

DECOUPLED DEEP REINFORCEMENT LEARNING FOR AUTONOMOUS LANDING OF UAVS ON SHIPS

Truman Gerner[1], Michael Candon[1], Amirali Khodadadian Gostar[1], Nirman Jayarathne[2], James Venning[3], Martin Saska[4] & Pier Marzocca[1]

[1]RMIT University, Melbourne, AU
[2]Navantia Australia, Melbourne, AU
[3]University of Tasmania, Hobart, AU
[4]Czech Technical University in Prague, Prague, CZ

Abstract

Autonomous landing of uncrewed aerial vehicles (UAVs) on ships poses a significant challenge due to wind disturbances and wave-driven vessel motion, which continuously perturb the landing target. While deep reinforcement learning (DRL) offers adaptive, model-free control, monolithic policies must jointly handle coupled horizontal positioning and heave-compensated descent, leading to high-dimensional learning problems and complex training environments. This paper proposes a decoupled actor–critic DRL framework in which two agents are trained for complementary subtasks: a horizontal controller that tracks the moving deck under randomised wind and target drift, and a vertical controller that regulates descent by compensating ship heave through commanded vertical velocity. The approach is evaluated in a Gazebo–ROS–PX4 software-in-the-loop simulation environment that includes variable wind and wave parameters. Trial results demonstrate robust disturbance rejection, achieving mean horizontal landing errors below 0.2 m across most wind–wave conditions, with a worst-case error of 0.31 m. The vertical controller achieves touchdown velocities below 1 m/s for sea states up to 4, corresponding to the majority of real-world maritime conditions. Compared with a single-agent controller, the proposed decoupled approach reduces horizontal landing error by up to an order of magnitude, and significantly improves landing reliability, enabling consistently safe autonomous landings across a range of wind and wave conditions.

Keywords: Deep Reinforcement Learning, Machine Learning, UAV, Maritime, Mechanics and Control

1. Introduction

In recent years, the operation of Uncrewed Aerial Vehicles (UAVs) in the maritime sector has increased substantially. Their versatility, cost-effectiveness, and ability to mitigate risks by performing hazardous tasks have made them a valuable asset in this domain. UAVs have been widely adopted for a range of applications, including environmental monitoring and remote sensing [1], search and rescue operations [2], emission monitoring [3], maritime communication [4], and logistics [5]. Safe landing of UAVs on vessels remains a key challenge in maritime operations, particularly during remote deployments such as offshore missions, where limited battery endurance necessitates reliable landing and recharging capabilities.

Unlike static landing platforms, maritime environments introduce complex, dynamic conditions that significantly increase the difficulty and risks during landing. Unsteady hydrodynamic effects on the vessel alter the landing zone's attitude and position, making tracking more difficult. The motion also increases the risk of damaging the UAV as it approaches the landing deck, requiring sensitive alignment between the UAV and the landing platform near touchdown. Marine winds introduce an additional disturbance that can significantly affect the aircraft's state during landing. Although piloted

control has been successfully used for such operations, this limits scalability and demands the presence of skilled personnel. A fully autonomous method is the long-term solution, but it requires a robust control scheme capable of handling highly variable landing conditions.

Reinforcement Learning (RL) has emerged as a rapidly growing area of robotics research, with significant applications in UAV control. Numerous studies have applied RL to UAV landing tasks, addressing both static and dynamic platforms in controlled environments [6]. However, few works address the specific challenges of maritime landings. One reason for this is the highly nonlinear and stochastic nature of the environment. RL algorithms require substantial amounts of data to learn effective policies, and this demand is further amplified in dynamic environments with rapidly changing disturbances such as wind and waves. Designing a comprehensive reward function for this task often leads to inefficient, intractable learning, thereby preventing the development of a robust control policy. Furthermore, overfitting can limit a learned policy's adaptability to variable environmental conditions, such as those encountered in the maritime landing task.

This paper proposes a novel RL control method for autonomous UAV landing on an Uncrewed Surface Vehicle (USV) by dividing control authority between two agents, leveraging decoupled learning to maximise computational tractability and overall performance. One agent is trained to control horizontal position above the landing target in the presence of wind disturbances, while the other is trained to control vertical position, compensating for the vessel's heave motion induced by waves. This separation of control authority enables specialised training environments, producing policies that operate in parallel to achieve autonomous landings on ships.

2. Related Work

Autonomous UAV maritime landings have traditionally been addressed using model-based and perception driven control strategies. Complementary techniques such as visual-servoing [7, 8, 9] and motion prediction algorithms [10, 11, 12] have been explored; however, model predictive control (MPC) has emerged as the dominant method due to its ability to explicitly account for system dynamics, operational constraints, and future platform motion during trajectory optimisation. The state-of-the-art in maritime UAV landing is presented by Procházka et al. [13], who propose an MPC-based trajectory generation framework that exploits predicted USV states derived from onboard sensor fusion and inter-vehicle communication. These predictions are informed by a 6-DOF hydrodynamic USV model augmented with wave dynamics, enabling more accurate estimation of vessel motion under realistic sea conditions. This allows continuously updated trajectories that account for both UAV dynamics and complex USV deck motion, while landing phase-dependent adaptation of the cost function provides precise, attitude-synchronised touchdown on a potentially tilting deck.

Classic approaches like this demonstrate strong performance in ideal conditions, but remain dependent on explicit dynamical models and finite-horizon predictions, which can limit robustness in highly stochastic maritime conditions characterised by complex and rapidly varying wave–vehicle interactions as well as wind.

In contrast, RL provides a data-driven framework for learning control policies directly from interaction, capturing nonlinear disturbance effects without explicit modelling, although its effectiveness relies heavily on sufficient training diversity. While RL has already been applied to UAV landing problems, relatively few works address maritime platforms specifically. Early studies typically assume static landing surfaces [14, 15, 16], whereas more recent work has begun to consider moving targets.

