Data Driven Control Based on Deep Q-Networks Algorithm for Heading Control and Path Following of a Ship in Calm Water and Waves

Sivaraman Sivaraj[a], Suresh Rajendran[a], Lokukaluge Perera Prasad[b]

[a] *Department of Ocean Engineering, Indian Institute of Technology, Madras, Chennai - 600036. India*
[b] *Department of Technology and Safety, UiT The Arctic University of Norway Tromsø 9037, Norway*

Abstract

A reinforcement learning algorithm based on Deep Q-Networks (DQN) is used for the path following and heading control of a ship in calm water and waves. The rudder action of the ship is selected based on the developed DQN model. The spatial positions, linear velocities, yaw rate, Heading Error (HE) and Cross Track Error (CTE) represent the state-space, and a set of rudder angles represents the action space of the DQN model. The state space variables are in continuous space and action spaces are in discrete space. The decaying ϵ-greedy method is used for the exploration. Reward functions are modelled such that the agent will try to reduce the Cross Track Error and the Heading Error. Based on the literature available, the L7 model of a KVLCC2 tanker is used for testing the algorithm. The vessel dynamics are represented using a 3DoF maneuvering model that includes hydrodynamic, propeller, rudder and wave forces. The wave disturbances are calculated from the second-order mean drift forces . The environment is assumed to have the Markov property. The CTE and HE are calculated based on the Line of Sight (LOS) Algorithm. The effect of Pre-trained weights on different heading actions is investigated based on the exploration threshold. The DQN is trained and tested for heading control and path-following in calm water and different wave headings.

Keywords: Path following, Heading Control, DQN, Cross Track Error (CTE), KVLCC2, Reward function, LOS, waves

Summary of Notations

u	Surge velocity	ψ_d	Desired heading angle
v	Sway velocity	y_e	Cross Track Error (CTE)
r	Yaw moment	Δh	Look ahead distance
x, y	Spatial positions	δ	Rudder angle
ψ	Heading angle	γ	Discount rate parameter
γ_p	Horizontal path tangential angle	α	Step size parameter
β	Side slip angle	ϵ	Probability of taking a random action in an ϵ-greedy policy
$K_{p,i,d}$	PID controller gains	U	Total speed = $\sqrt{u^2 + v^2}$
HE	Heading error($\psi - \psi_d$)	lbp	Length between perpendicular (L_{pp})
a	An action	N	Number of episodes
n_p	Propeller speed (rps)	y_k	Waypoint index number
r	A reward	s	state

A_t	Action at time t	R_t	Reward at time t
κ	Wave encounter angle t	S_t	State at time t
$r(s, a, s')$	expected immediate reward on transition from s to s' under action a	ω	Wave frequency
X_H,Y_H,N_H	Surge force, lateral force, yaw moment around midship acting on ship hull except added mass components XP Surge force due to propeller	X_R,Y_R,N_R	Surge force, lateral force, yaw moment around midship by rudder
X_P	Force due to propeller	X_D,Y_D,N_D	Surge force, lateral force, yaw moment by wave drift
X_S, Y_S, N_S	Second order mean drift force and moment by waves		

1. Introduction

Model based control techniques have been extensively used for ship navigation such as heading control and path following. This requires prior knowledge of the system dynamics. The accurate modelling of the nonlinear dynamics of the ships can be a complex process. Studies on autopilot systems on model based control techniques are widely available [1] [2] [3] [4] [5] [6]. Controllers like PID and LQR are simple to implement. However, they are limited in its robustness under wave disturbances. Literature are available on the autopilot design of ships based on PID, linear and Nonlinear Model Predictive Control (NMPC), etc.,[7][8][9][10] [11] [12]. With the advancement of the modern control theory, nonlinear control theory like adaptive control [13], backstepping control [14], robust control [15], sliding mode control [16] [17], fuzzy logic control [18] [19] etc. have also been widely used in the field of ship steering control. In fields that require precise motion control like automotive suspension systems, robotics, servo mechanics etc. state of the art methods have been developed to take account of the uncertainty in the system dynamics, nonlinear friction and bounded disturbances [20] [21]. Function approxiamtors like Neural Network (NN) and Fuzzy Logic System (FLS) are used for the parameter estimation and are further augmented by advanced nonlinear friction models [22] and novel adaptive control strategies [23].

However, with the technological leap that gave the tremendous capacity to store and retrieve data, data driven models are gaining attention among the ship research community. Reinforcement Learning (RL) is one of the three branches of the machine learning paradigms in which the agent explores the environment and takes appropriate decisions. RL helps to find the solution to an optimal control problem that is stated as Markovian Decision Process (MDP) [24]. The advantage of the MDP solution procedure is that it can handle nonlinear stochastic dynamics [25]. Even though the MDP solution works only for discrete state space, function approximators help to overcome this limitation. One such approximate RL technique is used in this paper for ship steering control problems. In addition to the generality of the MDP solution, the main advantage of the RL based control is that it's a model-free control technique. Therefore, it does not require any prior information on the system dynamics. The agent learns itself from the stored samples of transitions and rewards during the exploration and finally solves for the optimal or near-optimal controller by following the best policy. Therefore, RL can be a strong candidate for finding the optimal controllers for nonlinear stochastic systems with unknown dynamics and disturbances. However, training demands computational time and resources, and the model's convergence depends on the accurate tuning of the hyperparameters. Data driven models are popular in the field of self-driving cars, terrain robots and drones and are also gaining popularity among the ship navigation community. This paper develops a data-driven model based on the RL technique for the path following and heading control of a KVLCC2 tanker in calm water and waves.

In RL, the control terminologies like controller, controlled system and control signal can be replaced by agent, environment and action. The agent can generally learn from the experiences according to the corresponding states by either value iteration or policy iteration. Policy iteration is guaranteed for faster convergence when the system has a lower variance bound in the state space transformation. Recently there has been a surge in the research on the RL (DRL) based methods for marine vehicle navigation and control. Path planning and collision avoidance of marine vehicles have been the key areas in which RL based techniques have been widely used. RL based on Tabular Q-Learning [26], [27], Deep Deterministic Policy Gradient (DDPG) [28] and Proximal Policy Optimization (PPO) [29] [30] have been investigated for path planning. Different RL techniques based on policy searching methods such as Actor-critic methods [31] [32],Proximal Policy Optimization (PPO) [33] etc. have been investigated by researchers on collision avoidance. [34] [35] investigated the straight line and curved path following of three marine vessels using DDPG methods. Reward function based on Gaussian distribution was used. They investigated the RL controller performance during the path following of three vessels under ocean currents. However, a single random path was simulated and the cross track error was analyzed.

Even though policy search based methods are guaranteed for faster convergence, these works are limited in robustness and tend to local convergence [36] [37]. Meanwhile, Value iterations based RL algorithm relatively takes a longer time for convergence. However, they are suitable for systems with a significant variance during the state space transition [38]. The Agent can learn from the experiences in two ways. If it stores the previously sampled transitions and learns from the stored sampled transitions, then this is considered an OFF policy algorithm. Otherwise, if the agent uses only the currently sampled transitions for learning, then it is called an ON Policy algorithm. An Off policy and value iteration based method called Deep Q-Network (DQN) is used in this paper. The tabular method of Q-learning is popular among RL applications which have discrete state-action space, and have also been used for path planning of ships [39]. But, when the number of states is high, larger computational resources are required. In parallel, DQN offers a robust method that has continuous state space (S) and discrete action space (A) [40]. Deep Q network is a value iteration based RL algorithm that provides the contraction mapping from state to action ($\mathcal{S} \longrightarrow \mathcal{A}$). In dynamic programming, state values are generally updated by the Bellman equation where the probability to taking any action is known. However, in RL, the samples are obtained from the memory buffer which has information about the transition. The DQN is neuro-dynamic programming where the network weights are updated by the backpropagation method by calculating the appropriate gradients. The DQN is generally known for its robustness when the system has high variance [41].

In this paper, the heading control and path following tasks in calm water and waves are formulated using an RL algorithm based on DQN. Here, the ship is considered as an agent and the states in continuous space are obtained from a vessel dynamics model of a KVLCC2 tanker. The action space consists of a set of rudder angles. The vessel dynamics return the current states of the ship such as spatial position, yaw angle, yaw rate, etc. The reward function is represented by a step and linearly decrement function based on HE, CTE and the distance to the goal point. The LOS algorithm is used to estimate the CTE and HE based on the agent's current state. Many episodes (in the range of thousands) are simulated and the Q-values and best policy functions are calculated. The RL method aims to select optimal rudder angles from the allowed action space to achieve the required path based on a long-term cumulative discounted reward. Transfer learning techniques [42] [43] are tried for heading control. This is achieved by transferring the final weights obtained from training the agent in a particular heading and testing it for another heading action. The wave forces for different amplitudes are calculated from the second order mean drift forces. The capability of the RL method to reject the wave disturbances is analyzed.

Most of the aforementioned studies have focused on the development of RL algorithms for ship path planning and collision avoidance. A few studies have been conducted on the heading control and path following, however in the absence of any wave disturbances. To the best knowledge of the authors, no studies were conducted on the path following of a large tanker in calm water and waves. The main contribution of our work is the investigation of the performance of a DQN based RL controller in wave disturbance rejection during a path following task. The wave disturbances are represented by second order mean wave loads and hence give a good approximation to the practical scenario. The development of such a RL controller involves the designing of an appropriate reward function and tuning hyperparameter that can function in different environmental conditions. In order to ensure that the RL controller works for all scenarios, different kinds of paths and headings have been studied under different wave conditions. The main challenge is the building of a RL based controller with appropriate reward function and termination criteria for an under-actuated system with huge inertia. The ship dynamics is highly nonlinear, therefore designing of the reward function is a strenuous one. An inappropriate reward function will lead to local convergence during learning. Appropriate termination criteria will define the agent's exploration capacity during the training and result in faster convergence. In addition, the hyper parameters such as size of the memory

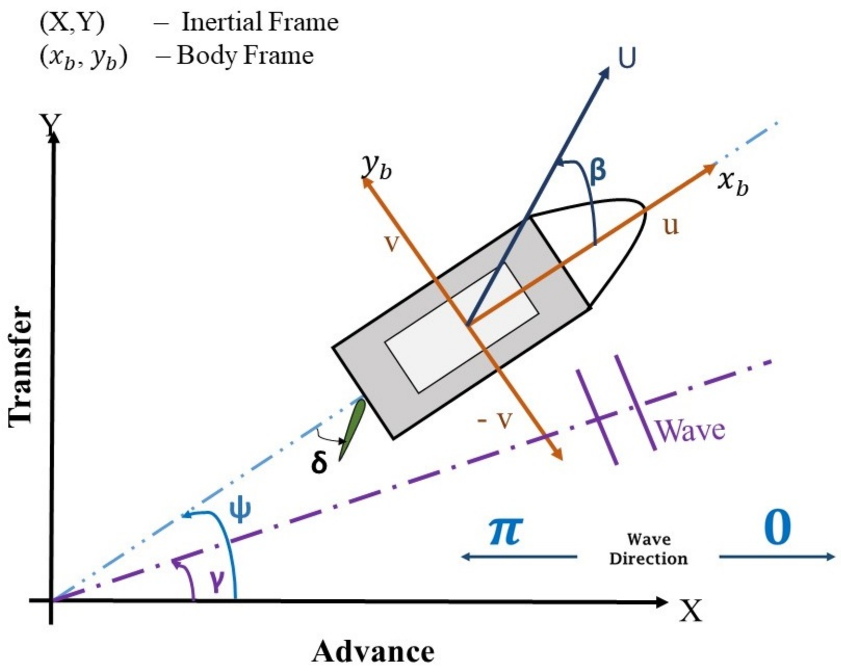


Figure 1: Ship inertial and body coordinate systems

buffer, neural network's architecture, loss function and epsilon decay rate,etc. should be tuned and chosen in such a way that faster convergence and global stability is achieved during the training period.

