extensive damage. Multiple offshore crane operations, including load-landing operations, have used reinforcement learning (RL) to control their activities. In this paper, the Q-learning algorithm is used to develop optimal control sequences for the offshore crane's actuators to minimize the impact velocity between the crane's load and the moving vessel. To expand the RL environment, a mathematical model is constructed for the dynamical analysis utilizing the Denavit–Hartenberg (DH) technique and the Lagrange approach. The Double Q-learning algorithm is used to locate the common bias in Q-learning algorithms. The average return feature was studied to assess the performance of the Q-learning algorithm. Furthermore, the trained control sequence was tested on a separate sample of episodes, and the hypothesis that, unlike supervised learning, reinforcement learning cannot have a global optimal control sequence but only a local one, was confirmed in this application domain.

Keywords: Marine operation, offshore crane, Q-learning algorithm, reinforcement learning

1. INTRODUCTION

Setting a heavy object down on the deck of a vessel is one of the most common marine operations. During the load-landing or -lifting process, any disturbance such as heave motion may lead to a significant impact on the load and equipment, which may cause fatal injury to the crew and permanent damage. Due to a variety of factors, such as ship motions, crane mechanics, and other factors, achieving a soft load landing with acceptable impact force and a small distance is challenging. Hence, a lot of efforts have been carried out to facilitate this operation; some introduced a variety of control algorithms to automate the process, and others intended to provide training methodology for the operators through simulators, virtual reality, and augmented reality. Several control techniques for marine operations have been proposed. A payload position control of offshore crane was developed by Park *et al.*[1] using uniformly ultimately bounded (UUB) theory, integrated with the input–output linearization control technique (IOLC). A passive heave compensation system for the offshore landing process was studied by Huster *et al.*[2] and Ni *et al.*[3]. The passive shock isolator has some limitations due to the higher weights of loads and the hard working environment; hence, Zhu *et al.*[4] presented a feedback control strategy where a shock isolator with an optimal controller is introduced to reduce and minimize the peak force transferred to the load. The commonality of the previous techniques is that they are model-based control algorithms. Hence, different dynamical models are presented in the literature. Mackojc *et al.*[5] presented a six-degree of freedom (6DOF) mathematical model for the vessel and a 3DOF model for the lifting system, and they combined them to produce a payload–vessel system to facilitate the comprehensive investigation of mutual interactions. Idres *et al.*[6] and Ellermann *et al.*[7] developed a nonlinear coupled model for a crane and cargo based on the assumption that the cargo was a point mass. Cha *et al.*[8] established the coupled model between the floating crane and the cargo considering the geometries of cargo. Spong's book[9] and Williams *et al.*[10] built a robot-like model for the knuckle crane using DH notation and Lagrange's approach. Another approach is using the reinforcement learning (RL) theory. RL is a learning method that gradually explores the optimal policy by interacting with the environment[11]. Andersson *et al.*[12] proposed actuator space control policies for a forestry crane manipulator to be energy efficient in log grasping by involving a simple curriculum in a deep reinforcement learning setup. Moreover, Gaudet *et al.*[13] introduced a new integrated guidance and control method for a spaceship based on reinforcement learning concepts. They used the control algorithm to learn a policy that directly maps the lander's estimated state to a commanded thrust for each engine, resulting in precise and fuel-efficient trajectories. Sun and Xie[14] introduced a backstepping control scheme based on reinforcement fuzzy Q-learning to control container cranes. Ding[15] built different environments and used deep neural networks with RL algorithms to set down the load softly. Based on the aforementioned results, in this work, we focus on providing optimal control sequences for an offshore crane to get acceptable impact velocity while landing its load on a moving vessel using reinforcement learning algorithms, where a mathematical model of the offshore crane was established to be involved in the reinforcement learning environment, Q-learning algorithm was created using MATLAB platform to control the crane actuators, and the algorithm performance was measured against the variation of its hyperparameters. Moreover, the bias that usually occurs in Q-learning algorithms was tested using Double Q-learning algorithm. Double Q-learning is an off-policy value-based reinforcement learning algorithm with no positive bias in estimating the action values in stochastic environments.

The remainder of this paper is organized as follows. In Section 2, the problem statement with the operation system assumptions are described in detail, and the algorithms' structures are provided as well. In Section 3, a description of the RL algorithms that are used during this work is given. In Section 4, a simulation for the control sequences is demonstrated, and a comparison between the behavior of the environment under different assumptions is introduced. Section 5 is a discussion of the obtained results, and it stands on the strength of this work.