For instance, Wu et al. [17] combine a standard PID module with a DRL agent to automatically tune PID gains for velocity control, enabling smooth and adaptive landings on both static and dynamic platforms. By randomising platform trajectories during training, the agent learns a robust policy that generalises to varying environmental conditions. Xie et al. [18] propose a decoupled framework in which separate models handle horizontal tracking and vertical descent. While the tracking model uses DRL, the descent phase relies on heuristic rules because the landing target moves only horizontally. Similarly, Rodriguez et al. [19] implement a Deep Deterministic Policy Gradient (DDPG) agent for horizontal position and velocity control using a Gazebo-based RL framework. However, vertical descent is not included in the action space; a constant velocity is commanded along the z-axis, simplifying the problem and accelerating convergence. This highlights a broader trend in RL-based UAV landing research: autonomous control typically handles horizontal positioning and perception,

while vertical descent and final touchdown are still governed by conventional or heuristic controllers. Although this is sufficient in many scenarios, maritime landings involve wave-induced heave, which demands a more robust and adaptive strategy for vertical descent.
Ali et al. [20] apply RL specifically to vertical descent and touchdown, while horizontal positioning and alignment are managed by a model-based controller. Their results demonstrate that policy-based DRL agents, particularly Proximal Policy Optimisation (PPO), can learn smooth and efficient landing strategies under wave-induced heave, achieving lower impact velocities than value-based agents. However, as with most RL studies in this domain, wind disturbances are not considered. Ma et al. [21] introduce a hybrid PID–DRL controller for UAV trajectory tracking. In their approach, a dual-loop PID stabilises position and attitude, while a DRL compensator, trained in staged wind-perturbed simulations, dynamically mitigates disturbances. The system uses realistic fractal-based composite wind models to ensure robust performance under varying conditions, but is applied to general flight tasks rather than landing scenarios. Lee et al. [22] employ a vision-based DRL model inspired by naval helicopter operations for ship landings under windy conditions. However, their method does not account for wave-induced heave, and the simulated and experimental winds are of constant speed and direction, limiting applicability to realistic maritime environments.
This work investigates the role of control structure in reinforcement learning for autonomous UAV ship landings under maritime disturbances. The main contributions are:

- A decoupled control architecture for disturbance-specific learning, in which separate training environments are used to isolate and address wind and wave-induced effects, enabling targeted policy development.
- A demonstration that control decoupling reduces learning complexity and improves landing reliability, shown through comparative training and evaluation across coupled and decoupled formulations.
- Insights into disturbance-specific policy learning, highlighting how isolating environmental effects influences policy behaviour, generalisation, and robustness in maritime landing scenarios.

3. Proposed Method

The UAV ship-landing problem is formulated as a Markov Decision Process (MDP) and solved using deep reinforcement learning (DRL). Figure 1 shows the overall architecture. Actor-critic algorithms are employed for their ability to handle continuous state and action spaces, providing fine adjustments essential for stable flight during complex manoeuvres such as landing on a moving ship. Both agents are trained using a DDPG algorithm, with added experience replay to enhance training stability.
The actor and critic networks are implemented as fully connected feedforward neural networks with two hidden layers of 400 and 300 units, respectively. Each hidden layer applies layer normalisation followed by a rectified linear unit (ReLU) activation function. The output layer uses a $\tanh$ activation, which is then linearly scaled to the environment's action bounds. Network parameters are optimised using the Adam optimiser. Weights of the hidden layers are initialised from a uniform distribution proportional to the inverse square root of the layer width, while the output layer weights are initialised in the range $[-3\times10^{-3}, 3\times10^{-3}]$ to stabilise early training.
Exploration noise is generated using an Ornstein–Uhlenbeck process with parameters: mean $\mu = 0$, volatility $\sigma = 0.15$, mean reversion $\Theta = 0.2$, time step $dt = 0.05$, and initial state $x_0 = 0$. The other hyperparameters selected for all agents are listed in Table 1.

3.1 Vertical controller

Both controllers use a state representation augmented with a fixed history of past observations to improve learning performance. Relative vehicle states are taken from the NED (North East Down) reference frame (see Figure 2) with

$$\Delta \boldsymbol{p}^{\mathcal{W}} = \boldsymbol{p}^{\mathcal{W}}_{\mathrm{UAV}} - \boldsymbol{p}^{\mathcal{W}}_{\mathrm{USV}} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{1}$$

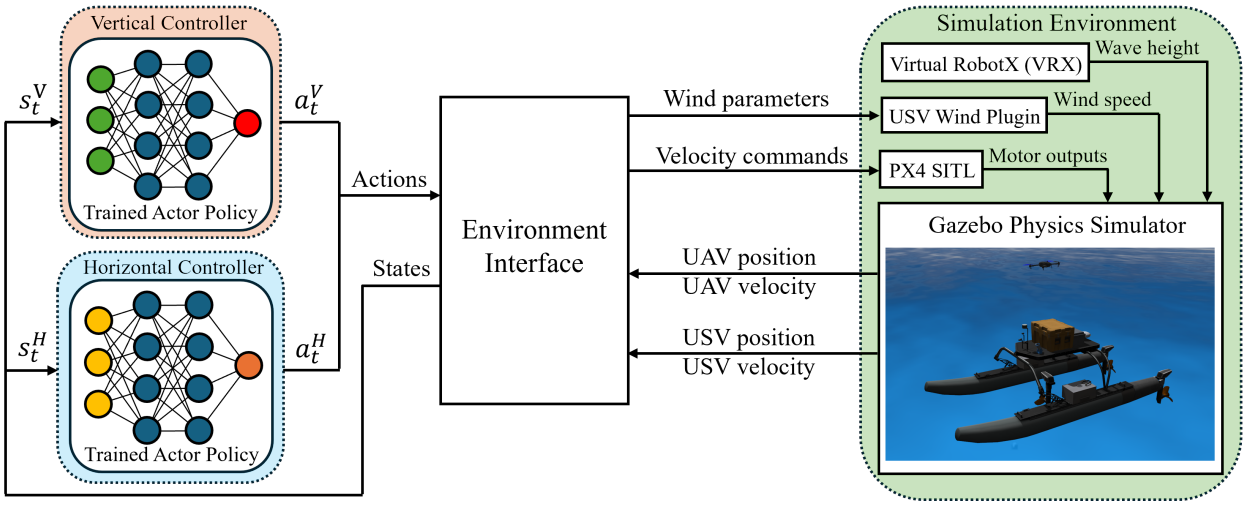


Figure 1 – Overview of the UAV control pipeline. The vertical and horizontal controllers receive state observations and output actions according to their trained policies. These actions are passed to the environment interface, where they are converted into velocity commands and transmitted to PX4. PX4 processes the commands through its cascaded control architecture to generate motor outputs, driving the simulated vehicle. Updated state observations are then returned to the controllers at the next time step. Wind disturbances are generated at each time step within the environment interface and applied via the USV Wind plugin, while wave parameters are predefined and applied through the Virtual RobotX plugins to generate wave dynamics in the virtual environment.