2. Ship Dynamics and Wave Modeling

The 3DoF ship dynamics represented in the body coordinate system is shown in Fig.1. The ship dynamics are modelled based on the Manoeuvring Modeling Group (MMG) model [44] [8].

$$\begin{aligned}(m+m_x)\dot{u}-(m+m_y)vr-x_Gmr^2 &= X_H+X_R+X_P+X_S\\ (m+m_y)\dot{v}-(m+m_x)ur+x_Gm\dot{r} &= Y_H+Y_R+Y_S\\ (I_zG+x_G^2m+J_z)\dot{r}+x_Gm(\dot{v}+ur) &= N_H+N_R+N_S\end{aligned} \tag{1}$$

The governing equations for the coupled surge, sway and yaw are represented by Eqn.1. The numerical simulations are carried out for a L7 model of a KVLCC2 tanker (model scale 1:45.7). The main particulars of the ship are given in the table 2. The hydrodynamic (H), propeller (P) and rudder (R) forces/moments are calculated based on [44]. Fig.2 compares the numerical and experimental open loop turning circle trajectory in calm water for a ±35° rudder angle. There is a good agreement between the numerical trajectory and the experimental trajectory obtained from [44].

Particular	Full scale	L7-Model
Length between perpendiculars (m)	320	7
Breadth (m)	58	1.27
Draft (m)	20.8	0.46
Displacement $\Delta(m^3)$	312600	3270
mass (kg)	3.12×10^8	3351.75
Longitudinal centre of gravity $X_G(m)$	11.2	0.25
Block coefficient C_b	0.810	0.810
Propeller dia. $D_p(m)$	9.86	0.216
Rudder height H_R (m)	15.80	0.345
Rudder Area $A_R(m^2)$	112.5	0.0539

The second order mean drift forces are calculated based on [45]. They affect the surge, sway and yaw motions. The second order longitudinal and lateral drift forces and yaw moments are pre-calculated in the range of -180 to 180 deg wave heading and for a range of frequency between 0 and 13 rad/sec in the model scale. During the numerical simulation, the wave incident angle with respect to the body frame is calculated and the corresponding mean second order force/moment is interpolated accordingly. The experimental open loop trajectory [46] is compared with the numerical trajectory in waves. The simulations are conducted in a wave steepness (H/λ) of 0.02 and 0.038, and the wave length is equal to the ship. The numerical and the experimental open loop trajectory

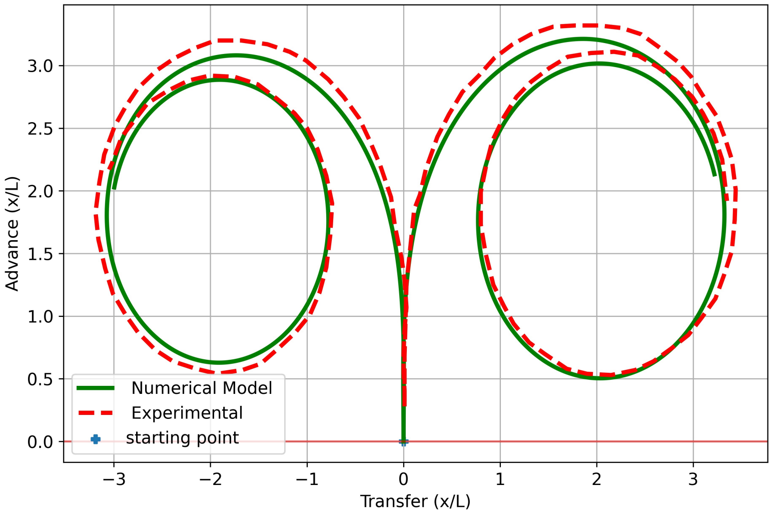


Figure 2: Comparison of the numerical and experimental open loop turning circle trajectory

in a regular wave of 0.02 steepness is shown in Fig.3. A detailed comparison between the numerical and experimental open loop trajectory in waves of different characteristics and the validation of the wave modelling can be observed in [11].

3. Line of Sight (LOS) Algorithm

The LoS algorithm is developed based on [47] [8]. Based on the current states of the vessel and its distance from the desired trajectory, the LOS algorithm returns the HE and CTE. Waypoints are kept at an equal distance of 2 times the *lbp*. The tracked way points are updated in the history cell. The agent switches between the way points when the distance to the next target point is less than the distance to the already traced point. The MMG model provides the yaw angle (ψ) in the range of -360 to 360 deg. Therefore, the yaw angle is clipped in the LOS algorithm between -180 and 180 deg.

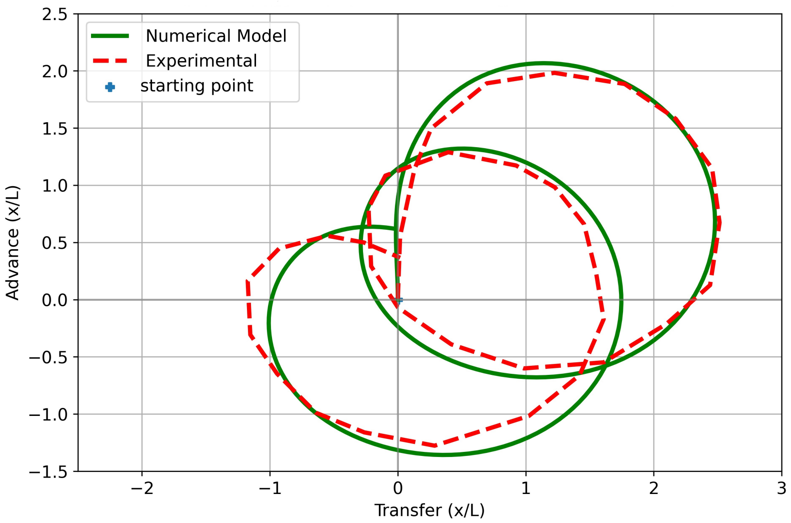


Figure 3: Comparison of open loop trajectory in waves

4. Deep Q Networks (DQN)

DQN is a neural network based Q-learning method. The mathematical model given in Eqn.1 is used to obtain the state space of the agent for each time instant. In calm water, the same equation without mean second order drift force and moment is used to obtain the state space. Initially, the ship is starting at the origin $(0, 0)$ with 1.2 m/s surge velocity and 12 *rps* . The objective of the DQN is to approximate the state-action value function through training a policy network which will return the maximum cumulative discounted reward ($R_{t_0} = \Sigma_{t=t_0}^{T} \gamma^{t-t_0} r_t$). γ is the discount factor and r_t is a reward at a particular time t. The agent chooses an optimal action (a^*) through the trained optimal policy $\pi^*(s)$, Eqn.2. The decaying ϵ-greedy algorithm with a threshold ϵ-value is used for exploration, Eqn.3. Many episodes are simulated and the Q-values are approximated.

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\ Q^*\ (s, a) \tag{2}$$

$$\epsilon(i) = 0.2 + (0.99 - 0.2) * \exp\left(\frac{-0.5 \times i}{600}\right) \tag{3}$$

Training episodes are bounded by a set of termination criteria. When HE is more than 180 degree and the cross track error is more than 9 times *lbp*, the episode will terminate. A negative reward below -0.5 will also result in episode termination. In addition, when the number of steps reaches the horizon or when the ship traced the final waypoint in the trajectory sequence, an episode will terminate. The fixed size horizon steps to terminate the episodes resemble the Monte Carlo Policy Evaluation in DQN learning. However, a proper approximation is required for estimating the horizon steps. The horizon is fixed based on an approximate time that the agent takes to track the points, which depends on the ship velocity and trajectory length. Depending on the initial heading error, tolerance will be added while calculating the horizon.

Feed Forward Neural Network (FFNN), has an architecture of two hidden layers of 128 nodes with sigmoidal activation function, and linear input-output layers are used as Q-network and is shown in Fig.4. The Q-values are approximated for every step action during the training phase based on Bellman Equation [48]. The trained Q-network provides the contraction mapping for every possible state to optimal action. The temporal difference error (Δ) between the policy net and the target net is used to calculate the loss function, Eqn.4.

$$\Delta = Q(s,a) - (r + \gamma \underset{a'}{\text{argmax}} \; Q(s',a)) \tag{4}$$

where s, s', a, and π are state, next state, action and policy at a time t. The Mean Square Error (MSE) is calcualted from Δ based on Eqn.5. $Q(s,a)$ values are calculated from the policy net and $Q^{\pi}(s', \pi(s'))$ are calculated from the target net.

$$Mean\ Square\ Error = \frac{1}{2}\Delta^2 \tag{5}$$

DQN is an offline policy algorithm, and it tries to minimize the error between the target and policy net. This is achieved by batch mode training in which

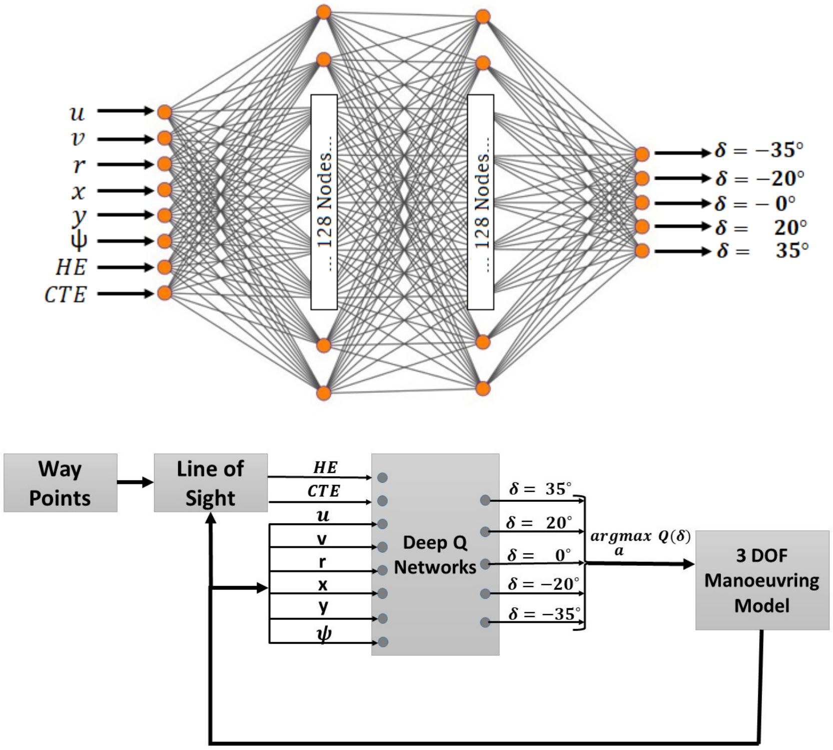


Figure 4: Architecture of Neural Network and Control Structure Diagram

samples are taken from the reply memory buffer. Batch mode training is used for gradient calculation and the calculated gradient weights are updated by Adam's optimizer gradient descent method [49] [50] [51]. The size of the memory buffer has a significant role in the training process. Since, the problem is modeled as a finite horizon MDP, memory size is selected as 10 times the event horizon size. For eg., if the desired trajectory length is 300m and the ship moves with an initial surge speed of $1m/s$ and the numerical integration time step is 1s, the horizon size will be approx. 300 steps with an additional tolerance. In this case, the memory size will be approx. 3000.

