




## Article

# Deep Reinforcement Learning for Integrated Non-Linear Control of Autonomous UAVs

Adnan Fayyaz ud Din <sup>1</sup>, Imran Mir <sup>2,\*</sup>, Faiza Gul <sup>3</sup>, Suleman Mir <sup>4</sup>, Nasir Saeed <sup>5,\*</sup> , Turke Althobaiti <sup>6</sup> ,  
Syed Manzar Abbas <sup>2</sup> and Laith Abualigah <sup>7,8</sup> 

- <sup>1</sup> Department of Mechanical & Aerospace Engineering, Institute of Avionics & Aeronautics, Air University, Islamabad 44000, Pakistan; adfdin@gmail.com
- <sup>2</sup> Department of Avionics Engineering, Air University, Aerospace & Aviation Campus Kamra, Islamabad 43600, Pakistan; manzar.abbas@aack.au.edu.pk
- <sup>3</sup> Department of Electrical Engineering, Air University, Aerospace & Aviation Campus Kamra, Islamabad 43600, Pakistan; faiza.gul@aack.au.edu.pk
- <sup>4</sup> Electrical Department, Fast-National University of Computer & Emerging Sciences, Peshawar 21524, Pakistan; suleman.mir@nu.edu.pk
- <sup>5</sup> Department of Electrical Engineering, Northern Border University, Arar 73222, Saudi Arabia
- <sup>6</sup> Department of Computer Science, Faculty of Science Northern Border University, Remote Sensing Unit, Arar 73222, Saudi Arabia; turke.althobaiti@nbu.edu.sa
- <sup>7</sup> Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan; laythyabat@aau.edu.jo
- <sup>8</sup> Faculty of Information Technology, Middle East University, Amman 11831, Jordan
- \* Correspondence: imranmir56@yahoo.com (I.M.); mr.nasir.saeed@ieee.org (N.S.)

**Abstract:** In this research, an intelligent control architecture for an experimental Unmanned Aerial Vehicle (UAV) bearing unconventional inverted V-tail design, is presented. To handle UAV's inherent control complexities, while keeping them computationally acceptable, a variant of distinct Deep Reinforcement Learning (DRL) algorithm, namely Deep Deterministic Policy Gradient (DDPG) is proposed. Conventional DDPG algorithm after being modified in its learning architecture becomes capable of intelligently handling the continuous state and control space domains besides controlling the platform in its entire flight regime. Nonlinear simulations were then performed to analyze UAV performance under different environmental and launch conditions. The effectiveness of the proposed strategy is further demonstrated by comparing the results with the linear controller for the same UAV whose feedback loop gains are optimized by employing technique of optimal control theory. Results indicate the significance of the proposed control architecture and its inherent capability to adapt dynamically to the changing environment, thereby making it of significant utility to airborne UAV applications.



**Citation:** ud Din, A.F.; Mir, I.; Gul, F.; Mir, S.; Saeed, N.; Althobaiti, T.; Abbas, M.S.; Abualigah, L. Deep Reinforcement Learning for Integrated Non-Linear Control of Autonomous UAVs. *Processes* **2022**, *10*, 1307. <https://doi.org/10.3390/pr10071307>

Academic Editor: Blaž Likozar

Received: 17 June 2022

Accepted: 28 June 2022

Published: 1 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** flight dynamics; linear quadratic regulator; machine learning; Reinforcement Learning; Deep Deterministic Policy Gradient; optimal reward function; optimal control theory; linear quadratic regulator nonlinear simulations

## 1. Introduction

The requirement for flexible and dependable network supporting systems grew in response to the enormous demand for trustworthy surveillance services [1–3]. Unmanned aerial vehicles (UAVs) apart from their conventional roles [3–9], have recently gained popularity as core network devices for delivering flexible and dependable network services such as mobile surveillance. It has been demonstrated that UAVs may adapt and dynamically update the positions of surveillance UAVs using their mobility capability [10,11]. Many critical characteristics of UAVs have been classified, including top-level configuration, restricting altitude, mean take-off weight, autonomous level, and even ownership. Similar nature optimization can be traced for ground applications [12–19].