2. PROBLEM DESCRIPTION

2.1. Environment

In this work, two different environments are considered, namely initial environment and upgraded environment, as described in the following.

2.1.1. Initial environment

In this environment, the wave of every episode is randomly generated from the JONSWAP spectra created by the same sea state. Therefore, the wave elevation differs between episodes but not the statistical properties [16]. Here, the sea state is assumed to be:

$$H_s = 1.5m$$

$$T_p = 30s$$

where H_s is the wave height and T_p is the peak wave period. In a real application, the vertical motion of a vessel can be predicted approximately 4 s ahead of time with high accuracy[17]. This gives us the reliability to plane a control sequence that can be visualized to the crane operator using technologies such as augmented reality. Moreover, the following assumptions are imposed on the load-landing operation problem:

Assumption 1

I Neglect all the crane and load dynamics.
II The mass and stiffness of the barge are neglected.
III The barge has the same amplitude of the wave (barge dynamics is neglected and its response is 1 to all the wave frequency).
IV Neglect the wave effect on the crane.
V Consider the hoist speed as the action to the control input directly.

State observation

In this environment, the agent–environment interaction is described through the following set of equations:

$$P_h(i+1) = P_h(i) + a_t \cdot t_{step} \tag{1}$$

$$D_r(i+1) = D_r(i) - |P_w| \tag{2}$$

where P_h is the hoist position; P_w is the wave height; D_r is the relative distance between P_h and P_w; a_t is the chosen action (hoist velocity); and t_{step} is the time step of discretization. The action space in this environment consists of 11 actions $[(-5/30) : (5/30)]$ with step (1 / 30) m/s , which are the hoist velocities[15].

2.1.2. Upgraded environment

The environment is upgraded by inserting the forward kinematic model of a knuckle crane. Considering the crane as a three-revolute joints robotics arm, the input to the environment, in this case, is the actuators' angles.

State observation (forward kinematic)

The mathematical model for the knuckle crane was generated by simulating the crane as a robotic arm with three joins, as shown in Figure 1, with the following assumptions:

Assumption 2

I Neglect the dynamics of the load and wires.
II The second and third joints are actuated through hydraulic cylinders, but the actuator models are not taken into account.
III The wave amplitude changes with time but not with the x-direction of the global reference; in other words, the crane end-effector is assumed to be interacting with the same wave amplitude in all of its x-direction positions at a specific time instant.

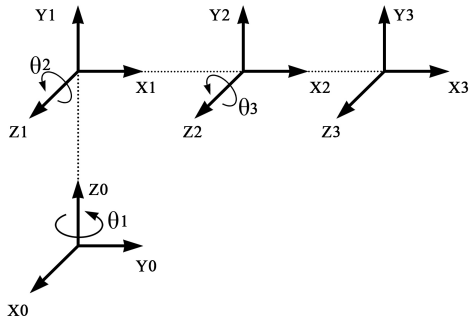

Figure 1. Crane kinematic structure.

Table 1. DH parameters

Link	a_i	α_i	d_i	θ_i
1	0	$\frac{\pi}{2}$	l_1	q_1
2	l_2	0	0	q_2
3	l_2	0	0	q_3

The model is established using the Denavit–Hartenberg (DH) parameters in Table 1.

Hence, the four homogeneous transformation matrices from frame 0 to 3 are obtained as follows:

$$T_1^0(q_1) = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_2^1(q_2) = \begin{bmatrix} c_2 & -s_2 & 0 & l_2c_2 \\ s_2 & c_2 & 0 & l_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_3^2(q_3) = \begin{bmatrix} c_3 & s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

and

$$T_3^0(q) = \begin{bmatrix} c_3c_{23} & -c_1s_{23} & s_1 & c_1(l_2c_2 + l_3c_{23}) \\ s_1c_{23} & -s_1s_{23} & -c_1 & s_1(l_2c_2 + l_3c_{23}) \\ s_{23} & c_{23} & 0 & l_1 + l_2s_2 + l_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The position of the end-effector, as a function of the joints angles, can be represented as:

$$P_e = \begin{bmatrix} c_1(l_2c_2 + l_3c_{23}) \\ s_1(l_2c_2 + l_3c_{23}) \\ l_1 + l_2s_2 + l_3s_{23} \end{bmatrix}$$

where q_i, with $i = 1, 2, 3$, indicates the crane joints angles and q is the joints vector representation. Moreover, c_i and s_i denote the sine and cosine of the corresponding angle q_i, respectively.