Table 3. DQN Hyper Parameters

Shape of the NN	8-128-128-5
Activation function	linear-sigmoid-sigmoid-linear
Step size parameter	0.98
Target Update Interval	10 episodes
Optimizer	Adam's Method
Learning Rate	0.001
β_1,β_2	0.9,0.999
Batch size	128
Exploration threshold	0.2
Memory Buffer Size	10 × horizon

5. Reward Function

The reward function is a performance measure for the contraction mapping ($Q^* : \mathit{State} \times \text{Action} \longrightarrow \mathcal{R}$) and designed based on the HE, the CTE and distance to the goal. Two functions from Eqn.6 and Eqn.7 are stepwise linear decremented and used in reward Algorithm.1. Eqn.6 defines the CTE based reward distribution. When the CTE is within the range of a $\pm 0.3 \times lbp$, it yields high reward. Similarly Eqn.7 defines the HE based reward distribution. When the heading error is higher than $90°$, the algorithm will only focus on minimising the HE. Eventually, the algorithm focuses on minimizing the CTE and HE to zero value.

$$\mathcal{R}_1(ye) = \begin{cases} 100 & \text{if} \quad 0 < abs(\text{ye}) < 2. \\ 20 & \text{elif} \quad 2 < abs(\text{ye}) < 3.5 \\ (1 - \frac{abs(ye)}{7 \times lbp}) \times 20 & \text{otherwise} \end{cases} \tag{6}$$

$$\mathcal{R}_2(HE) = \begin{cases} 1 - (\frac{HE}{90}) & \text{HE} \leq 90^\circ \\ 2 - (\frac{HE}{90}) & \text{otherwise} \end{cases} \tag{7}$$

Algorithm 1: Reward Function

```
Data: HE, CTE, Current state, Previous state, Goal
Result: Numerical value of the reward (range from -0.5 to 101)
if current state is closer to goal than previous state then
    Flag = True
else
    Flag = False
if Flag = True then
    Reward = 1 + ($\mathcal{R}_2$(HE)×$\mathcal{R}_1$(CTE));
else
    Reward = $\mathcal{R}_2$(HE);
```

6. Results

The DQN algorithm is tested for the heading control and path following of a tanker in calm water and waves. The cumulative discounted reward, the length of episodes (i.e. no of time step) and the heading error during the training phase are plotted against the corresponding episode numbers and discussed. The rudder deflection for the final episode (i.e. the test case) against each time step is also plotted. A time step of 1s is used in the simulation. This section is divided into the following subsections 1) Heading control in calm water 2) Path following in calm water 3) Heading control through the transfer of pre-trained network weights. 4) Heading control in different wave headings 5) Path following in different wave headings 6) A senstivity study on the effect of different hyperparameters on the convergence of the reward function.

6.1. Calm Water Condition

Initially, the agent is at the origin of the global frame with 1.2 m/s surge velocity and 12 *rps*. Waypoints are used to represent the desired trajectory. The waypoints are kept at an interval of 2 × *lbp*. The transition formations $(s_t, a_t, r, s_{(t+1)})$ are stored in the memory buffer. Once the memory buffer is filled, the learning will take place for each step by mini-batch gradient back propagation. During the training, the old information is eradicated and restored by the new transition information in the memory buffer. Mini-batch for the network's error back propagation is sampled from the memory buffer using random call.

6.1.1. Heading Control

The Heading control of the ship in all directions is achieved by proper DQN training. The heading actions such as -45°, -90°, -135° and -180° are simulated and are shown in Fig.5. The given vessel headings are negative based on Fig.1. 8 state variables as shown in Fig.4 are used as input. However, the heading control can also be achieved by various combinations of input such as using 2 states (HE and CTE), six state (u, v, r, x, y, ψ), 5 states $(x, y, \psi$, HE, CTE) and 3 states (x, y, ψ). For heading control, since the agent always tries to reduce the heading error, the yaw angle is an important input and hence common in all the combinations of input. 8 input model facilitates faster convergence and is used in this paper.

A running average of the cumulative discounted reward, length of episodes and the heading error for 100 episodes for -90° heading control are plotted in the Fig.6. The given cumulative reward is the average of the sum of all the rewards for a particular episode. A high cumulative reward is achieved after

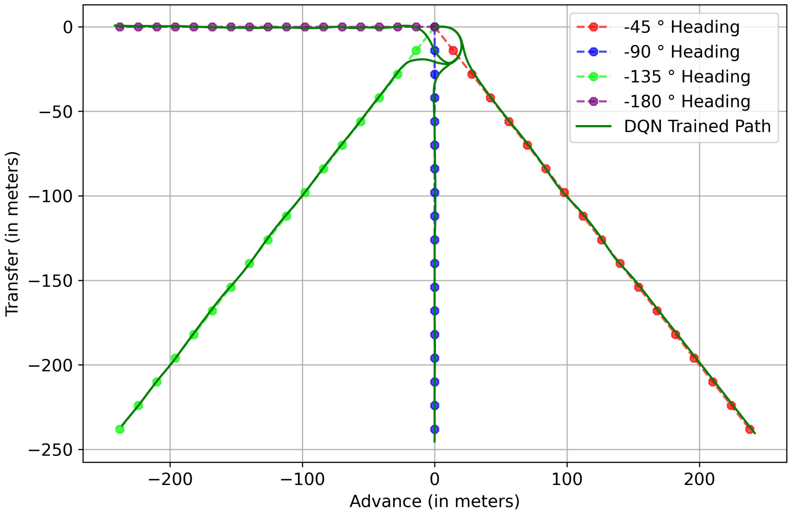


Figure 5: Heading Control in Calm Water

approximately 500 episodes, and the episode duration also increases. The major part of the learning happens during the first 500 episodes. By 500 episodes, the ϵ value, i.e exploration, reaches 0.72. The cumulative reward converges even though the agent explores it in subsequent episodes . The agent performs the task within 300 steps (i.e. 300s). Rudder deflections for the final test case are also plotted in Fig.6. Unlike the model based control techniques, the DQN employs a discrete action space. Even though the number of allowed rudder angles is limited to 5 for this study, the DQN algorithm is able to perform heading control with the allowed rudder actions.

6.1.2. Path Following in Calm Water

Path following is performed for a S-shaped and an elliptical trajectory. The S-shaped curve is defined by the Eqn.8

$$y = \frac{250}{1 + e^{-0.03(x-200)}} \tag{8}$$

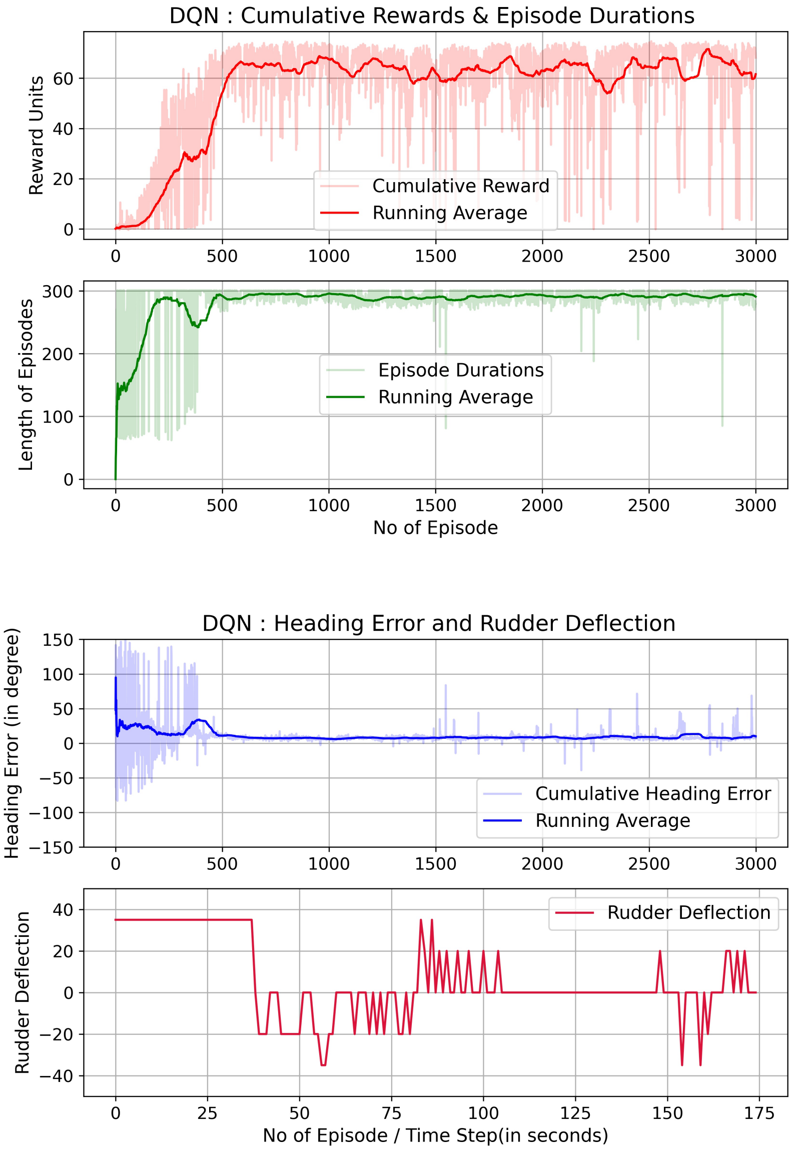


Figure 6: -90 Degree heading case- Cumulative reward, Episode duration, and Heading error vs. Episode number, Final rudder deflection for the test case vs. time step

Similarly, an elliptical trajectory ($a = 110$m, $b = 55$m) is also considered for the path following task. After sufficient training, the RL algorithm is able to track both paths as shown in Fig.7. Unlike the heading control case, HE and CTE are inevitable input to the NN for the path following task. The initial heading angle of the agent is zero and the initial surge speed is tangent to the X-axis (ie. along advance direction). Therefore, the agent is able to track the S-curve smoothly. While tracking the elliptical path, even though the agent takes the rudder action at the start of the simulation, the hull responds slowly due to the agent's huge inertia. Additionally, because of the waypoint switching strategy adopted in the LOS algorithm, the agent skips the first two points and joints the third one. The running average of 100 episodes for the cumulative reward, length of the episodes and the heading error for the S-shaped path are plotted in Fig.8. Like the heading control, a high reward (≈80) is achieved after 500 seconds and the length of the episode converges to ≈500 steps. The running average of HE converges. The final rudder deflection during the test phase is shown.

6.2. Effect of Pre-trained Weights

The neural network weights are randomly initialized at the start of the training. Based on the back propagation, the final converged weights are obtained by the end of training. To reduce the training time, the training can be started with the pre-trained weights with a limited range of exploration. This facilitates faster convergence in a few episodes. The heading control in 90,-55 and -105 deg is performed using the weights obtained from 45, -45 and -90 deg as shown in Fig.9. For example, the trained weights of -45 deg heading action are used for training the -55 deg heading. It has converged within 300 episodes, where the exploration rates are set between 35 % to 20 %. The gradual decrements in the exploration rate follow the eqn.3. Similarly, the DQN algorithm is trained for -105 deg heading control using the pre-trained weight from -90 deg. Here the exploration rate is kept between 40 % and 20 %. It takes approx. 300 episodes for convergence. The transfer of 45° trained weights for 90 deg heading action

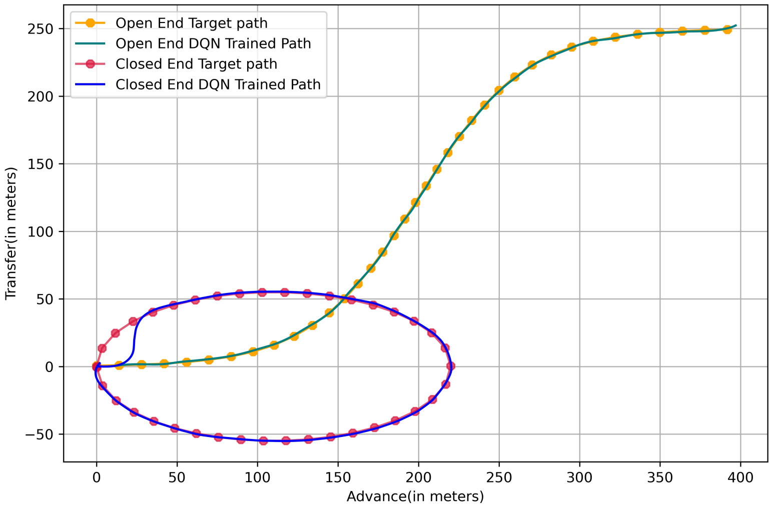


Figure 7: Trajectory Comparison of S-shaped and Elliptical Path Following in Calm Water

requires further training of episodes with a higher exploration rate of 60 % to 20 %. Fig.10 shows average running reward for the three heading actions using the pre-trained weights.