$$\Delta \dot{\boldsymbol{p}}^{\mathcal{W}} = \dot{\boldsymbol{p}}^{\mathcal{W}}_{\text{UAV}} - \dot{\boldsymbol{p}}^{\mathcal{W}}_{\text{USV}} = \begin{bmatrix} \Delta\dot{x} \\ \Delta\dot{y} \\ \Delta\dot{z} \end{bmatrix} \tag{2}$$

For the vertical controller, the state s^V at time step n is defined as

$$s_n^V = (\mathbf{o}_n^V, \mathbf{o}_{n-1}^V, \mathbf{o}_{n-2}^V, \mathbf{o}_{n-3}^V, \mathbf{o}_{n-4}^V, \mathbf{o}_{n-5}^V) \tag{3}$$

with

$$\mathbf{o}_n^V = (\Delta z_n, \Delta \dot{z}_n, \Delta\phi_n, \Delta\theta_n). \tag{4}$$

where Δz_n and $\Delta \dot{z}_n$ denote the relative vertical position and velocity between the two vehicles at time step n, and $\Delta\phi_n$ and $\Delta\theta_n$ denote the relative roll and pitch angles.
The action space is defined as a bounded continuous space:

$$a^V \equiv \dot{z}_{\text{cmd}} \in [-0.3,\, 0.3]\ \text{m/s} \tag{5}$$

where $\dot{z}_{\text{cmd}}$ is the vertical command velocity sent to the autopilot firmware.
To aid learning, a shaping function determines the reward. The shaping function seeks to minimise vertical offset, relative velocity and command velocity to the flight controller, and is defined as:

Table 1 – Hyperparameters For Both Agents

Parameter	**Value**
Learning rate (α)	1.0×10^{-4}
Target smoothing (β)	1.0×10^{-3}
Discount factor (γ)	0.99
Soft update rate (τ)	1.0×10^{-3}
Batch size	64
Replay buffer size	1.0×10^{6}

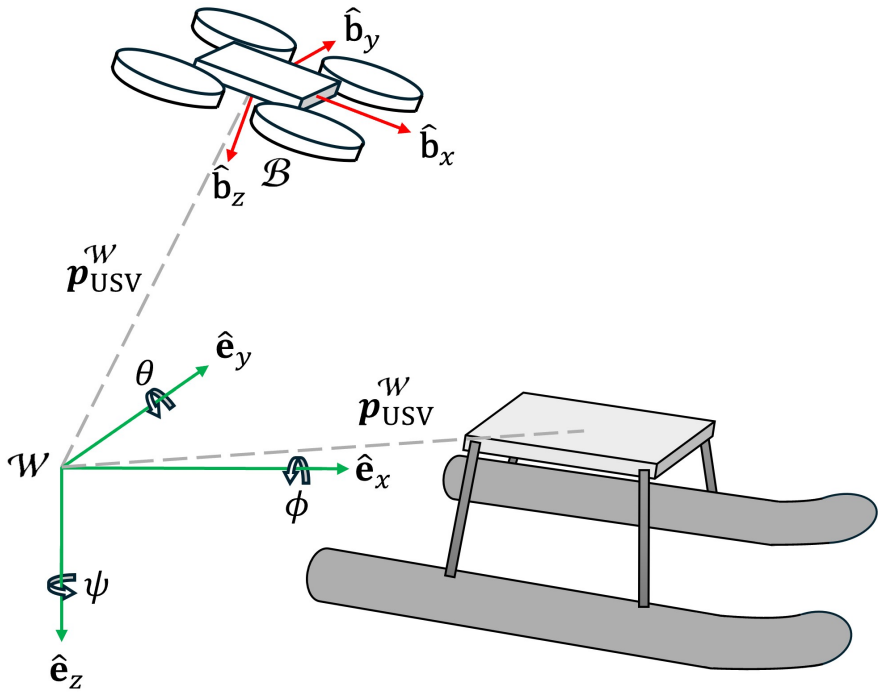


Figure 2 – Illustration of the coordinate system, where $\mathcal{W}$ represents the local North East Down (NED) frame and $\mathcal{B}$ the UAV body frame. $\boldsymbol{p}^{\mathcal{W}}_{\text{UAV}} = [x,\, y,\, z]^{\text{T}}$ and $\boldsymbol{p}^{\mathcal{W}}_{\text{USV}}$ are the translation vectors of the UAV and USV, respectively.

$$\Phi^V(s_n, a_n) = -\beta_1|\Delta z_n| - \beta_2|\Delta \dot{z}_n| - \beta_3|\dot{z}_{\text{cmd,n}}| \tag{6}$$

where β_1, β_2, and β_3 are positive weighting constants, with values $\beta_1 = 90$, $\beta_2 = 60$, and $\beta_3 = 5$. The reward at time step n is computed as

$$r_n^V = \Phi^V(s_n, a_n) - \Phi^V(s_{n-1}, a_{n-1}) + R_{\text{land}} \tag{7}$$

where R_{land} is defined as

$$R_{\text{land}} = \begin{cases} +50, & \text{if } |\Delta z_n| < \epsilon \cap |\Delta \dot{z}_n| < v_1, \\ -100, & \text{if } |\Delta z_n| < \epsilon \cap |\Delta \dot{z}_n| > v_2, \\ 50\left(1 - \dfrac{|\Delta \dot{z}_n| - v_1}{v_2 - v_1}\right), & \text{if } |\Delta z_n| < \epsilon \cap v_1 \leq |\Delta \dot{z}_n| \leq v_2, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

with $\epsilon = 0.05$ m, $v_1 = 0.5$ m/s, $v_2 = 1.0$ m/s.
By structuring the reward function to apply constant penalties throughout an episode, the agent is incentivised to complete the task as quickly as possible. R_{land} is added to penalise high-impact landings and reward softer touchdowns, discouraging the agent from learning a reckless policy for landing.