For the action space in this environment, the effect of the hydraulic cylinders that actuate the crane links is neglected, and the crane angles are assumed to be controlled directly. Hence, the action space in this environment is described as:

$$q_1 = |\ 0\ \ 0\ \ 0 \ldots|$$
$$q_2 = |-0.2 : 0.1 : 0.2|$$
$$q_3 = |-8 : 4 :\ \ 8|$$

Note that q_1 is an array of zeros and only a 2D workspace is considered. Hence, in the upgraded environment, we have 25 different combinations of actions, where we take the permutations of q_2 and q_3 and set the number of elements of q_1 to 25 elements. Moreover, the dynamical model [Equation (3)] was developed using the Lagrange approach and the equation of motion, which are used to extract the torque control sequence from the optimal control sequence of joint angles.

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + g(q) = \tau \tag{3}$$

where q, $\dot{q}$, and $\ddot{q}$ denote angle, angular velocity, and acceleration, respectively, and M(q), C($\dot{q}$, q), and g(q) denote inertial, centrifugal, and gravitational matrices, respectively.

2.2. Reward Structure

The impact velocity v_i is calculated at the time step, t_m, in which the relative distance between the crane's load position and the wave amplitude, D_r, becomes zero. Equation (4) is used to calculate v_i for the initial and upgraded environments. The v_i value is used in the structure of the instant reward [15], which is assigned to the agent when the impact occurs, according to Equation (5). Due to the fact that, as with any other RL algorithm, the previously mentioned reward structure causes a delay to the learning process as the agent has to wait until the impact takes place to receive its reward, an intermediate reward structure was developed to give the agent the possibility to collect rewards before reaching to the impact point. More precisely, the reward in this case is related to the change of the relative distance D_r; when there is a reduction in D_r, it means that the agent is approaching the barge and a positive reward is taken, according to Equation (6).

$$v_i = \frac{D_r(t_m) + |D_r(t_m + 1)|}{t_{step}} \tag{4}$$

where $t_{step} = 0.2$ s.

$$r = \begin{cases} 250, & \text{if } 0 < v_i \leq 0.05 \\ 200, & \text{if } 0.05 < v_i \leq 0.3 \\ -30, & \text{if } v_i > 0.3 \\ -0.01, & \text{else} \end{cases} \tag{5}$$

$$r = \begin{cases} 1, & \text{if } \mathrm{Dr}(i+1) < \mathrm{Dr}(i) \\ 0, & \text{else} \end{cases} \tag{6}$$

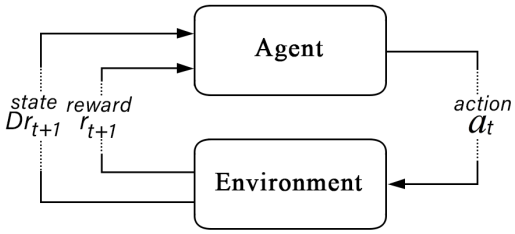

Figure 2. Agent–environment interactions.

3. RL ALGORITHMS

3.1. Q-Learning algorithm

The state-of-the-art Q-learning algorithm is used in this work with deterministic dynamics state estimation and one-step TD (TD(0)) learning approach; the structure of the Q-Learning algorithm was modified by implementing the reward structures described in Section 2, the state observation methodology depending on the environment under consideration, and the setting parameters in Table 2. The full structure of the algorithm is described in Algorithm 1.

Algorithm 1 Q-Learning Algorithm.

initialize Q-function, $Q(s, a) = 0, \forall s, a$
set initial state s
policy = $[exploratory, greedy]$
let P(exploratory) = Probability of selecting an action according to exploratory policy.
repeat
  set P(exploratory) = ϵ
  for every time step k=0,1,2, … **do**
    if policy = exploratory **then**
      select action a with an exploratory policy
    else
      select action a with an greedy policy
    end if
    apply a, measure next state s' (according to the considered State Observation)
    if impact occurred **then**
      measure impact velocity v_i and the instant reward
      set r' = the instant reward
    else
      measure intermediate reward
      set r' = the intermediate reward
    end if
    $Q'(s, a) = Q(s, a) + \alpha(r' + \gamma \max_{a'} Q(s', a') - Q(s, a))$
    $s = s'$
  end for
  update N-steps, update ϵ
until v_i in the acceptable range, or N-steps reaches the limit.