Therefore, the heading actions with a smaller deviation from the originally trained simulations require only a smaller number of training episodes. However, larger deviations from the trained path require a higher rate of exploration.

6.3. Heading control and path following in waves

During ship maneuvering in waves, the first order wave force effects on the yaw motion are filtered using a low pass filter. The second order drift force effects are counteracted by the rudder forces. Therefore, only the second order mean drift forces are considered in the following simulations. Prior to the RL simulation, the second order mean drift forces are calculated based on [45] for a range wave heading and frequency. The wave forces corresponding to the instantaneous ship heading are interpolated from the drift force data. The wavelength is kept the same as the ship length. To build a robust DQN model,

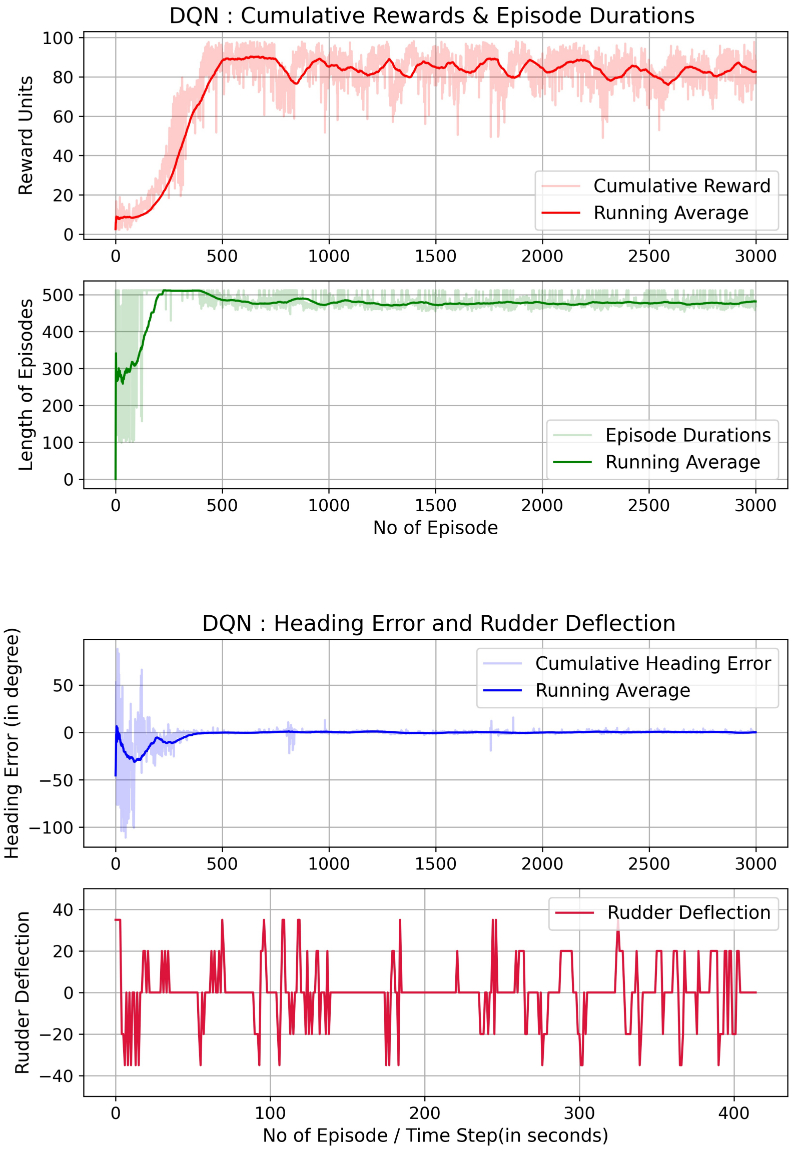


Figure 8: S-shaped Curve - Cumulative reward, Episode duration, and Heading error vs. Episode number, Final rudder deflection for the test case vs. time step

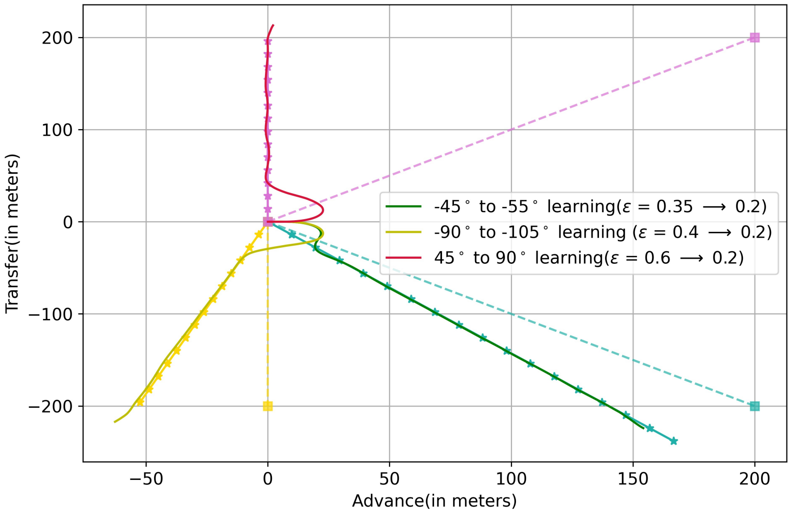


Figure 9: The effect of pre-trained weights on the ship heading control

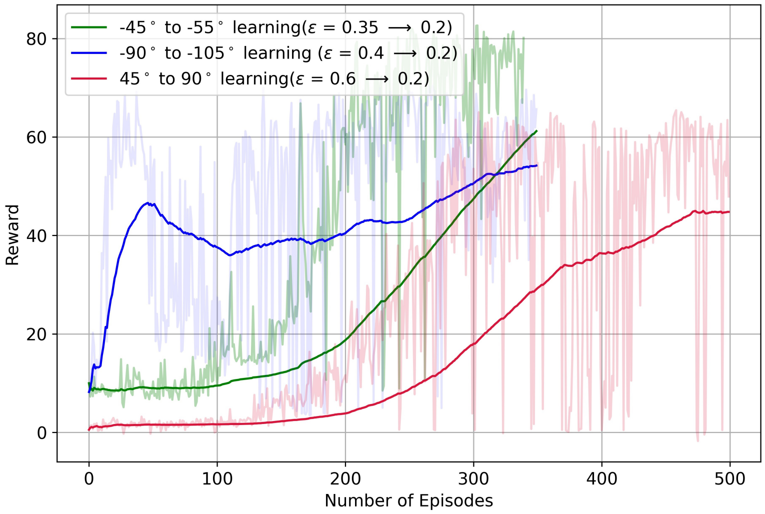


Figure 10: Convergence of average reward for three ship heading using pre-trained weights from the other headings

waves in multi-directions i.e. (-180,...-90,..0,..90,...,180 deg) are added as disturbances during each episode. A random call to select the wave direction for each episode is used, which results in a uniform distribution of the disturbance to the model during training. In this section, the agent is trained and tested to perform heading control and path following in waves.

6.3.1. Heading Control in waves

The heading control in waves has been performed for a range of wave headings. The ability of the RL algorithm to achieve -45, -90,-135, and -180 deg ship heading control is tested. For each course heading, the ship is tested for (0, $\pi/4$, $\pi/2$, $3\pi/4$, π, $-\pi/4$, $-\pi/2$, $3 - \pi/4$, $-\pi$ rad) wave headings. In this manner, we can assure that the ship is able to achieve a particular course heading irrespective of the wave direction. In addition, the ship is also tested for a wave steepness of 0.02 and 0.038. In the head and bow quartering waves, the surge speed reduces because of the added resistance in waves [8]. The propeller rotates at 12.1 *rps* in the smaller amplitude wave, and 20 *rps* in the larger amplitude wave so that it maintains the same surge speed for both cases. The ship always starts at the origin (0,0) in the global frame. The horizon steps for training in waves are two times the calm water ones. During the training, a particular wave direction in an episode is chosen based on a random call. In this manner, the agent is trained in all wave directions, and the final weights help the agent to achieve the heading control irrespective of the wave directions.

The DQN algorithm with the final weights are tested in regular waves of 0.07 and 0.13m wave amplitude and for eight different wave headings (0, $\pi/4$, $\pi/2$, $3\pi/4$, π, $-\pi/4$, $-\pi/2$, $3 - \pi/4$, $-\pi$ rad) . Fig.11 shows the heading control in 0.07m wave, and Fig.12 and 13 show the heading control in 0.13m wave. For all the cases, the ship starts at 0° heading. The trajectory is made of waypoints and the guidance principle is based on the LOS algorithm. Due to its huge inertia and being an under actuated system, the agent traverses the lower quadrants to achieve the -135 and -180 deg heading. The agent is able to achieve the headings irrespective of the wave directions. The agent switches between the

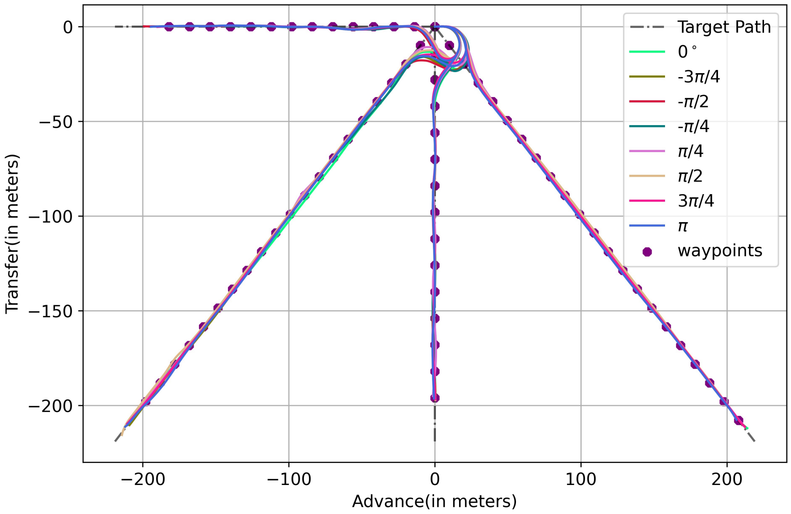


Figure 11: Heading Control in 0.07 meter waves

way points when the distance to the next target point is less than the distance to the already traced point. In this manner, the agent's path smoothly converges to the desired trajectory. The agent experiences large drifting force/moment in 0.13m wave as observed in Fig.13. Among the four tested ship heading scenario, -180 deg heading can be considered as the difficult maneuvering since the ship has to make a U-turn for achieving the heading control. In this case, the agent experiences the largest drift in -90 deg wave heading i.e. when the wave is coming from top to down in the Fig.13. Because of the vessel dynamics and the LoS algorithm, the vessel skips the first two way points. However, when the agent experiences low drift forces while performing the desired maneuvering (for e.g. in 90 deg wave heading), it tracks all the way points. It shows the capability of the DQN algorithm to capture the dynamics of the vessel.