3.2 Horizontal controller

For the horizontal controller, the state s^H at time step t is defined as

$$s_n^H = (\mathbf{o}_n^H, \mathbf{o}_{n-1}^H, \mathbf{o}_{n-2}^H, \mathbf{o}_{n-3}^H, \mathbf{o}_{n-4}^H, \mathbf{o}_{n-5}^H) \tag{9}$$

with

$$\mathbf{o}_n^H = (\Delta x_n, \Delta y_n, \Delta z_n,\, \Delta \dot{x}_n, \Delta \dot{y}_n,\, \Delta \phi_n, \Delta \theta_n). \tag{10}$$

where $\Delta x_n, \Delta y_n, \Delta z_n$ is the relative position, $\Delta \dot{x}_n, \Delta \dot{y}_n$ are the relative horizontal velocities, and $\Delta \phi_n, \Delta \theta_n$ are the relative roll and pitch.
The continuous action space is

$$a^H \equiv (\dot{x}_{\text{cmd}}, \dot{y}_{\text{cmd}}) \tag{11}$$

where

$$\dot{x}_{\text{cmd}}, \dot{y}_{\text{cmd}} \in [-3,\ 3]\ \text{m/s} \tag{12}$$

are the horizontal command velocities.
The shaping function for horizontal motion is defined as

$$\begin{aligned}\Phi^H(s_n, a_n) = &-\beta_4|\Delta x_n| - \beta_5|\Delta \dot{x}_n| - \beta_6|\dot{x}_{\text{cmd},n}| \\ &- \beta_7|\Delta y_n| - \beta_8|\Delta \dot{y}_n| - \beta_9|\dot{y}_{\text{cmd},n}|\end{aligned} \tag{13}$$

The reward function for the horizontal controller at time step n, r_n^H is identical to that used for the vertical controller (7) but with the horizontal shaping function (13) and no landing reward R_{land}.

4. Simulation Environment and Training

The simulation environment used in this study integrates Gazebo with Robot Operating System (ROS). The UAV model is controlled by the PX4 software-in-the-loop (SITL) autopilot via MAVLink to emulate realistic flight dynamics. ROS acts as the middleware, handling state data and control interfaces between PX4 and external nodes. A custom OpenAI Gym-compatible environment wraps this system, enabling the RL agent to interact with the drone via velocity commands and to receive state observations. The UAV states are obtained from PX4's onboard state estimator, which fuses IMU, GNSS, compass, and barometer using an Extended Kalman Filter (EKF). The WAM-V states are taken as ground truth directly from the simulation environment.
This setup enables training of RL policies in a high-fidelity physics-based simulation, with real-time communication among Gazebo, ROS, and PX4. Training and testing are performed using the default 3DR Iris quadrotor 3.6 kg and the WAM-V unmanned surface vessel, with a landing deck measuring 1.02 m wide by 1.84 m long.

4.1 Wave Model

Waves are simulated using the Virtual RobotX (VRX) plugin [23], which generates irregular ocean waves by sampling a parametric wave spectrum. The simulator employs a two-parameter form of the Pierson–Moskowitz (PM) spectrum [24], representing a fully developed sea in deep water.
The spectrum is parameterised by the peak angular frequency ω_p, which determines the dominant wave period T_p according to

$$\omega_p = \frac{2\pi}{T_p} \tag{14}$$

and a user-specified significant wave height $\bar{H}_s$, representing the average height of the highest one-third of waves. This value is related to the PM significant wave height H_s through a gain factor

$$K_H = \frac{\bar{H}_s}{H_s} \tag{15}$$

which scales the spectral energy.
The spectral density function $S_B(\omega)$ describing the distribution of wave energy across angular frequencies ω is given by

$$S_B(\omega) = K_H^2\, \alpha \frac{g^2}{\omega^5} \exp\left[-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^4\right] \tag{16}$$

where α is the Phillips constant (0.0081) and g is gravitational acceleration.

A finite set of sinusoidal wave components is sampled from the spectrum, with amplitudes

$$A_i = \sqrt{2S_B(\omega_i)\,\Delta\omega_i} \tag{17}$$

where ω_i are the sampled angular frequencies and $\Delta\omega_i$ are the corresponding frequency intervals used in the spectral discretisation.
The resulting free-surface elevation is obtained through superposition of these components as

$$\eta(x,t) = \sum_{i=1}^{N} A_i \cos\left(k_i x - \omega_i t + \delta_i\right) \tag{18}$$

where k_i is the wavenumber determined by the deep-water dispersion relation

$$\omega_i^2 = g k_i \tag{19}$$

and δ_i is a random phase.
This approach provides a simple, physically consistent, and user-customisable method for generating sinusoidal wave fields for maritime simulations.

4.2 Wind and Turbulence Model

Wind disturbances are modeled using a stochastic Dryden-inspired turbulence model. The turbulent wind components u, v, and w are generated using first-order shaping filters driven by Gaussian white noise. The time constants of the filters are derived from Dryden turbulence length scales and the mean wind speed U,

$$T_u = \frac{L_u}{U}, \qquad T_v = \frac{L_v}{U}, \qquad T_w = \frac{L_w}{U} \tag{20}$$

where L_u, L_v, and L_w are the turbulence length scales in the longitudinal, lateral, and vertical directions.
The turbulent velocity components evolve according to the discrete stochastic update

$$u_k = \left(1 - \frac{\Delta t}{T_u}\right) u_{k-1} + \sigma_u \sqrt{\frac{2\Delta t}{T_u}}\,\xi_u \tag{21}$$

$$v_k = \left(1 - \frac{\Delta t}{T_v}\right) v_{k-1} + \sigma_v \sqrt{\frac{2\Delta t}{T_v}}\,\xi_v \tag{22}$$

$$w_k = \left(1 - \frac{\Delta t}{T_w}\right) w_{k-1} + \sigma_w \sqrt{\frac{2\Delta t}{T_w}}\,\xi_w \tag{23}$$

where Δt is the simulation timestep, σ_u, σ_v, and σ_w are turbulence intensity parameters, and ξ_u, ξ_v, and ξ_w are independent Gaussian random variables with zero mean and unit variance.
The mean wind vector is added to the turbulent components to produce the final wind field used in the simulation. While this does not reproduce the exact Dryden spectral model, it preserves the characteristic correlation structure of atmospheric turbulence and provides a computationally efficient disturbance model for RL training.