The categorization of UAVs into fixed-wing, rotary-wing, and hybrid-wing aircraft is the first topic to be covered. Similar to ordinary fixed-wing human planes, UAVs also have a stiff wing with an airfoil that functions by boosting forward airspeed to fly. In comparison to the second design with rotary wings, this arrangement offers longer endurance flights and loitering, provides high-speed motion, and retains high payloads. Some of the drawbacks with this arrangement include the necessity for a runway to take-off/land because it relies on forward airspeed, as previously indicated, and the inability to do hovering activities because it must maintain continuous flight till landing at the conclusion of each journey. The rotary-wing arrangement, on the other hand, provides manoeuvrability benefits through the use of rotating plates. Its rotating blades can provide enough aerodynamic push to fly without using airspeed relative velocities. These properties enable this type of UAV to execute vertical take-off/landing, fly at low altitudes, such as in congested metropolitan areas, and hover. It cannot, however, sustain the same payload as a fixed-wing arrangement.

The third is the hybrid configuration, which has been proposed as a new type of aerial platform that combines the benefits of both known fixed and rotary wings (Convertpianes and Tail-sitters). UAVs have seen a dramatic surge in civilian applications in a variety of industries in recent years due to their low cost, adaptability, automation capabilities, and reduced safety restrictions because of no human on board. Power line inspection [20], wildlife conservation [21], construction inspection [22], and precision agriculture [23] are some of the few examples. UAVs, on the other hand, have payload size, weight, and power consumption limits, as well as restricted range and endurance. UAVs also face unique and variable environments because of their wide spread usages and thus appears as one of the biggest challenges. These drawbacks however should not be neglected, and they are especially important when deep learning algorithms are required to operate on a UAV.

### 1.1. Relevant Studies

Reinforcement Learning (RL) is the process of learning and selection of actions that what to perform in a given situation in order to maximize a numerical reward signal. The interactive intelligent agent in RL has a specific aim to achieve. In order to determine the best strategy that maximizes the cumulative reward from interactions with the environment, the agent must balance exploitation and exploration of the state space. An agent alters its behaviour or policy in this setting by being aware of the states, actions space, and rewards for each time step. Reinforcement Learning uses an optimization technique to maximize the cumulative reward throughout the whole state space. The environment model reflects the behaviour of the surrounding environment and aids algorithm performance by comprehending it.

In reinforcement learning, an agent is the entity that is required to do any action in the environment given its present state and previous experiences. Any reinforcement learning algorithm's fundamental goal is to allow the agent to quickly learn the best policy, which is represented by the symbol, that accurately completes the specified task and so yields the largest reward value [24].

Deep reinforcement learning (DRL) is a machine learning topic that combines deep learning (DL) i.e., use of neural nets and reinforcement learning together to deliver an optimal solution based on prior experience. This experience is built on iterations and analyzing a reward function in order to determine an agent's best conduct. The value-based method, policy-based approach, and model-based approach are the three primary kinds of deep reinforcement learning approaches. The agent's goal in value-based reinforcement learning is to identify the policy that maximizes a value function across a sequence of actions in the long term. The agent must then identify the policy that leads to the optimum value for the objective function in policy-based reinforcement learning. There are two types of techniques in this category: deterministic and stochastic. The former strategy takes the same action in every state, but the latter allows for differences in action depending on probabilistic assessments. Finally, model-based reinforcement learning relies on the agent

being given a model of the environment or being asked to learn a model of the environment in order to accomplish tasks in that environment.