The wave forces affect the ship velocity. Therefore, the number of steps required to trace the waypoints in a particular wave direction is different from the other wave cases. However, this does not pose any problem during learn-

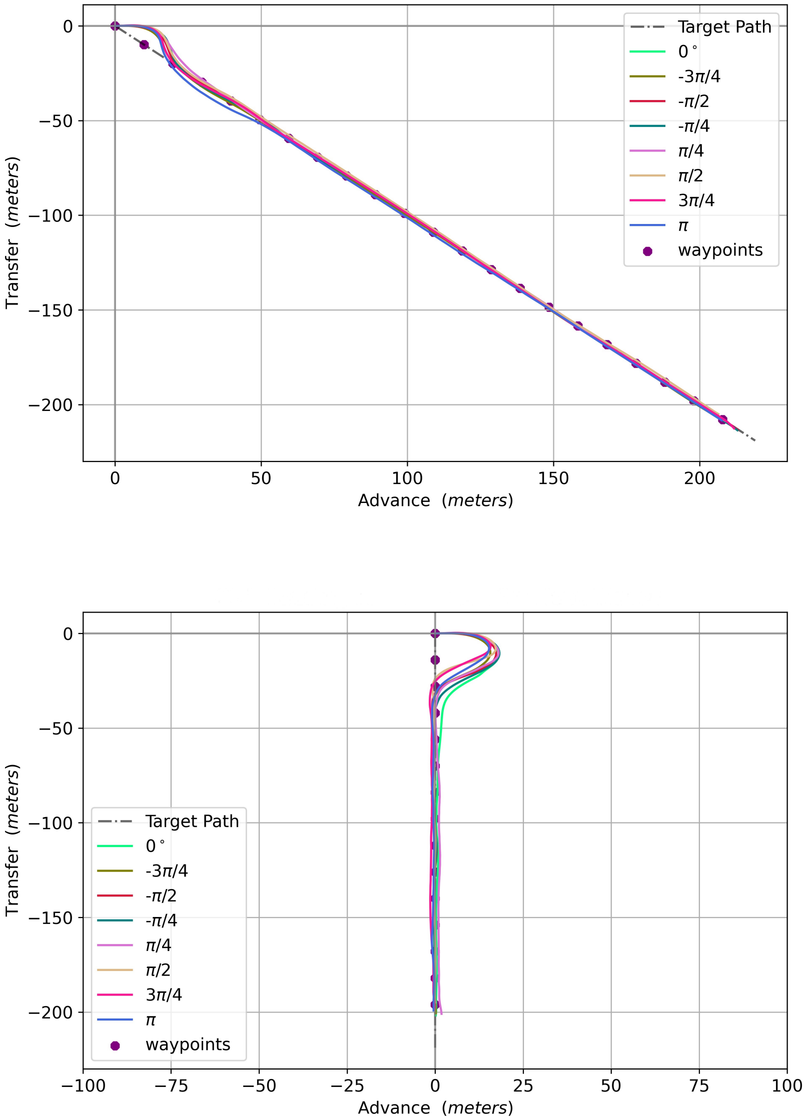
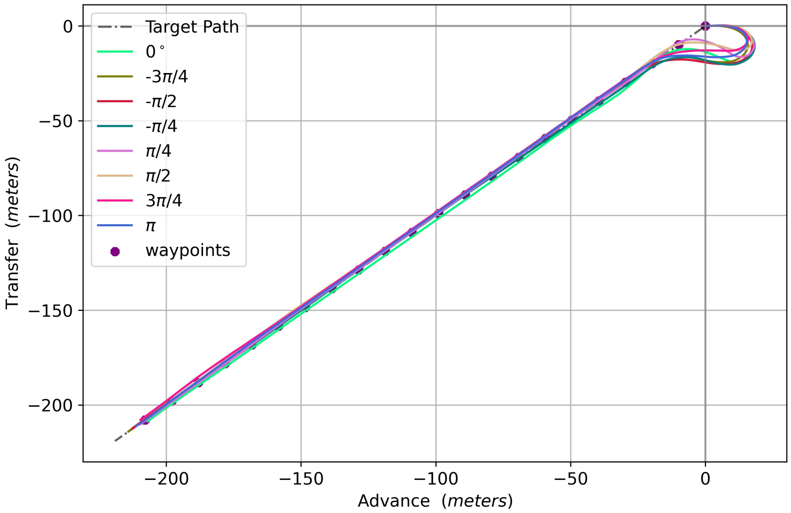


Figure 12: Trajectory Comparison of -45 and -90 degree Heading Control in 0.13 meter amplitude wave



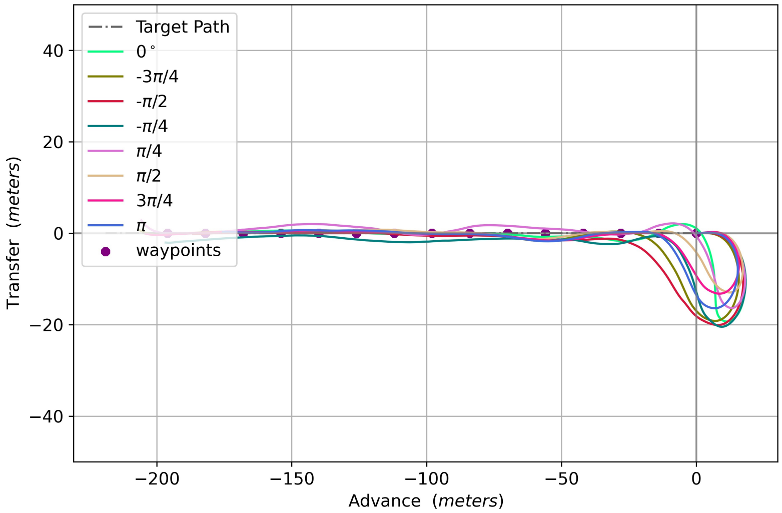


Figure 13: Trajectory Comparison of -135 and -180 degree Heading Control in 0.13 meter amplitude wave

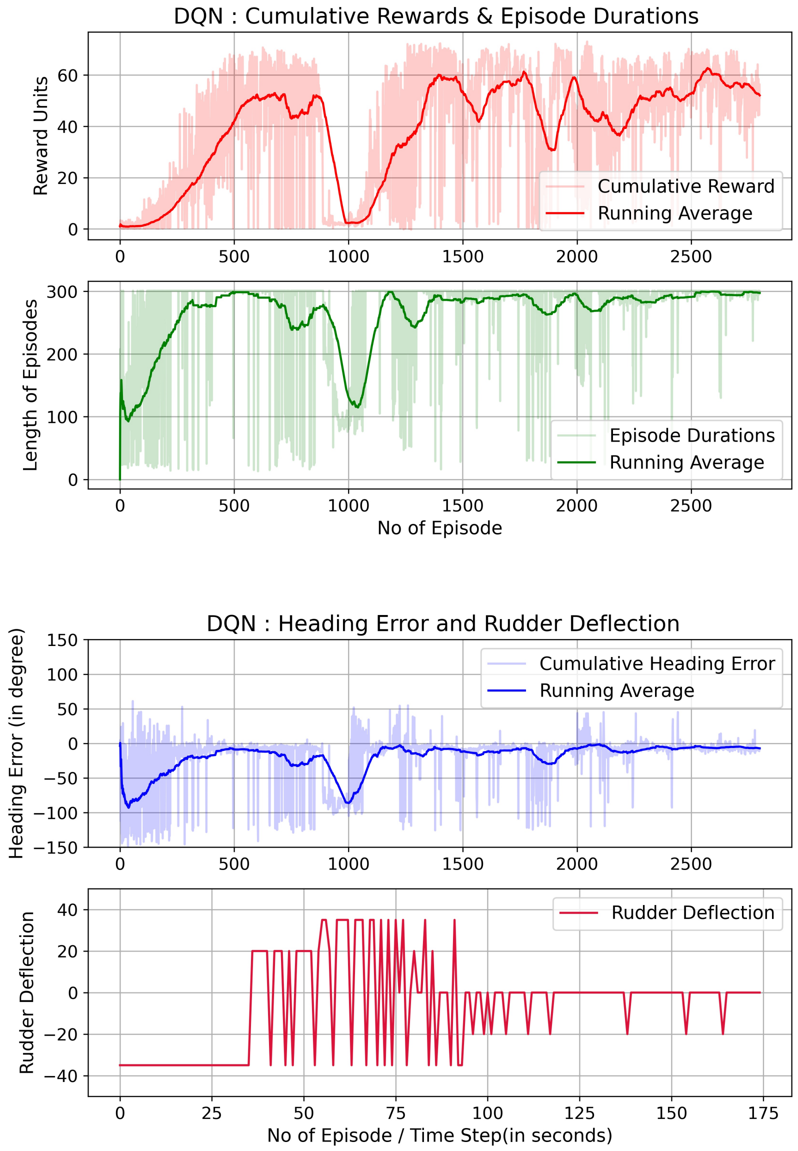


Figure 14: -135deg ship Heading in 0.07m amplitude wave- Cumulative reward, Episode duration, and Heading error vs. Episode number, Final rudder deflection for the test case vs. time step

ing. Fig. 14 shows the the running average of the cumulative reward, episode duration and the heading error plotted against the episode number during -135 deg heading control training. The reward plot shows the running average of 100 episodes and the cumulative discounted reward for each episode. The reward converges to 50 by approx. 500 episodes. However, at approx. 1000 episodes, it suddenly drops. It may be due the exploration that the agent performs at this stage. This is not observed for the calm water case. At 1000 episodes, the agent has an exploration rate, ϵ, of 50% (Eqn.3). The plot of the length of the episodes vs. episode number shows that the agent is able to track the given way points within 300 time steps (i.e. 300s). The heading error converges by 2800 episodes. The slight fluctuations is due to the exploration rate at this episode (28%). The rudder angle during the test phase is shown in Fig.14. Here the weights from the finally trained weights obtained from the last episode is used for testing the heading control in different wave headings. Since the action space is discrete, the rudder is allowed to take only 5 actions (-35,-20,0,20, and 35 deg). However, within the limited action space, the agent is able to track the way points and achieve heading control. A larger action space will facilitate a smoother rudder action however at the expense of higher computational time for the training. The DQN based control algorithm is able to reject the wave disturbances for both 0.07 and 0.13m amplitude waves.

6.3.2. Path Following in Waves

A S-shaped and a cardioid curve are designed for the path following problem in waves. The S-shaped curve which is already used in the calm water case represents an open end trajectory (Eqn.8). The cardioid curve generated based on the Eqn.9 represents a closed end trajectory. The variable ’a’ in Eqn.9 is chosen as 25.

$$\begin{aligned} x(\phi) &= 2a(1-\cos\phi).\cos\phi \\ y(\phi) &= 2a(1-\cos\phi).\sin\phi \end{aligned} \tag{9}$$

The path following in 0.07m amplitude wave is shown in Fig. 15 and 16, and the path following in 0.13m amplitude wave is shown in Fig. 17 and Fig.18. The