4.3 Sea States and Beaufort Scale

There are several methods for classifying the severity of a maritime environment. Most commonly, sea state classifications are used, based on the Douglas Sea Scale [25], which characterises wave conditions according to significant wave height and wave period. The World Meteorological Organization (WMO) formalises these classifications through its sea state code [26], summarised in Table 2. However, sea-state classifications are limited to water conditions and do not account for the corresponding wind speeds. Consequently, in this study, wind speeds are classified using the Beaufort scale [27] as seen in Table 3.

Table 2 – Sea state classifications based on the WMO sea state code

Sea State	Significant Wave Height (m)	Description
0	0	Calm (glassy)
1	0 – 0.1	Calm (rippled)
2	0.1 – 0.5	Smooth (wavelets)
3	0.5 – 1.25	Slight
4	1.25 – 2.5	Moderate
5	2.5 – 4.0	Rough
6	4.0 – 6.0	Very rough
7	6.0 – 9.0	High
8	9.0 – 14.0	Very high
9	>14.0	Phenomenal

Table 3 – Beaufort scale for wind classification

Beaufort No.	Wind Speed (m/s)	Description
0	<0.5	Calm
1	0.5 – 1.5	Light air
2	1.6 – 3.3	Light breeze
3	3.4 – 5.4	Gentle breeze
4	5.5 – 7.9	Moderate breeze
5	8.0 – 10.7	Fresh breeze
6	10.8 – 13.8	Strong breeze
7	13.9 – 17.1	Near gale
8	17.2 – 20.7	Gale
9	20.8 – 24.4	Strong gale
10	24.5 – 28.4	Storm
11	28.5 – 32.6	Violent storm
12	>32.6	Hurricane

4.4 Training

Each controller is trained independently in a unique environment designed to learn simple and effective policies tailored to its control task. Each controller trains for 1000 episodes of varying length as described below.

The horizontal controller is trained in a windy environment, with the USV drifting due to wave-induced motion. At the start of each episode, a mean wind speed and direction is randomly selected in the ranges $U \in [0, 4.5]$ m/s, $\psi \in [0, 360°]$, and $\theta \in [-10°, 10°]$ and turbulence intensities: $\sigma_u = \sigma_v = 2.12, \sigma_w = 1.4$ to produce sufficient variance to train adaptable policies. The UAV begins the episode hovering at a fixed altitude and randomised horizontal error from the USV in the range $x \in [-5, 5]$ m, $y \in [-5, 5]$ m to avoid overfitting to one approach path. The episode terminates after 12 seconds.

The vertical controller is trained with the UAV positioned at an initial altitude of 10 m above the deck in sea state 3. During training, the vessel undergoes wave-induced heave motion, and the UAV must descend while regulating relative vertical velocity. An episode terminates when the UAV reaches the predefined landing altitude, 0.05 meters above the deck. If this condition is not satisfied, the episode terminates after a maximum duration of 45 seconds. Each episode begins at a random point in wave motion to avoid overfitting to a single condition.

5. Results and Discussion

For each combination of sea states 0–5 and Beaufort numbers 1–5, both controllers are evaluated in parallel over 100 episodes of the autonomous ship landing task. The episode architecture is identical to that used to train the vertical controller, with termination occurring only after reaching a fixed height above the USV's deck or after a timeout. The results are presented in Tables 4 and 5.

Table 4 – Mean Horizontal Landing Error Magnitude (m)

Sea State	**Beaufort Number**						σ
	0	1	2	3	4	5	
0	0.125	0.115	0.118	0.127	0.131	0.123	0.064
1	0.130	0.117	0.122	0.115	0.121	0.121	0.062
2	0.146	0.146	0.142	0.120	0.183	0.105	0.064
3	0.131	0.131	0.126	0.124	0.122	0.105	0.067
4	0.120	0.160	0.129	0.114	0.122	0.180	0.081
5	0.168	0.169	0.173	0.198	0.185	0.308	0.115

Mean error across trials; σ denotes the representative standard deviation for each sea state.

Table 5 – Mean Magnitude of Vertical Relative Landing Velocity (m/s)

Sea State	**Beaufort Number**						σ
	0	1	2	3	4	5	
0	0.252	0.242	0.255	0.272	0.314	0.306	0.075
1	0.260	0.245	0.257	0.276	0.305	0.322	0.080
2	0.309	0.297	0.308	0.343	0.362	0.397	0.130
3	0.494	0.465	0.480	0.487	0.489	0.491	0.169
4	0.688	0.665	0.670	0.675	0.713	0.755	0.235
5	1.207	1.274	1.378	1.201	1.285	1.353	0.477

Mean error across trials; σ denotes the representative standard deviation for each sea state.

Results demonstrate robust horizontal tracking across all test conditions, with mean landing deviations from the target remaining below 20 cm except for the worst case. The controller develops a policy that effectively mitigates wind disturbances, adapting to sudden changes in wind speed while maintaining precise tracking of the USV's horizontal drift, as illustrated in Figure 5. The vertical controller achieves soft landings for sea states up to 4, with a mean vertical landing velocity less than $1\ \mathrm{m/s}$, well within safe thresholds. At sea state 5, the significant motion of the USV impairs the controller's velocity-tracking performance, thereby elevating the likelihood of UAV damage upon landing. According to [28], sea states 0–4 account for approximately 83% of conditions worldwide, indicating that the operating window of this method encompasses the vast majority of real-world scenarios, and even in more extreme conditions, landing remains possible but at increased risk.