Dao et al. [25] employed Sliding Mode Control (SMC) with Adaptive Dynamic Programming. Researchers have intelligently fused non-linear technique with RL technique. In current research, RL based dynamic Programming has been used for evaluating the Optimal Reward Function to be incorporated with conventional DDPG Algorithm, which makes it novel and unique as compared to other approaches. In another paper, Dao et al. [26] employed Actor Critic Network through ARL. This work is in accordance with the current work in which both DDPG and PPO have been used which are also from a family of Actor-Critic networks, with prime difference of a Reward Function. Pham et al. [27] discussed a tracking problem which applies Adaptive Reinforcement Learning (ARL) along with Disturbance Observer (DO) to achieve the objective. However, in current research Optimal RL based Dynamic Programming has been used for the reward function followed by using two variants of DRLs for devising Optimal Control Strategy. Similarly, Ref. [28] addressed a tracking control problem for an uncertain SV using ARL based cascaded structure. Current research on the other hand also employs Actor-Critic network by employing DDPG and PPO. At the heart however, lies RL based DP reward function which helps achieve convergence besides making the learning quicker.

Various researchers have employed RL for solving various control problems. Similar nature work [29,30] is done in the field of ground robotics [31–33] and wheeled inverted pendulum [25] by employing non-linear Sliding Mode Control (SMC) technique with Adaptive Dynamic Programming. Based on our review of the related research and cited papers, it has been assessed that in spite of being an active area of research, application of RL for UAV application is still in its infancy. The applications are focused towards limited segments mainly handling segmented flight phases [34,35]. Keeping in view the immense potential of RL algorithms and its limited application in entirety for UAV Flight Control systems development, it is considered mandatory to explore this dimension.

### 1.2. Research Contributions

In this paper, authors have presented the efficacy of DRL algorithm for a Unmanned Aerial Vehicle. Continuous state and control space domains are used to design a DRL-based control method that covers the whole flying regime of the Unmanned Aerial Vehicle while including nonlinear dynamical path limitations. Current study work. To our knowledge, the proposed method varies from previous research in the following ways:

- This study represents one of the pioneering work that applies DRL on controlling a non conventional UAV over its complete trajectory and flight envelope.
- Although a conventional DDPG algorithm lies at the core of current problem solving but it is pertinent to highlight that applied DDPG was modified with regards to its learning architecture through data feeding sequence to the replay buffer. Generated data was fed to the agent in smaller chunks to ensure positive learning through actor policy network. This data feeding distribution also makes it easier for the critic network to follow the policy and to help in positive learning of the agent.
- An optimal reward function was incorporated which primarily focuses on controlling the roll and yaw rates of the platform because of strong coupling between them due to inherent inverted V- tail design of the UAV. Optimal reward function was formulated from initial data collected in Replay Buffer before the formal commencement of agent's learning.

## 2. Problem Setup

Current research analyzes a pure Flight Dynamics problem from a perspective of controlling an experimental UAV in its entire flight regime employing intelligent control techniques that can handle continuous domains.

### 2.1. Flight Dynamics Modeling

The geometric, flight and mass parameters of the experimental UAV utilized in this study are elaborated in Table 1. The UAV configuration consisting of 'V' tail with inverted fin have been adopted to meet the challenging performance requirements and aerodynamic characteristics. In this paper, the flight dynamics are modelled using the 6-DOF model, which is often used to represent vehicle motion in 3D space [36–42].

**Table 1.** UAV parameters.

No.	Nomenclature	Value	No.	Nomenclature	Value
1	Weight	600 Kg	6	Wing Area/Mean Aerodynamic chord/Wing Span	$9.312 \text{ ft}^2/0.8783 \text{ ft}/4.101 \text{ ft}$
2	Angle of incidence of wing	$6^\circ$	7	Horizontal tail incidence Angle	$0^\circ$
3	cg location	$3.78 \text{ ft}$	8	Vertical location of center of gravity from reference plane (Vcg)	$2.4 \text{ ft}$
4	Airfoil of Tail (Vertical)	NACA-6-65A007	9	Wing airfoil	NACA-6-65-210
5	Airfoil of Tail (Horizontal)	NACA-6-65A007	10	Moment of Inertia Matrix $\text{slug} - \text{ft}^2$	$\begin{bmatrix} 41 & 0.16 & -12.4 \\ 0.16 & 690 & 0.0037 \\ -12.4 & 0.0037 & 716 \end{bmatrix}$

### 2.2. State and Action Space Characterization

The problem is formulated as a nonlinear system and depicted in Equation (1):