Table 2. Algorithm setting

Setting	Value	Setting	Value
t_{step}	0.2 s	α	0.0001
Episode length	90 s	γ	0.97
Action space	A_v or A_q	ϵ	0.9
State space	150 states	ϵ_d	$0.95/100step$
Agent	P_h(1)+ 30 s of P_w	N-steps(limit)	10,000

3.2. Double Q-Learning algorithm

The Double Q-learning algorithm has the ability to determine an unbiased estimate of the Q-value, because for each update, one set of weights is used to determine the greedy policy and the other to determine its value [18]. Algorithm 2 is the structure of the Double Q-learning in this framework, where the action a^* is the maximal valued action in state s', according to the value function, for example, Q^A.

Algorithm 2 Double Q-Learning Algorithm.

initialize $Q^A = 0, Q^B = 0, \forall s, a,$
set initial state s
repeat
  Choose a, based on $Q^A(s, \cdot)$ and $Q^B(s, \cdot)$, observe r, s'
  Choose (random) either UPDATE(A) or UPDATE(B)
  if UPDATE(A) **then**
    Define $a^* = \arg\max_a Q^A(s', a)$, $Q^A(s, a) = Q^A(s, a) + \alpha(r + \gamma Q^B(s', a^*) - Q^A(s, a))$
  else if UPDATE(B) **then**
    Define $b^* = \arg\max_a Q^B(s', a)$, $Q^B(s, a) = Q^B(s, a) + \alpha(r + \gamma Q^A(s', b^*) - Q^B(s, a))$
  end if
  apply $[exploratory, greedy]$ on the average value of Q^A, Q^B
  if impact occurred **then**
    measure impact velocity v_i
  end if
  $s = s'$
until v_i in the acceptable range, or N-steps reaches the limit.

3.3. Algorithm setup

In Table 2, α is the learning rate, γ is the discount factor, ϵ is the exploration rate, and ϵ_d is Decay factor of ϵ. A_v and A_q are the action spaces in the initial environment and the upgraded one, as mentioned in Section 2, and α is tested for different values to measure the algorithm performance.

4. SIMULATION RESULTS

4.1. In the initial environment

4.1.1. Episode examples

Figure 3 contains acceptable landing operations, where the agent managed to land the load on the vessel with impact velocity less than 0.3 m/s.

Note that, at the beginning of the episodes, the hoist velocity is fixed to 0.3 m/s; hence, the learning process takes place after the hoist reaches 2 m from the average wave amplitude. Moreover, this fixed starting action helps to avoid the delusion at the beginning of an episode, for example, going above the starting position,

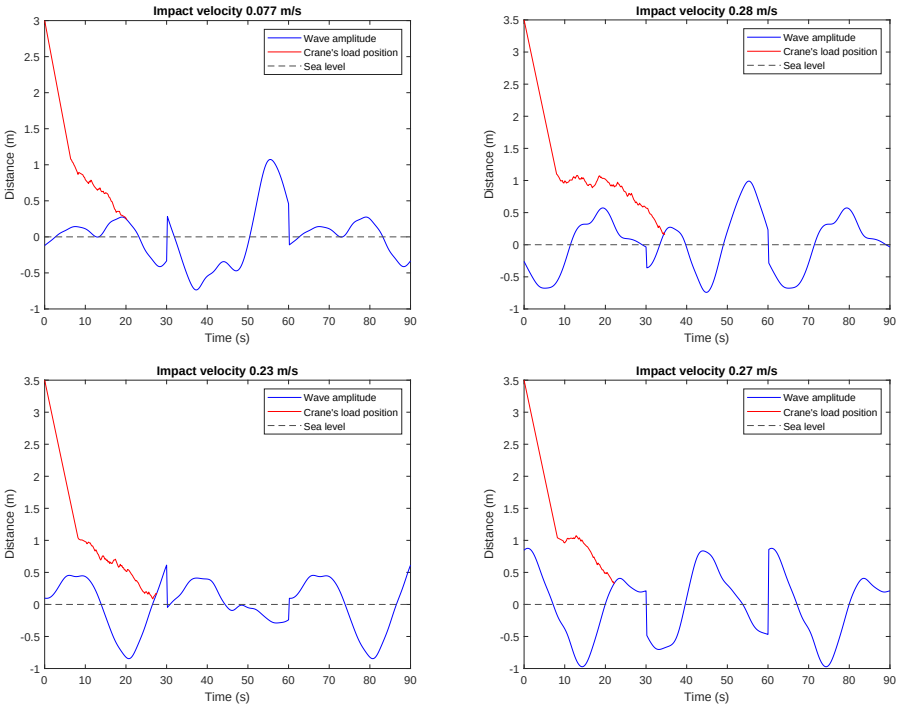

Figure 3. Acceptable landing episodes.