During trials, vertical landing velocity increases with stronger wind conditions, highlighting a correlation between wind speed and touchdown rate, whereas the horizontal error shows little sensitivity to changes in sea state. This is likely due to the more static conditions in the vertical controller's training environment compared to those in the horizontal controller's. Without sufficient domain randomisation, it is unable to adapt to environmental changes, showing overfitting to its training conditions.

For comparison, a single-agent method is trained for 1000 episodes that combines the state spaces, action spaces, and reward functions of the horizontal and vertical agents. Over 100 trial landings, when sea state and Beaufort number were matching at 0, 3, and 5, the single-agent method produces landing errors of 0.72 m, 0.74 m, and 1.61 m respectively, which are approximately five times higher than the decoupled method. The vertical landing speed differentials were 0.2 m/s, 0.49 m/s, and 0.75 m/s respectively, which are lower than the decoupled method. However, the single-agent takes approximately three times longer to land, as the agent fails to learn when it can safely descend (see Figure 3), instead hovering above the platform. This is especially evident when compared to the decoupled method's descent shown in Figure 4, which limits variation in control inputs to the touchdown stage. Furthermore, the horizontal landing error is so significant that the majority of test trials landed outside the bounds of the USV's landing deck. This highlights the benefit of separating horizontal and vertical control: a single-agent approach cannot effectively learn a policy to complete the task because of the intractability of the combined state-action space and the complexity of its reward function.

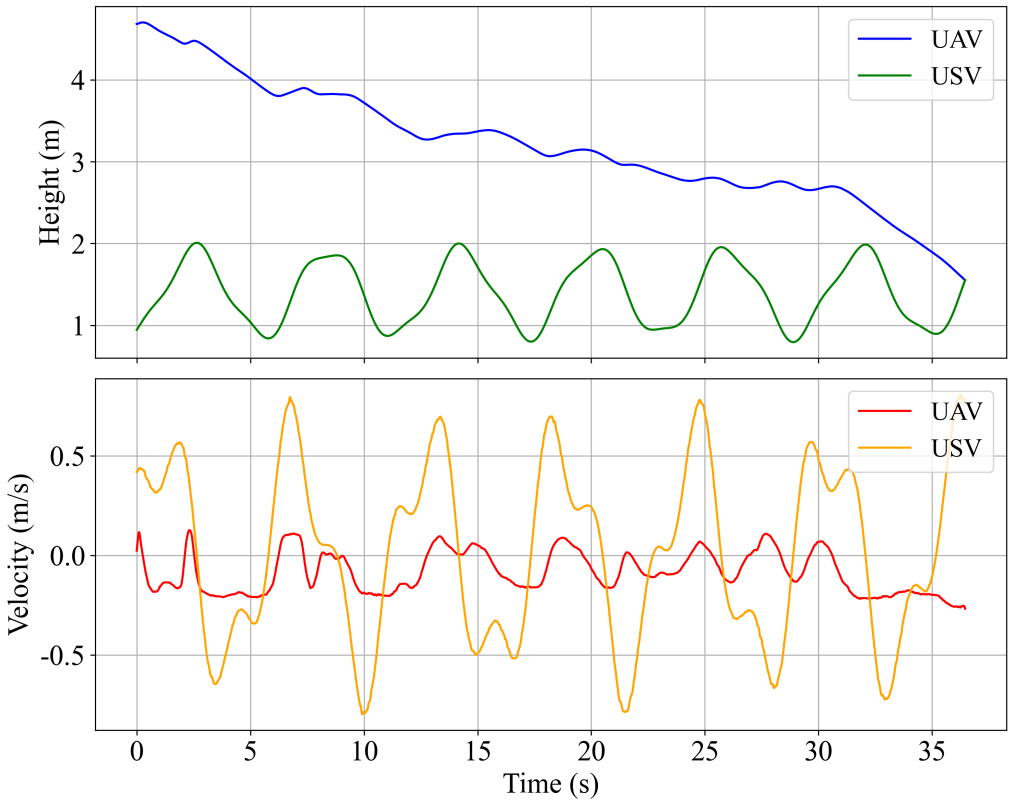


Figure 3 – Single agent method sample landing showing timeseries of the UAV and USV vertical position (top) and vertical velocity (bottom). Sea state = 3.

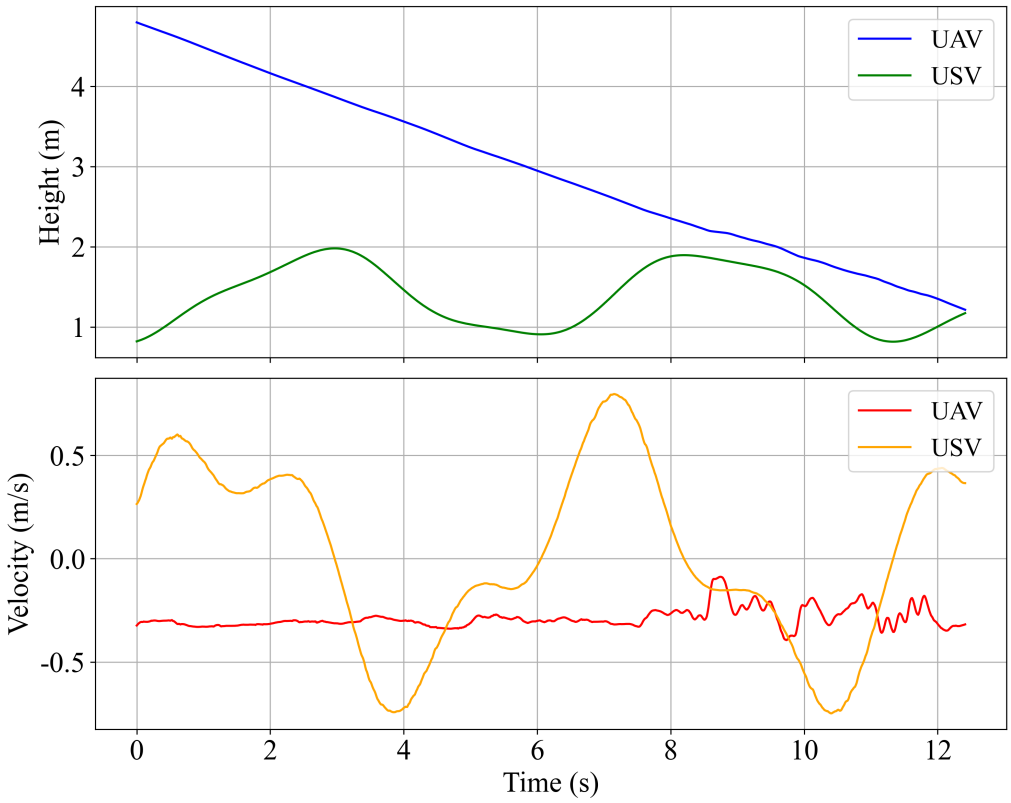


Figure 4 – Decoupled agents method sample landing trial showing timeseries of the UAV and USV vertical position (top) and vertical velocity (bottom). Sea state = 3.