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) \quad (1)$$

hence,  $\vec{x} \in \mathbb{R}^{12}$  is called as state vector,  $\vec{u} \in \mathbb{R}^2$  is called as control vector, and  $\hat{\vec{x}} \in \mathbb{R}^{12}$  are the updated state estimates. The state vector is depicted by Equation (2).

$$\vec{x} = [U, V, W, \phi, \theta, \psi, P, Q, R, h, P_N, P_E]^T, \quad \vec{x} \in \mathbb{R}^{12} \quad (2)$$

A control vector is depicted in Equation (3)

$$\vec{u} = [RCF, LCF]^T, \quad \vec{u} \in \mathbb{R}^2 \quad (3)$$

As the intended motion of the UAV spreads over a localized area of Earth, a flat non-rotating Earth is assumed for all mathematical analysis. The governing equations of motion represent (a) dynamics of translation, (b) dynamics of rotation, (c) kinematics and (d) navigation assuming non-rotating Earth.

### 2.3. Drl Algorithms and Appropriate Selection

Reinforcement learning algorithms for discrete domains are mainly used for finding a state-value function  $\mathcal{V}\pi^*$ , by following a policy  $\pi$ . The  $\pi$  is dependent on time which helps in guiding the choice of action to be taken.

$$\pi(z|y) = \mathbb{P}[A_t = z|S_t = y] \quad (4)$$

The output from state  $y$ , by following policy  $\pi$  and collecting scalar incentives, during transitioning between states is the state-value function (5). The agent's behaviour is closely

monitored to ensure that all states are visited at least once throughout the learning process. The return that is gathered by the agent existing in any specific state  $y$  and doing an action determines the action-value function (6).

$$v_{\pi}(y) = \mathbb{E}_{\pi}[G_t | S_t = y] \quad (5)$$

$$q_{\pi}(y, z) = \mathbb{E}_{\pi}[G_t | S_t = y, A_t = z] \quad (6)$$

It is appraised that Equations (4) and (5) presents novel mathematical architecture of the 'Optimal Reward Function'. It has been developed with an Iterative process while implementing the Reinforcement Learning based Dynamic Programming concept in an innovative manner. This Optimal Reward function has been embedded with the conventional DDPG Algorithm thus making it a novel approach for solving a complex control problem with continuous space and action domains. Incorporated Optimal reward function is one of the reasons that ensures positive convergence.

Selection of appropriate RL algorithm is difficult and it is not easy to implement due to complexity of states and actions [43,44]. Elements such as state ( $y$ ) and action space ( $z$ ), policy search ( $\pi$ ) or value function ( $v$ ), either model free/based, requires neural nets (deep RL) etc are deriving parameters in formulating RL algorithms. RL algorithms range from Policy Gradients to Q-learning besides Actor-Critic methods. All the methods have their own strengths and weaknesses, however few factors like hyper-parameters, random seeds or environment properties have profound effects [45] in DRL algorithms.

As our problem has a complex action domain with continuous state, so policy gradient methods incorporating neural nets were preferred as they directly optimize the parameterized policy by using an estimator of the gradient of the expected cost. These primarily include Trust Region Policy Optimization (TRPO) [46], Proximal Policy Optimization (PPO) [47], Deep Deterministic Policy Gradient (DDPG) [48], and its variants Twin Delay DDPG (TD3), Soft Actor-Critic (SAC), Advantage Actor-Critic (A2C), Asynchronous Advantage Actor-Critic (A3C) and ACKTR (Actor-Critic using Kronecker-Factored Trust Region) [49]. TRPO and PPO use constraints and advantage estimation to perform network update.

TRPO uses conjugate gradient descent as the optimization method with a KL constraint while PPO reformulates the constraint as a penalty (or clipping objective). DDPG and ACKTR use actor-critic methods which estimate  $Q(s, a)$  and optimize a policy that maximizes the Q-function. DDPG does this using deterministic policies, while ACKTR uses Kronecker-factored trust regions to ensure stability with stochastic policies. Owing to the nature of problem at hand our requirement was to handle multi-processed continuous actions which further narrowed down our search to TRPO, PPO, DDPG and A3C only.