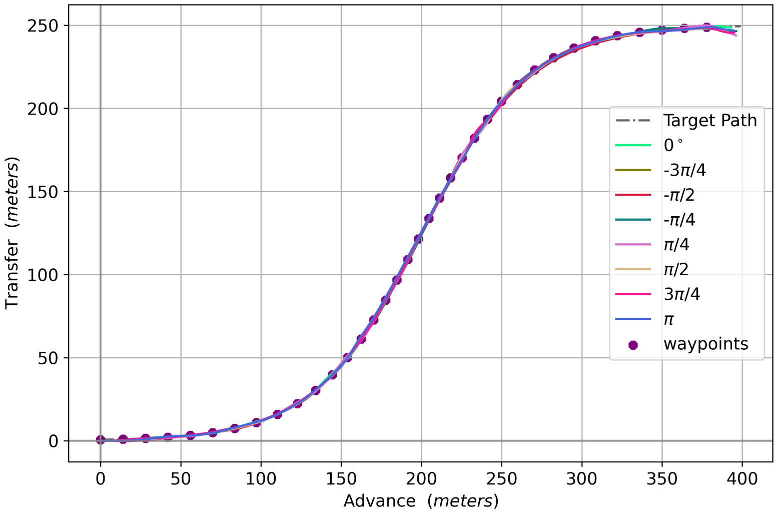


Figure 15: Trajectory Comparison of S-shaped Path Following 0.07 meter Waves

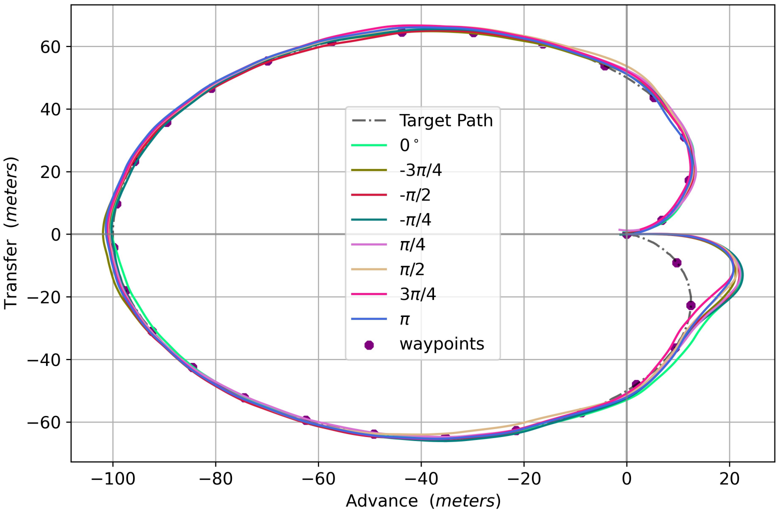


Figure 16: Trajectory Comparison of Cardioid Path Following 0.07 meter Waves

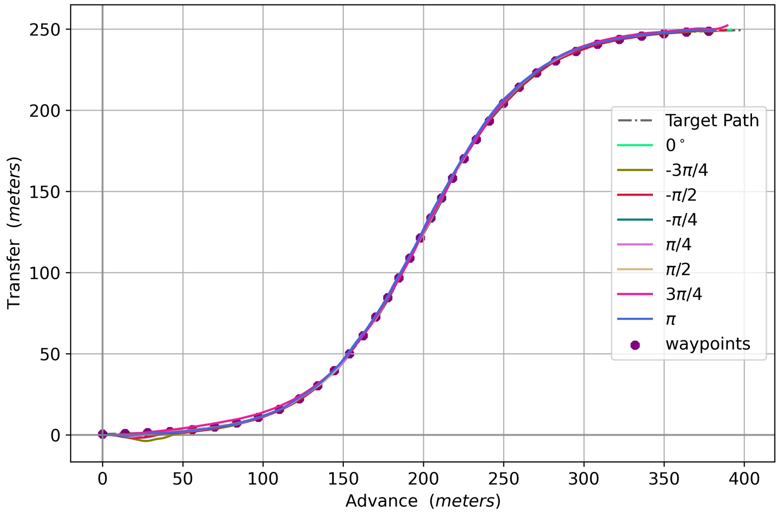


Figure 17: Trajectory Comparison of S-shaped Path Following in 0.13m Waves

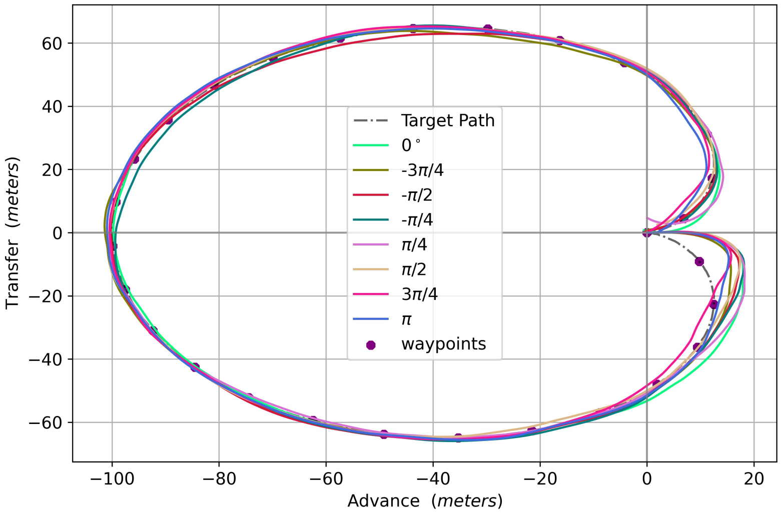


Figure 18: Trajectory Comparison of Cardioid Path Following in 0.13m Waves

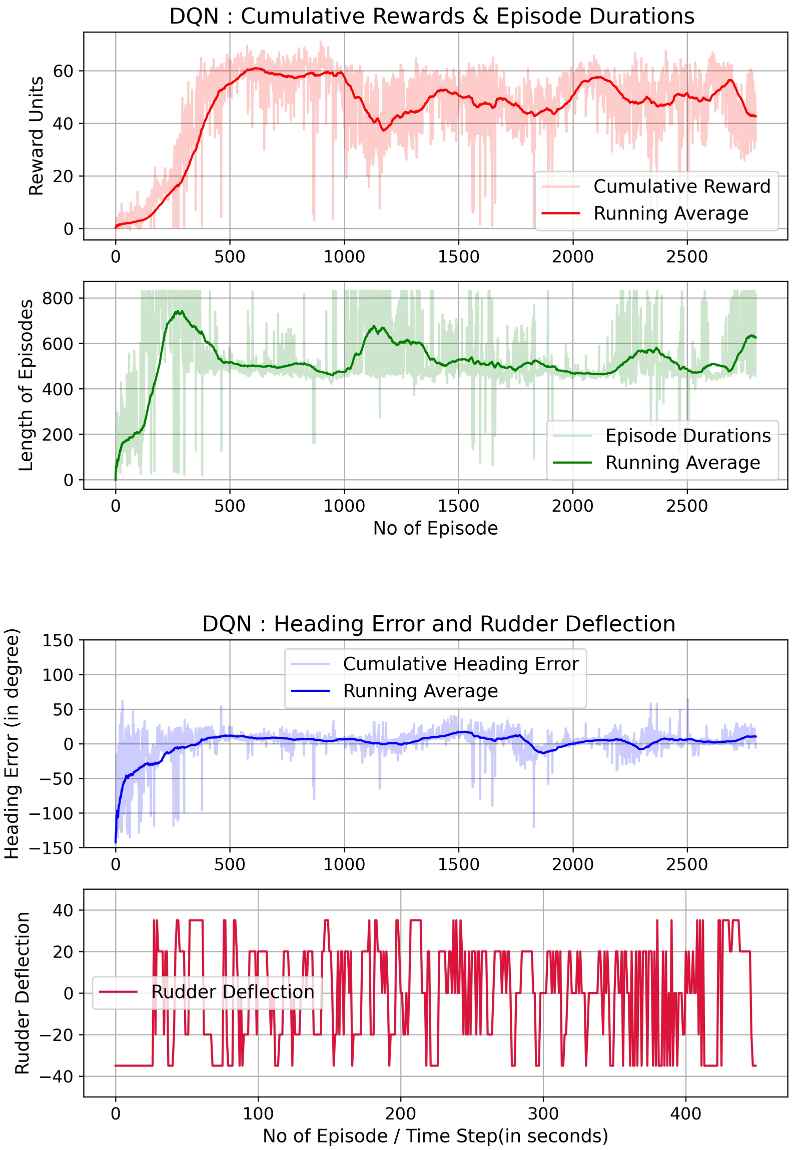


Figure 19: Cardioid path following in 0.07m amplitude wave- Cumulative reward, Episode duration, and Heading error vs. Episode number, Final rudder deflection for the test case vs. time step

ship is tested for 8 wave headings as mentioned in the previous sections. The trajectories in the wave headings are plotted in Fig.15 to Fig.18. The ship starts with a zero heading angle. For S-shaped curves, the surge speed is tangent to the curve at the starting of the simulation. Therefore, the ship is able to track all the way points for both wave amplitudes. However while tracking the cardioid curve, the surge speed is not tangent to the curve at the initial position . The waypoints are uniformly distributed along the length of the curves at an interval of $2lbp$. Because of the vessel dynamics and the LoS algorithm as mentioned in the previous section for -180 deg heading control, the vessel skips the first two way points and smoothly tracks the rest of the way points as shown in Fig.16. The vessel experiences larger drifting in 0.13m wave as shown in Fig.18. The cumulative and average rewards, length of episodes and the heading error during the training of the agent for the cardioid path following in waves are shown in Fig.19. The reward converges after 600s and the agent take approx. 500-600 steps (i.e. 500-600s) to complete the mission. The episode duration in waves increases than the calm water case for the convergence.The cumulative heading error for each episode represents the summed up value of the instantaneous heading error that the vessel experiences during a particular episode. Initially, the cumulative heading error is very high and it slowly converges later on. Since there will always be a heading error as the vessel follows the curved path, the cumulative error will never be zero. The DQN model is able to learn the dynamics of the vessel and is able to keep this error close to a minimum. The rudder deflection for the test case is shown Fig.19. DQN model can follow the path even with a limited rudder action. Even though the maximum rudder rate may be clipped in a practical scenario, a larger rudder action space in the DQN model will help to overcome the issue.

7. Performance of a PID Controller in waves

The RL based controller is able to achieve the heading control and path following in waves of different wave steepness. It will be interesting to see whether the same can be achieved using a simple PID controller. The KVLLC2 model performance in calm water and waves using a simple PID controller was investigated by [52]. The controller gains were estimated based on the pole placement method for a damping factor of 0.7 and closed loop natural period of 100s. The performance of the controller was tested for heading control in low amplitude waves of wave steepness $\frac{H}{\lambda} = 0.02$ i.e. wave amplitude of 0.07m in model scale. The same controller is further tested here for a larger wave amplitude of wave steepness $\frac{H}{\lambda} = 0.037$ i.e. 0.13m wave amplitude in the model scale and as given in the previous sections. Fig.20 shows the performance of the PID controller in waves of 0.13m wave amplitude. The PID controller is not able to achieve the desired heading and path following in larger wave amplitudes. Therefore, further tuning of the gains is required in order to obtain the optimal performance in waves of larger amplitudes. However, such a scenario is not observed for the RL based controller. The RL based controller is able to reject the disturbances due to larger waves as shown in Fig.12-18.

8. Sensitivity study on the hyper parameters

The DQN algorithm involves many hyper parameters. In this section, a sensitivity study on some important parameters such as node size and batch size, discount rate (γ), memory size and the selection of optimizer is conducted to understand its performance on an efficient learning of the agent.