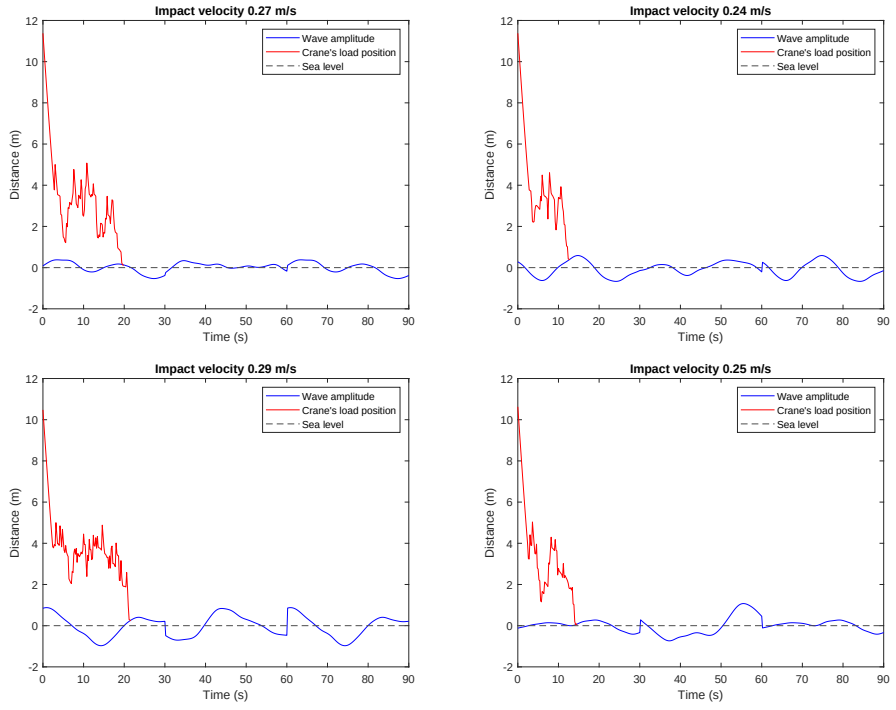

Figure 4. Acceptable landing episodes.

which would slow down the learning process. In addition, these results were obtained with the initial setting mentioned in Table 2.