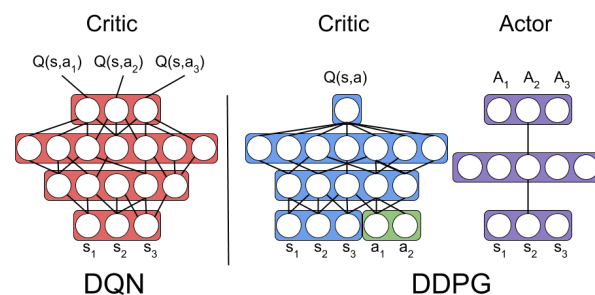
Critical challenge related to DDPG is sample inefficiency because actor is updated based on gradients evaluated when training of the critic neural network is taking place. Gradient is usually noisy because it relies on the outcome of the simulated episodes. Therefore, to avoid divergence off-policy DRL training algorithms maintain a copy of the actor and critic neural networks while undergoing training. DDPG usually faces convergence issues which are handled by employing various optimization algorithms among which Adam optimizer outperforms others because of its minimum training cost. Adam optimizer has also been employed in current research as well. But the best part about DDPG is that its Q value based and is more intuitive to implement.

DDPG is an effective policy gradient based RL algorithm [50], that can be configured for problems involving high dimensional continuous state space domain [51]. It is an off-policy algorithm, refer [52], whose behavioral policy is stochastic in nature while target policy is deterministic. Being model-free, it uses deep learning techniques that were introduced along with Deep Q Networks (DQNs) for efficient learning [53]. It utilizes the concept of replay buffer and then use experience replay to break up the temporal correlations [54].

Based on the basic architecture of the DDPG algorithm as articulated in [48,55,56], actor and critic neural nets along with their target networks were established in Python. TFlearn [57], a modular higher-level API to TensorFlow deep learning library [58] has been utilized during the research and exhibits remarkable performance. Designed Neural nets had three layers each for both actor and critic networks with first layer having 400 Neurons while second layer having 300 Neurons. It is pertinent to highlight that the selection of the number of neurons was finalized after repeated hit and trial by evaluating the learning performance every time. Two different activation functions have been used in the neural nets. tanh is used for the actor network function in order to include for both the positive and negative deflections of the controls while relu is used for the critic network function which gives a Q-value of present state based on the action as dictated by actor.

#### 2.4. Selection of Optimizer Algorithm

Adam optimizer which is an extension to stochastic gradient descent as explained by [59] was used for ensuring efficient learning of all the four actor critic and their target networks. Empirical results as evidenced in Figure 1 retrieved from the analysis of [60] demonstrates that Adam works well in practice and compares favorably to other stochastic optimization methods besides bearing minimum training cost, however some people have also used derivative of DDPG for positive optimization [55].



**Figure 1.** DDPG Actor-Critic Neural Networks.

Modern optimization algorithms such as Aquila Optimization Algorithm [61] and Hybrid Algorithm of Arithmetic Optimization Algorithm With Aquila Optimizer (AOAAO) [62], have special applications for machine learning based problem solving. However, Adam which possess inherent advantages over the two other extensions of stochastic gradient descent namely Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp) have been used in current research [63,64]. AdaGrad maintains a per-parameter learning rate which improves performance on problems with sparse gradients. RMSProp also maintains per-parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight. Adam instead of adapting the parameter learning rates based on the average first moment as in RMSProp, Adam also makes use of the average of the second moments of the gradients. Acquiring benefits of both, results show that it has minimum training cost among the various optimizers in use for DRL algorithms.