8.1. Hidden Layer Nodes and Batch Size

The NN model has two hidden layers. Its node size and the mini batch size affect the learning. A sensitivity study of these variables on the average cumulative reward of the agent is carried out. The contour plot, fig. 21, shows the average of the average cumulative reward during 500-1000 episodes plotted against the number of nodes and the batch size.. The sensitivity test is conducted for a 45° heading action in calm water. The average reward increases with an increase in the number of nodes and batch sizes. Any value of the

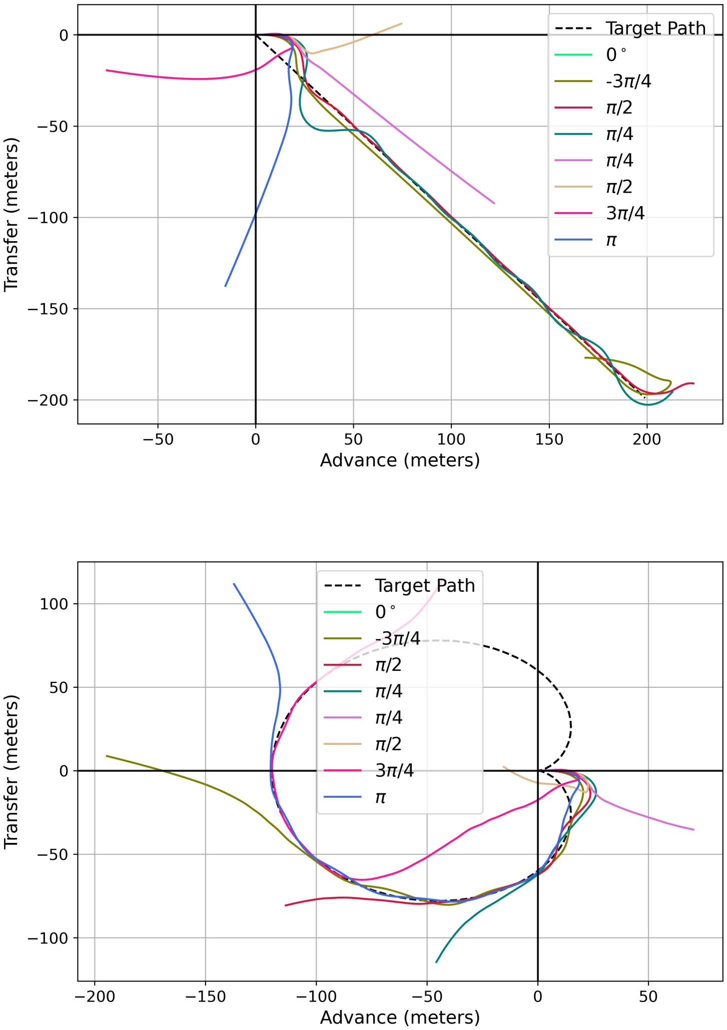


Figure 20: PID performance in 6m waves

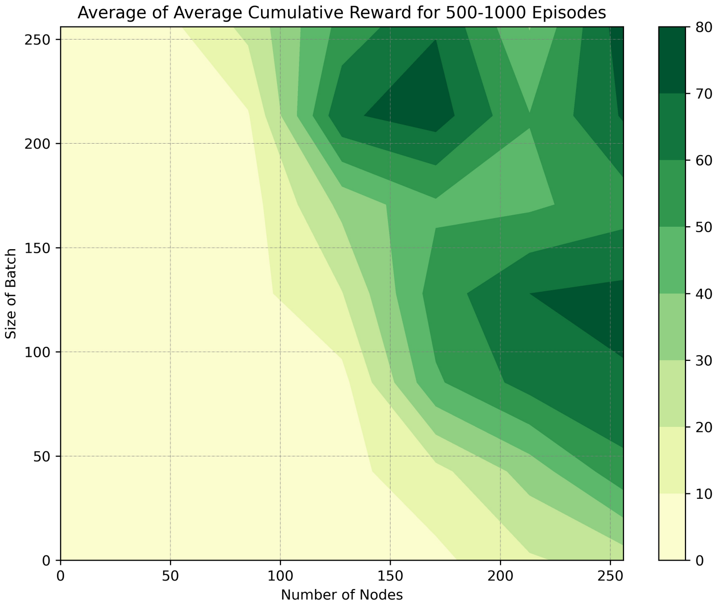


Figure 21: Node- Batch Size Comparison Against the Average of the Average Cumulative Reward

node or the batch size above 200 results in a very high reward. In this paper, the node and the batch size is kept at 128. Even though this yields a lower reward, the required computational time is only half of a 256-256 node-batch size combination.

8.2. Discount rate parameter and memory buffer

The discount rate parameter (γ) plays an important role in learning. A very high (0.99) and low (0.5) values of discount rate results in poor convergence of the reward function. The γ value between 0.7-0.95 is found to be optimal for the current problem since it facilitates faster learning as observed in Fig.22,.

Batch mode of training samples the data from the memory buffers which stores the current state, next state, reward and action. Various studies have been conducted on the effect of memory size[51]. For the current problem, it is

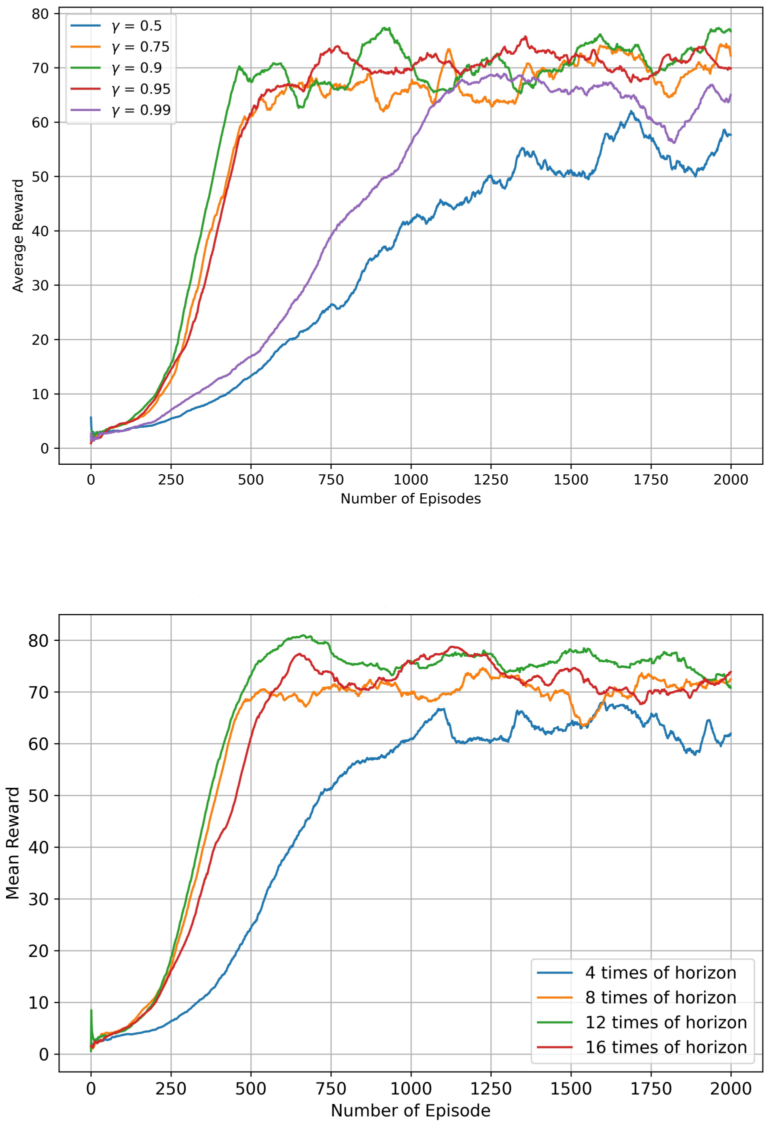


Figure 22: Effect of Discount Rate and the Length of Memory Buffer on the Average Reward

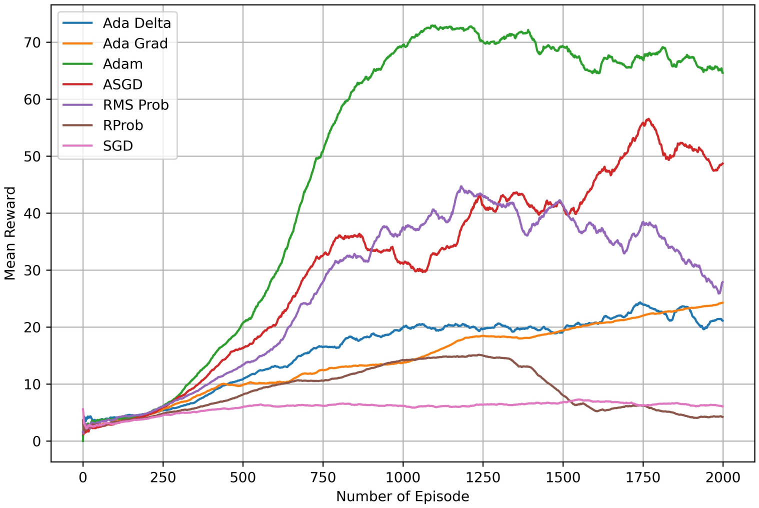


Figure 23: Comparison of Different Optimizer Performance

optimal to use a memory size of 8-16 times of the horizon as shown in Fig. 22.

8.3. Stochastic Optimizer Comparison

The optimizer plays an important role in minimizing the loss function. The performance of different optimizer in the convergence of the average reward value is plotted in fig.22. The average of the cumulative reward for each episode is plotted against the number of episodes in Fig.22 . The case study is carried out for a 45-degree heading action in calm water. The other hyperparameters remain the same as given in table 3. The performance of the optimizers such Ada delta [53], Ada Grad [54], Adam [50], ASGD [55], RMSProb [56], RProb [57] and SGD [58] are compared. The Adam optimizer results in the largest reward and ASGD and RMSprob show good results. The adaptive learning rate methods like AdaDelta and AdaGrad give better results than one step gradient update method[58].

9. Conclusion and future work

In this work, we have successfully implemented a DQN algorithm for the heading control and path following of a tanker in calm water and waves. It shows the capability of the data driven model to steer an under-actuated system with huge inertia. A 3DoF mathematical model is used for simulating the ship dynamics. LoS algorithm is used for calculating the heading error and cross track error. 8 continuous space states are considered for the input and the discrete action space consists of 5 rudder angles. Appropriate reward function and the hyper-parameters are chosen so that the convergence of the results happens within 500-1000 episodes. The DQN algorithm is tested for course heading and path following problems in calm water, and in waves of steepness 0.02 and 0.038. A finite horizon model is followed in the DQN formulation. Transfer learning for different heading actions is performed based on the exploration threshold. A detailed sensitivity study on the effect of DQN hyperparameters on the convergence of the reward function has been conducted. Good performances are attained for both the heading control and path following problems.

The paths are discretized using waypoints. The waypoints are uniformly distributed along the path and are kept at an interval of 2× lbp. The agent is tested for vessel headings of -45,-90,-135 and -180 deg in calm water and waves. The average reward, episode duration and the heading error converge after 500s. Within the allowed limited rudder actions, the agent is able to achieve the heading control. The pre-trained weights from a particular heading is tested for other vessel headings. This resulted in a lesser training period and shorter episode duration for convergence. An S-shaped and an elliptical path are used for the path following in calm water and the agent follows both the paths smoothly. Similarly, the heading control and path following tasks are simulated in regular waves. The DQN based control is able to reject the disturbances and achieve the vessel heading irrespective of the wave directions by assuming that the vessel has adequate propeller and rudder actuation power to overcome the wave forces. The path following in waves are tested for a S-shaped curve and a cardioid curve, which lies in four quadrants. The agent's ability to track the waypoints in all the four quadrants are demonstrated. A sensitivity study on the hyperparameters such as node size, batch size, discount rate and memory size are conducted. Larger values of the node and batch size result in larger reward function. The discount parameter value between 0.7-0.95 is found to be optimal for the problem. A memory size of 8-16 time the size of the horizon yields larger reward and faster convergence. Among the different types of optimizers, the Adam optimizer gives the best performance and SGD resulted in poor performance. Thus, a data driven control based on DQN algorithm is successfully implemented in the numerical simulations for the autonomous navigation of a tanker in calm water and in waves.

This work can be further extended to Lyapunov stability based learning to ensure system stability. In addition to this, the scope can be extended to collision avoidance and cooperative control between multiple vehicles. Since it's been shown in paper that the RL controller can execute any path following tasks effectively, a proper reward function for collision avoidance may be added on the existing algorithm for safe autonomous ship navigation. The proposed method is an off-policy learning algorithm, which could be converted to an on-policy learning algorithms.