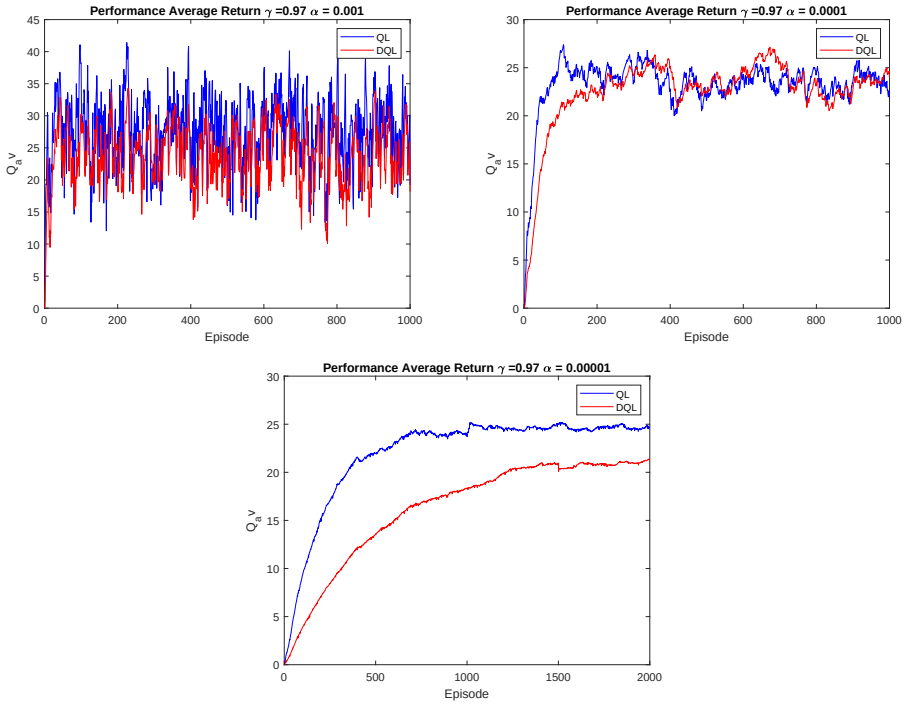

Figure 5. A comparison of QL and DQL with $\gamma = 0.97$.

4.1.2. Q-learning vs. double Q-learning

Since the algorithm is model-free, it is not possible to measure the bias that usually exists in the Q-learning algorithm. Hence, the results of the Q-learning are compared to the Double Q-learning results so the bias value can be measured. Figure 5 shows that DQL has the same sparsity α relation as QL, but the bias of the asymptotic value between the QL and the DQL increases when the α value is decreased.

Moreover, it is clear that DQL does not suffer from the delusion of the values of the average return at the beginning of the learning process; on the contrary, QL does suffer from overshooting the asymptotic value, which is clear in the case of $\alpha = 10^{-4}$.

Note that the episodes number in the case of $\alpha = 10^{-5}$, as shown in Figure 5, had to be increased as the convergence of the DQL was delayed to the episode number 1500, and this is one of the drawbacks of the DQL, while decreasing α, the convergence to the asymptotic value delays more.

4.2. In the upgraded environment

The same Q-learning algorithm was tested in the upgraded environment, and it can be noticed that in the majority of the episodes in this case, the agent could find an accepted impact velocity at early stages of the episode time span, as shown in Figure 4.

Moreover, by using the dynamical model, the input torques corresponding to the optimal control sequence of angles of one of the episodes can be calculated, as shown in Figure 6, where it is clear that Link 2 has the highest inertia, which extends the limits of the torque domain.

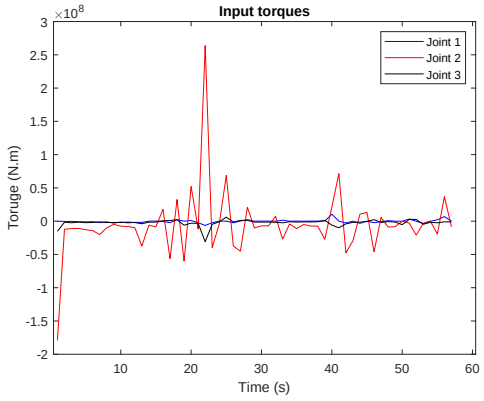

Figure 6. Example of a torque optimal control sequence.

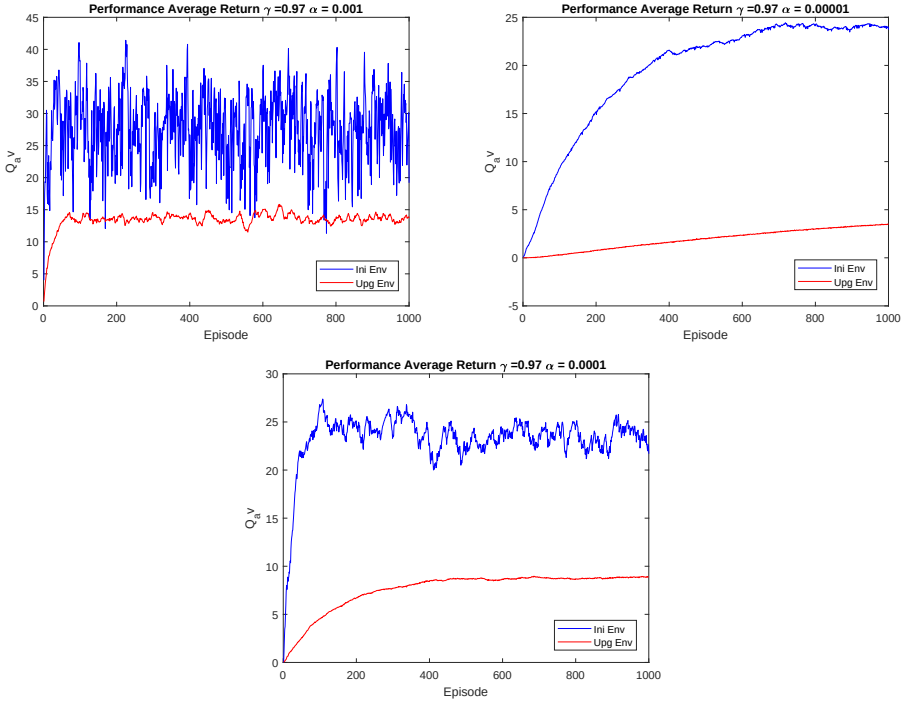

Figure 7. Environments comparison.