Adam updated the network weights iterative in training data during the learning phase. For the back-propagation optimisation the learning rate of both the actor and the critic was set to  $1 \times 10^{-3}$  with first and second moments set to 0.9, 0.999, respectively. Experience Replay Buffer size was set as 1 million i.e., after the complete replay buffer is filled the oldest data is popped out making place for the new incoming data. Batch size for calculating the gradient descent was maintained as 64 to improve the optimization. The reward discount was set as  $\gamma = 0.95$  and the soft update of the target neural networks was selected as  $\tau = 0.005$ . To allow exploration a simple Gaussian noise with  $\sigma = 0.25$  was also added and during the training the best model was saved. Keeping in view the wide ranging and varying numerical data of states and rewards owing to the peculiar nature of the problem, batch normalization was incorporated before feeding the data to neural nets for efficient training of Neural nets. Additionally, the data being generated

during simulated episodes was fed to the neural nets in chunks with an aim to speed up the learning curve.

In order to improve the efficacy of conventional DDPG algorithm, optimal penalty and reward function developed after an iterative process was utilized. Scalar reward function lies at the core of any RL problem as it guides the agent towards its goal. Significance of the reward function can be realized from the fact that it is the only measure for gauging success or failure of any particular action for the agent. If is not formulated correctly it may lead to the choice of actions which may take the agent away from priority goals. Reward function incorporated in this research aims at maximizing the glide range of the experimental glide UAV.

It is critical to understand that during the iteration process weights management was done carefully on logical grounds as otherwise random increase in the weights would increase pen, refer Equation (7). thus bringing the reward for each step sharply down, finally resulting in an unstable reward function. Due to the same, the difference of rates with the *desired absolute values for each different state* were also included in the cost function. The iterative process continued before reaching a final reward function to give a complete optimization process. In order to improve the control of states corresponding to experimental design vehicle, additional dynamic weights  $w7$ ,  $w8$ ,  $w9$  and  $w10$  were also added to the already finalized structure aiming to gain effective control of the changing rates with each step of the episode. Subsequently,  $ydis$  parameter was also added in the penalty to restrict platforms lateral movement in the Y-direction. Additionally, the  $r$  also included the altitude decrease i.e.,  $zcurr$  which is the decreasing altitude with every step. Additionally, rates were made part of the reward function in order to arrest any abnormal trends of roll and yaw coupling

$$pen = w1|P| + w2|Q| + w3|R| + w4\Delta P + w5\Delta Q + w6\Delta R + w7\delta P + w8\delta Q + w9\delta R + w10ydis \quad (7)$$

$$r = 10^{-3} \times xcurr^2 + (36000 - zcurr) \\ rew = r - pen \quad (8)$$

where  $pen$  represents the penalty defined at each step of the simulation.  $xcurr$  is the incremental current  $x$  value or covered gliding distance.  $r$  is the instantaneous scalar reward value based on increasing  $xcurr$ .  $w$  represent dynamic weights which are varied by the agent during the learning process, for limiting the rates in order to improve control of the gliding platform.  $zcurr$  is the decreasing altitude of the platform with every step of the episode.  $ydis$  parameter represents distance covered in East direction and is added in the penalty to restrict platform's lateral movement.  $P$ ,  $Q$  and  $R$  are the Roll, Pitch and Yaw rates respectively, whereas  $\Delta P$ ,  $\Delta Q$  and  $\Delta R$  represent the change in these instantaneous rates while  $\delta P$ ,  $\delta Q$  and  $\delta R$  are the difference between instantaneous and ideal targeted selected values for each of the three rates.

### 3. Results and Discussion

The results achieved through implementation of modified DDPG RL controller are discussed in this section. The launch conditions are taken as

1. Launch Condition No. 1 Altitude 35,500 ft, Mach 0.85; Angle of Attach  $0^\circ$
2. Launch Condition No. 2 Altitude 35,000 ft, Mach 0.7; Angle of Attach  $2^\circ$

Terminal State of the current MDP is recognized as the state when the "gliding UAV hits the ground with the employed condition of 'h' is less than or equal to zero that is when the altitude reduces to 'ZERO'".