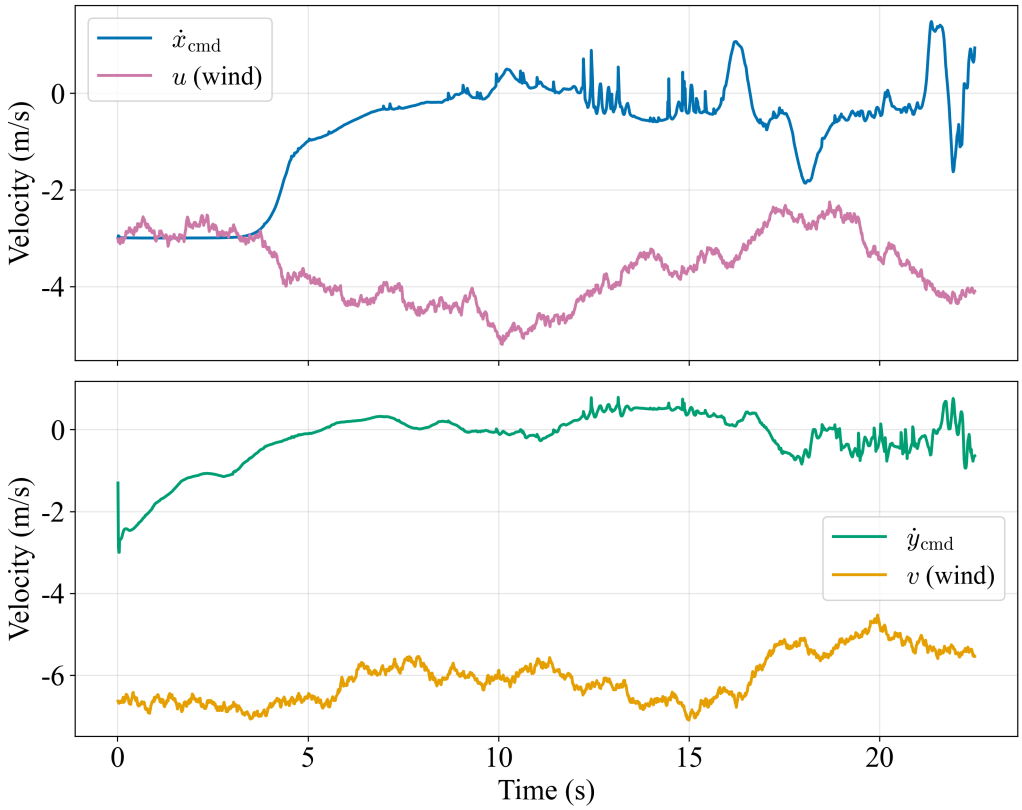


Figure 5 – Sample landing trial showing the effectiveness of the decoupled agents method at Beaufort Number 5. Timeseries of both command and wind velocities are presented in both the lateral (top) and longitudinal (bottom) directions.

6. Conclusions

This paper presented a novel decoupled DRL approach for autonomous UAV landings on maritime platforms. By separating horizontal and vertical control into specialised agents, the method achieved robust tracking across a range of wind and wave conditions, effectively mitigating nonlinear disturbances that challenge monolithic control strategies. Comparisons with a single-agent approach demonstrate that dividing the state and action spaces between two agents and training in disturbance-specific environments substantially improved landing accuracy and reliability.

Despite these successes, the vertical controller shows limited generalisation under extreme sea states, indicating the need for a more diverse and dynamic training environment. Additionally, the current framework does not address attitude alignment at touchdown, which becomes critical under severe wind and wave conditions. Future work will focus on extending the training environment to cover a wider operational envelope and integrating attitude control to enable fully autonomous, safe, and precise maritime landings under harsher environmental conditions, with efforts to test the method in real-world trials.

Acknowledgments

The authors gratefully acknowledge the support of the CSIRO Data61 Next Generation Graduate Program (NGGP) and Navantia Australia for sponsoring this research. Their financial support, resources, and ongoing engagement have been instrumental in enabling the successful completion of this work. The authors also appreciate the valuable insights and collaboration provided through this partnership, which have significantly contributed to the development and outcomes of this study. The authors further acknowledge additional support from the Czech Science Foundation (GAČR) under research project No. 26-22419S.

References

[1] Carlos A. Trasviña-Moreno, Rubén Blasco, Álvaro Marco, Roberto Casas, and Armando Trasviña-Castro. Unmanned Aerial Vehicle Based Wireless Sensor Network for Marine-Coastal Environment Monitoring. *Sensors*, 17(3):460, March 2017.

[2] Martin Messmer, Benjamin Kiefer, Leon Amadeus Varga, and Andreas Zell. UAV-Assisted Maritime Search and Rescue: A Holistic Approach, March 2024.

[3] Theodoros Karachalios, Panagiotis Moschos, and Theofanis Orphanoudakis. Maritime Emission Monitoring: Development and Testing of a UAV-Based Real-Time Wind Sensing Mission Planner Module. *Sensors*, 24(3):950, January 2024.

[4] Nikolaos Nomikos, Panagiotis K. Gkonis, Petros S. Bithas, and Panagiotis Trakadas. A survey on uav-aided maritime communications: Deployment considerations, applications, and future challenges. *arXiv preprint arXiv:2209.09605*, sep 2022.