4.3. Environment comparison

Figure 7 indicates two differences between the initial environment (Ini Env) and the upgraded environment (Upg Env). First, the average return in the case of the upgraded environment is much less sparse than the initial environment. Second, the asymptotic average return values do not coincide.

4.4. Testing trials

Reinforcement learning is, by definition, an online learning environment. There is no separate test phase because the agent never stops learning. In other words, RL problems are usually of the "continual learning" type and the goal is to get the highest total reward, which is usually discounted over time. The most common technique for measuring RL performance is to look at the average return. In contrast with the supervised learning case, there are no standard performance measures yet, even in popular application domains[19]. Hence, the usual testing phase is to compare the algorithms to each other, as in our comparison between Q-learning and

Table 3. Agent accuracy on 500 episodes with $\gamma = 0.97$

Agent	Initial pos	Accuracy
QL $\alpha = 0.001$	$3.5m$	77%
DQL $\alpha = 0.001$	$2.5m$	73%
QL $\alpha = 0.0001$	$3.5m$	75%
DQL $\alpha = 0.0001$	$3m$	71%
QL $\alpha = 0.00001$	$3.5m$	77%
DQL $\alpha = 0.00001$	$2.5m$	80.8%

Double Q-learning in Section 4. In this section, we aim to generalize the optimal control sequences that were achieved in the online learning process. Hence, a testing sample of 500 episodes was generated considering the initial environment. For specific configurations, the agent was forced to follow a randomly selected control sequence from those considered optimal in the online learning process. The accuracy, which describes the agent success ratio to achieve acceptable impact velocity among the 500 episodes of the testing sample, is reported in Table 3 for different configurations. Note that the agent accuracy on the upgraded environment does not exceed 1.5% for all configurations.

5. DISCUSSION

The purpose of this work is to control the impact that occurs in offshore crane load landing operations in one of the most complicated environments due to the impact of several conditions. Although strong assumptions were set, two environments were established. The first considered the crane hoist as a point mass and the second was enlarged with the crane structure using a robot-like mathematical model, which was established and simulated using MATLAB and Simulink. Both environments have the sea waves model involved in the vessel's motion where the crane's load is supposed to be placed.

Optimal control sequences were generated using the Q-learning algorithm, and they managed to reach acceptable impact velocities in online learning processes. The performance of the Q-learning algorithm was tested, and we conclude the following: an intermediate reward structure is needed to overcome the effect of the agent delusion that occurs, as the reward in the usual Q-learning algorithm is delayed to the impact point; hence, less time would be required to accomplish the task.

For the same discount factor, the lower the learning rate, the less sparse the values of the average return, and the more delayed the convergence to the asymptotic value. Moreover, the asymptotic value of the average return is reduced while reducing the learning rate; although this effect is not essential in the initial environment, it is clear in more complicated environments such as the upgraded environment in our work. The change of the discount factor—for the same other hyperparameters values—is directly proportional to the asymptotic value of the average return but with no magnificent effect on the convergence time. The algorithm performance was compared to the Double Q-learning technique; hence, the bias in the Q-learning technique can be adjusted.

Reinforcement learning, in general, does not have a separate testing sample, as every learning process is online and every control sequence is unique on its own episode. In other words, we can say that the obtained control sequence has local optimality. This was verified. We tested the agents on a sample of 500 episodes and concluded that the accuracy of the agent is facing a high variance; even when we set all the algorithm hyperparameters to be fixed, the initial position of the crane where the control sequence is tested still has a significant effect in the variation of the accuracy value.