10. Acknowledgement

The collaboration with the foreign faculty is through the Scheme for Promotion of Academic and Research Collaboration (SPARC) for "the Advanced Ship Manoeuvring Prediction based on Machine Learning and Artificial Intelligence for Autonomous Ship Navigation" under the project number SPARC/2018-2019/P1181/SL. We would like to thank Dr. Awanish Dubey for conducting experiments for validating the RL algorithm in order to answer the reviewer question.

References

[1] M. Moradi, M. Katebi, Predictive pid control for ship autopilot design, IFAC Proceedings Volumes 34 (7) (2001) 375–380.

[2] M. Tomera, Fuzzy self-tuning pid controller for a ship autopilot, in: Proceedings of the 12th International Conference on Marine Navigation and Safety of Sea Transportation, 2017.

[3] H. Zheng, R. R. Negenborn, G. Lodewijks, Trajectory tracking of autonomous vessels using model predictive control, IFAC Proceedings Volumes 47 (3) (2014) 8812–8818.

[4] L. Kuczkowski, R. Smierzchalski, Path planning algorithm for ship collisions avoidance in environment with changing strategy of dynamic obstacles, Springer International Publishing, Cham, 2017, pp. 641–650.

[5] A. Lazarowska, A trajectory base method for ship's safe path planning, Procedia Computer Science 96 (2016) 1022–1031.

[6] M. Tomera, Ant colony optimization algorithm applied to ship steering control, Vol. 35, 2014, pp. 83–92. doi:10.1016/j.procs.2014.08.087.

[7] L. Moreira, T. I. Fossen, C. Guedes Soares, Path following control system for a tanker ship model, Ocean Engineering 34 (14) (2007) 2074–2085. doi:https://doi.org/10.1016/j.oceaneng.2007.02.005.

[8] S. Rajendran, P. Sadhappa, A unified seakeeping and manoeuvring model with a pid controller for path following of a kvlcc2 tanker in regular waves, Applied Ocean Research 116 (10 2021). doi:10.1016/j.apor.2021.102860.

[9] B. J. Guerreiro, C. Silvestre, R. Cunha, A. Pascoal, Trajectory tracking nonlinear model predictive control for autonomous surface craft, IEEE Transactions on Control Systems Technology 22 (6) (2014) 2160–2175.

[10] L. P. Perera, V. Ferrari, F. P. Santos, M. A. Hinostroza, C. G. Soares, Experimental evaluations on ship autonomous navigation and collision avoidance by intelligent guidance, IEEE Journal of Oceanic Engineering 40 (2) (2014) 374–387.

[11] R. Sandeepkumar, S. Rajendran, R. Mohan, A. Pascoal, A unified ship manoeuvring model with a nonlinear model predictive controller for path following in regular waves, Ocean Engineering 243 (2022) 110165. `doi:https://doi.org/10.1016/j.oceaneng.2021.110165`.

[12] A. B. Martinsen, A. M. Lekkas, S. Gros, Reinforcement learning-based nmpc for tracking control of asvs: Theory and experiments, Control Engineering Practice 120 (2022) 105024. `doi:https://doi.org/10.1016/j.conengprac.2021.105024`.

[13] N. E. Kahveci, P. A. Ioannou, Adaptive steering control for uncertain ship dynamics and stability analysis, Automatica 49 (3) (2013) 685–697.

[14] X. Zhang, G. Yang, Q. Zhang, G. Zhang, Y. Zhang, Improved concise backstepping control of course keeping for ships using nonlinear feedback technique, The Journal of Navigation 70 (6) (2017) 1401–1414.

[15] L. G. García-Valdovinos, T. Salgado-Jiménez, M. Bandala-Sánchez, L. Nava-Balanzar, R. Hernández-Alvarado, J. A. Cruz-Ledesma, Modelling, design and robust control of a remotely operated underwater vehicle, International Journal of Advanced Robotic Systems 11 (1) (2014) 1.

[16] Y. Liu, R. Bu, X. Gao, Ship trajectory tracking control system design based on sliding mode control algorithm, Polish Maritime Research (3) (2018) 26–34.

[17] R. Zhang, Y. Chen, Z. Sun, F. Sun, H. Xu, Path control of a surface ship in restricted waters using sliding mode, IEEE Transactions on Control Systems Technology 8 (4) (2000) 722–732.

[18] X. K. Dang, H. N. Truong, V. D. Do, A path planning control for a vessel dynamic positioning system based on robust adaptive fuzzy strategy, Automatika 63 (3) (2022) 580–592.

[19] T. Brcko, J. Švetak, Fuzzy reasoning as a base for collision avoidance decision support system 25 (12 2013).

[20] S. Wang, H. Yu, J. Yu, J. Na, X. Ren, Neural-network-based adaptive funnel control for servo mechanisms with unknown dead-zone, IEEE transactions on cybernetics 50 (4) (2018) 1383–1394.

[21] S. Wang, J. Na, X. Ren, Rise-based asymptotic prescribed performance tracking control of nonlinear servo mechanisms, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (12) (2017) 2359–2370.

[22] J. Na, S. Wang, Y.-J. Liu, Y. Huang, X. Ren, Finite-time convergence adaptive neural network control for nonlinear servo systems, IEEE Transactions on Cybernetics 50 (6) (2019) 2568–2579.

[23] J. Na, Y. Huang, X. Wu, S.-F. Su, G. Li, Adaptive finite-time fuzzy control of nonlinear active suspension systems with input delay, IEEE Transactions on Cybernetics 50 (6) (2019) 2639–2650.

[24] M. L. Puterman, Markov decision processes: discrete stochastic dynamic programming, John Wiley & Sons, 2014.

[25] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, I. Palunko, Reinforcement learning for control: Performance, stability, and deep approximators, Annual Reviews in Control 46 (2018) 8–28.

[26] B. Yoo, J. Kim, Path optimization for marine vehicles in ocean currents using reinforcement learning, Journal of Marine Science and Technology 21 (2) (2016) 334–343.

[27] C. Wang, X. Zhang, R. Li, P. Dong, Path planning of maritime autonomous surface ships in unknown environment with reinforcement learn-

ing, in: International Conference on Cognitive Systems and Signal Processing, Springer, 2018, pp. 127–137.

[28] S. Guo, X. Zhang, Y. Zheng, Y. Du, An autonomous path planning model for unmanned ships based on deep reinforcement learning, Sensors 20 (2) (2020) 426.

[29] E. ARTUSI, Ship path planning based on deep reinforcement learning and weather forecast, in: 2021 22nd IEEE International Conference on Mobile Data Management (MDM), 2021, pp. 258–260. doi:10.1109/MDM52706.2021.00052.

[30] C. Wang, X. Zhang, R. Li, P. Dong, Path Planning of Maritime Autonomous Surface Ships in Unknown Environment with Reinforcement Learning, 2019, pp. 127–137. doi:10.1007/978-981-13-7986-4_12.

[31] A. B. Martinsen, A. M. Lekkas, S. Gros, J. A. Glomsrud, T. A. Pedersen, Reinforcement learning-based tracking control of usvs in varying operational conditions, Frontiers in Robotics and AI 7 (2020) 32. doi:10.3389/frobt.2020.00032.

[32] S. Xie, X. Chu, M. Zheng, C. Liu, A composite learning method for multi-ship collision avoidance based on reinforcement learning and inverse control, Neurocomputing 411 (06 2020). doi:10.1016/j.neucom.2020.05.089.

[33] Z. Luman, M.-I. Roh, S.-J. Lee, Control method for path following and collision avoidance of autonomous ship based on deep reinforcement learning (2019).

[34] A. B. Martinsen, A. M. Lekkas, Straight-path following for underactuated marine vessels using deep reinforcement learning, IFAC-PapersOnLine 51 (29) (2018) 329–334, 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018. doi:https://doi.org/10.1016/j.ifacol.2018.09.502.

[35] A. B. Martinsen, A. M. Lekkas, Curved path following with deep reinforcement learning: Results from three vessel models, in: OCEANS 2018 MTS/IEEE Charleston, IEEE, 2018, pp. 1–8.

[36] E. Smirnova, E. Dohmatob, On the convergence of approximate and regularized policy iteration schemes (2019). `arXiv:1909.09621`.

[37] Q. Wei, D. Liu, Q. Lin, R. Song, Discrete-time optimal control via local policy iteration adaptive dynamic programming, IEEE Transactions on Cybernetics 47 (10) (2017) 3367–3379. `doi:10.1109/TCYB.2016.2586082`.

[38] S. Ohnishi, E. Uchibe, Y. Yamaguchi, K. Nakanishi, Y. Yasui, S. Ishii, Constrained deep q-learning gradually approaching ordinary q-learning, Frontiers in Neurorobotics 13 (2019). `doi:10.3389/fnbot.2019.00103`.

[39] C. Wang, X. Zhang, R. Li, P. Dong, Path planning of maritime autonomous surface ships in unknown environment with reinforcement learning (2019) 127–137.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. A. Riedmiller, Playing atari with deep reinforcement learning, CoRR abs/1312.5602 (2013). `arXiv:1312.5602`.

[41] R. Agarwal, D. Schuurmans, M. Norouzi, An optimistic perspective on offline reinforcement learning (2020).

[42] W. M. Kouw, M. Loog, An introduction to domain adaptation and transfer learning (2019). `arXiv:1812.11806`.

[43] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning (2020). `arXiv:1911.02685`.

[44] H. Yasukawa, Y. Yoshimura, Introduction of mmg standard method for ship maneuvering predictions, Journal of Marine Science and Technology 20 (1) (2015) 37–52.

[45] N. SALVESEN, Second-order steady-state forces and moments on surface ships in oblique regular waves, Int Symp.on the Dynamics of Marine Vehicles and Structures in Waves (1974).
URL https://ci.nii.ac.jp/naid/20000368491/en/

[46] D. J. Kim, K. Yun, J.-Y. Park, D. J. Yeo, Y. G. Kim, Experimental investigation on turning characteristics of kvlcc2 tanker in regular waves, Ocean Engineering 175 (2019) 197–206.

[47] A. M. Lekkas, T. I. Fossen, Line-of-sight guidance for path following of marine vehicles, Advanced in marine robotics (2013) 63–92.

[48] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, 2nd Edition, The MIT Press, 2018.
URL http://incompleteideas.net/book/the-book-2nd.html

[49] P. Baldi, Gradient descent learning algorithm overview: General dynamical systems perspective, Neural Networks, IEEE Transactions on 6 (1995) 182 – 195. doi:10.1109/72.363438.

[50] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2017). arXiv:1412.6980.

[51] R. Liu, J. Zou, The effects of memory replay in reinforcement learning, CoRR abs/1710.06574 (2017). arXiv:1710.06574.

[52] A. C. Dubey, R. Gajapathy, S. Rajendran, An experimental and numerical investigation on the autopilot design of a kvlcc2 tanker, in: OCEANS 2021: San Diego–Porto, IEEE, 2021, pp. 1–6.

[53] M. D. Zeiler, ADADELTA: an adaptive learning rate method, CoRR abs/1212.5701 (2012). arXiv:1212.5701.

[54] R. Ward, X. Wu, L. Bottou, Adagrad stepsizes: Sharp convergence over nonconvex landscapes (2021). arXiv:1806.01811.

[55] S. Zheng, Q. Meng, T. Wang, W. Chen, N. Yu, Z.-M. Ma, T.-Y. Liu, Asynchronous stochastic gradient descent with delay compensation (2020). `arXiv:1609.08326`.

[56] S. De, A. Mukherjee, E. Ullah, Convergence guarantees for rmsprop and adam in non-convex optimization and an empirical comparison to nesterov acceleration (2018). `arXiv:1807.06766`.

[57] A. Mosca, G. D. Magoulas, Adapting resilient propagation for deep learning (2015). `arXiv:1509.04612`.

[58] A. Jentzen, A. Riekert, A proof of convergence for stochastic gradient descent in the training of artificial neural networks with relu activation for constant target functions (2021). `arXiv:2104.00277`.