[5] Jingbo Wang, Kaiwen Zhou, Wenbin Xing, Huanhuan Li, and Zaili Yang. Applications, Evolutions, and Challenges of Drones in Maritime Transport. *Journal of Marine Science and Engineering*, 11(11):2056, November 2023.

[6] José Amendola, Linga Reddy Cenkeramaddi, and Ajit Jha. Drone Landing and Reinforcement Learning: State-of-Art, Challenges and Opportunities. *IEEE Open Journal of Intelligent Transportation Systems*, 5:520–539, 2024.

[7] Gangik Cho, Joonwon Choi, Geunsik Bae, and Hyondong Oh. Autonomous ship deck landing of a quadrotor UAV using feed-forward image-based visual servoing. *Aerospace Science and Technology*, 130:107869, November 2022.

[8] Haohua Dong, David Cabecinhas, and Rita Cunha. Vision-based control for landing an aerial vehicle on a marine vessel, April 2024.

[9] Zhe-Cheng Xu, Bin-Bin Hu, Bin Liu, X. D. Wang, and Hai-Tao Zhang. Vision-based autonomous landing of unmanned aerial vehicle on a motional unmanned surface vessel. In *Proceedings of the 39th Chinese Control Conference (CCC)*, pages 6845–6850, Shenyang, China, July 2020. Chinese Association of Automation.

[10] J. M. Riola, J. J. Diaz, and J. M. Giron-Sierra. The prediction of calm opportunities for landing on a ship: Aspects of the problem. In *OCEANS 2011 IEEE - Spain*, pages 1–8, June 2011.

[11] Shadi Abujoub, Johanna McPhee, and Rishad A. Irani. Methodologies for landing autonomous aerial vehicles on maritime vessels. *Aerospace Science and Technology*, 106:106169, November 2020.

[12] Christopher M. Hendrick, Emma R. Jaques, Joseph F. Horn, Jack W. Langelaan, and Anish J. Sydney. Model-scale evaluation of autonomous ship landing guidance and control modes. *Journal of the American Helicopter Society*, 69(2):1–20, April 2024.

[13] Ondřej Procházka, Filip Novák, Tomáš Báča, Parakh M. Gupta, Robert Pěnička, and Martin Saska. Model predictive control-based trajectory generation for agile landing of unmanned aerial vehicle on a moving boat. *Ocean Engineering*, 313:119164, December 2024.

[14] Z. Jiang and G. Song. A Deep Reinforcement Learning Strategy for UAV Autonomous Landing on a Platform, September 2022.

[15] Riccardo Polvara, Massimiliano Patacchiola, Sanjay Sharma, Jian Wan, Andrew Manning, Robert Sutton, and Angelo Cangelosi. Toward End-to-End Control for UAV Autonomous Landing via Deep Reinforcement Learning. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 115–123, June 2018.

[16] Shaofan Wang, Ke Li, Jiaao Chen, and Tao Zhang. Unmanned Aerial Vehicle Autonomous Visual Landing through Visual Attention-Based Deep Reinforcement Learning. In *2023 42nd Chinese Control Conference (CCC)*, pages 4143–4148, July 2023.

[17] Lizhen Wu, Chang Wang, Pengpeng Zhang, and Changyun Wei. Deep Reinforcement Learning with Corrective Feedback for Autonomous UAV Landing on a Mobile Platform. *Drones*, 6(9):238, September 2022.

[18] Jingyi Xie, Xiaodong Peng, Haijiao Wang, Wenlong Niu, and Xiao Zheng. UAV Autonomous Tracking and Landing Based on Deep Reinforcement Learning Strategy. *Sensors*, 20(19):5630, January 2020.

[19] Alejandro Rodriguez-Ramos, Carlos Sampedro, Hriday Bavle, Paloma de la Puente, and Pascual Campoy. A Deep Reinforcement Learning Strategy for UAV Autonomous Landing on a Moving Platform. *Journal of Intelligent & Robotic Systems*, 93(1):351–366, February 2019.

[20] Ali M. Ali, Aryaman Gupta, and Hashim A. Hashim. Deep Reinforcement Learning for sim-to-real policy transfer of VTOL-UAVs offshore docking operations. *Applied Soft Computing*, 162:111843, September 2024.

[21] Qun Ma, Yibo Wu, Muhammad Usman Shoukat, Yukai Yan, Jun Wang, Long Yang, Fuwu Yan, and Lirong Yan. Deep Reinforcement Learning-Based Wind Disturbance Rejection Control Strategy for UAV. *Drones*, 8(11):632, November 2024.

[22] Bochan Lee, Vishnu Saj, Moble Benedict, and Dileep Kalathil. A Deep Reinforcement Learning Control Strategy for Vision-based Ship Landing of Vertical Flight Aircraft. In *AIAA AVIATION 2021 FORUM*. American Institute of Aeronautics and Astronautics, 2021.

[23] Brian Bingham, Carlos Agüero, Michael McCarrin, Joseph Klamo, Joshua Malia, Kevin Allen, Tyler Lum, Marshall Rawson, and Rumman Waqar. Toward Maritime Robotic Simulation in Gazebo. In *OCEANS 2019 MTS/IEEE SEATTLE*, pages 1–10, October 2019.

[24] Willard J. Pierson Jr. and Lionel Moskowitz. A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. *Journal of Geophysical Research (1896-1977)*, 69(24):5181–5190, 1964.

[25] C. H. Douglas. *A Classification of Sea States*. Hydrographic Department, Admiralty, London, UK, 1921. Original scale describes sea states 0–9 based on visual observations of wind-driven waves and swell.

[26] World Meteorological Organization. *Guide to Meteorological Instruments and Methods of Observation*. WMO-No. 8. World Meteorological Organization, Geneva, Switzerland, 2018 edition edition, 2018.

[27] Francis Beaufort. *Remarks on the Classification of Winds, and Their Effects on Sailing Ships*. British Admiralty, London, UK, 1805. Introduced the Beaufort Wind Force Scale, 0–12, based on observed effects on sails.

[28] R. E. D. Bishop and W. G. Price. On modal analysis of ship strength. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 341(1624):121–134, 1974.

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.