On the other hand, although the accuracy range of the best trials in the initial environment was between 71% and 80%, this range was tremendously reduced to between 0.1% and 1.5% in the upgraded environment. We attribute that to the size of action_space, as the initial environment agent has one input of 11 actions and the

upgraded environment agent has three inputs and 25 actions; hence, the more complicated the action_space, the more unique the optimal control sequence and the harder it is to find a global optimal control sequence. Thus, choosing the action_space is important not only from the learning point of view but also from the feasibility in the physical domain; in this work, it is shown that avoiding setting up sudden movements and more effort to the links with high inertia are needed.

DECLARATIONS

Authors' contributions

Made substantial contributions to conception and design of the study and performed data analysis and interpretation: Maamoun KSA, Karimi HR

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was partially supported by the Italian Ministry of Education, University and Research through the Project "Department of Excellence LIS4.0-Lightweight and Smart Structures for Industry 4.0" and in part by the Horizon Marie Skłodowska-Curie Actions program (101073037).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Park HC, Chakir S, Kim YB, Lee DH. A Robust payload control system design for offshore cranes: experimental study. *Electronics* 2021;10:462. DOI
2. Huster A, Bergstrom H, Gosior J, White D. Design and operational performance of a standalone passive heave compensation system for a work class ROV. In: OCEANS 2009. IEEE; 2009. pp. 1–8. DOI
3. Ni J, Liu S, Wang M, Hu X, Dai Y. The simulation research on passive heave compensation system for deep sea mining. In: 2009 International Conference on Mechatronics and Automation. IEEE; 2009. pp. 5111–16. DOI
4. Zhu M, Zhang P, Zhu C, Jia X. Dynamic analysis and optimal control of the landing process of the offshore installation. *Adv Mech Eng* 2017;. DOI
5. Mackojć A, Chiliński B. Preliminary modelling methodology of a coupled payload-vessel system for offshore lifts of light and heavyweight objects. *Bulletin of the Polish Academy of Sciences: Technical Sciences* 2022;70:e139003. Available from: DOI
6. Idres M, Youssef K, Mook D, Nayfeh A. A nonlinear 8-DOF coupled crane-ship dynamic model. In: 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; 2003. p. 1855. DOI
7. Ellermann K, Kreuzer E, Markiewicz M. Nonlinear dynamics of floating cranes. *Nonlinear Dynamics* 2002;27:107–83. DOI
8. Cha JH, Roh MI, Lee KY. Dynamic response simulation of a heavy cargo suspended by a floating crane based on multibody system dynamics. *Ocean Engineering* 2010;37:1273–91. DOI
9. Spong MW, Hutchinson S, Vidyasagar M, et al. Robot modeling and control. vol. 3. Wiley New York; 2006. DOI
10. Williams LA. Modelling, Simulation and Control of offshore crane Develop a kinematic and dynamic crane model and study of several control designs [MastersThesis]. Universitetet i Agder; University of Agder. Norway; 2018. Available from: http://hdl.handle.net/11250/2564033.

11. Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018. Available from: https://mitpress.mit.edu/9780262039246/.
12. Andersson J, Bodin K, Lindmark D, Servin M, Wallin E. Reinforcement Learning Control of a Forestry Crane Manipulator. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2021. pp. 2121–26. DOI
13. Gaudet B, Linares R, Furfaro R. Deep reinforcement learning for six degree-of-freedom planetary landing. *Adv Space Res* 2020;65:1723–41. DOI
14. Sun X, Xie Z. Reinforcement Learning-Based Backstepping Control for Container Cranes. *Mat Pro Eng* 2020;2020. DOI
15. Ding M. Reinforcement Learning For Offshore Crane Set-down Operations [MastersThesis]. University of Croningen. Netherlands; 2018. Available from: https://www.ai.rug.nl/~mwiering/Thesis_Mingcheng_Ding.pdf.
16. Vazirizade SM. An intelligent integrated method for reliability estimation of offshore structures wave loading applied in time domain [PhdThesis]. The University of Arizona. USA; 2019. Available from: https://repository.arizona.edu/handle/10150/636592.
17. Küchler S, Mahl T, Neupert J, Schneider K, Sawodny O. Active control for an offshore crane using prediction of the vessel's motion. *IEEE/ASME Transactions on Mechatronics* 2010;16:297–309. DOI
18. Hasselt H. Double Q-learning. *Advances in neural information processing systems* 2010;23. Available from: https://proceedings.neurips.cc/paper/2010/hash/091d584fced301b442654dd8c23b3fc9-Abstract.html.
19. Zhang C, Vinyals O, Munos R, Bengio S. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:180406893* 2018.