### DDPG RL Controller Results

Body axis rates variation during the flight of UAV are depicted in Figure 2. With all 3 rates initialized at zero, agent selects random actions during the exploration phase while all the states resulting from the actions are stored in the buffer replay. Incorporated optimal reward function and the learning of agent based on the replay buffer gradually starts to make optimal trade-off among all three rates. Superior learning of DDPG agent based on neural nets can be appreciated from the smooth varying graphs instead of pointy ones. Though the rates are contained in the major part of the episode, however strong coupling between roll and yaw due to complex controls of the UAV, the roll and yaw rates show an increasing trend just before the culmination of one of the optimal episodes and thus validate the coupling behaviour. This behaviour of the agent gives us a peak into its exploration behaviour that is being managed through the added noise in the action policy after initially following learnt policy for a good reward.

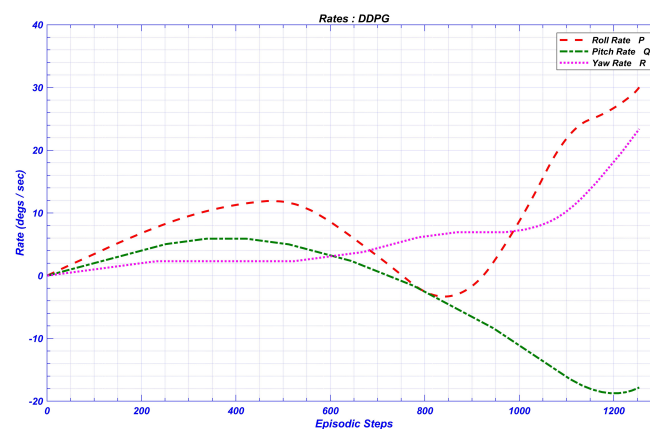


Figure 2. Angular rates variation.

Figure 3 explains the Euler angles variation during the flight. Both pitch and yaw angles are contained close to  $1^\circ$  during almost half of the initial flight phase. UAV shows a wing rocking behaviour for initial part of the episode because of the roll angle variation due to a sinusoidal behavior in the roll rate. Because of the inherent complex geometry of the vehicle the roll and yaw dynamics are complex. With agent subsequently learning to optimally trade-off rates, decreases the roll angle variation however, pitch and yaw angles show more variation. But overall the trade-off appears to be controllable and optimal path is maintained as variation in roll rate does not hamper the glide path range.

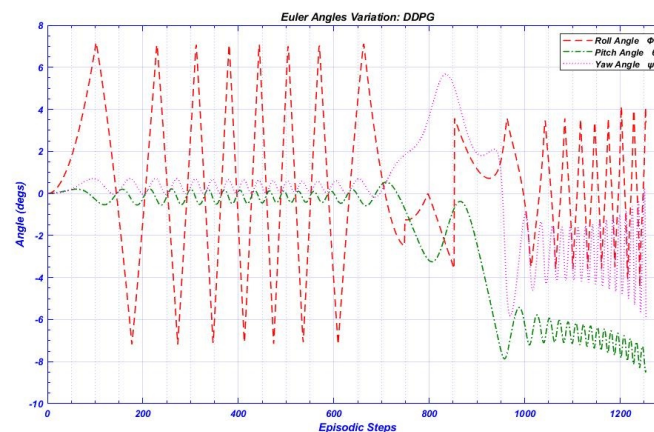
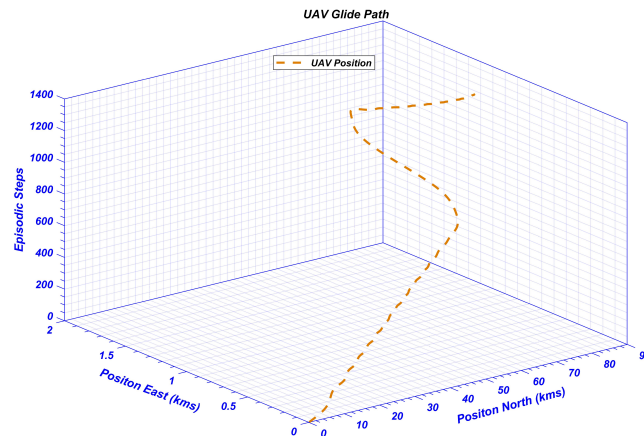


Figure 3. Euler Angles variation.

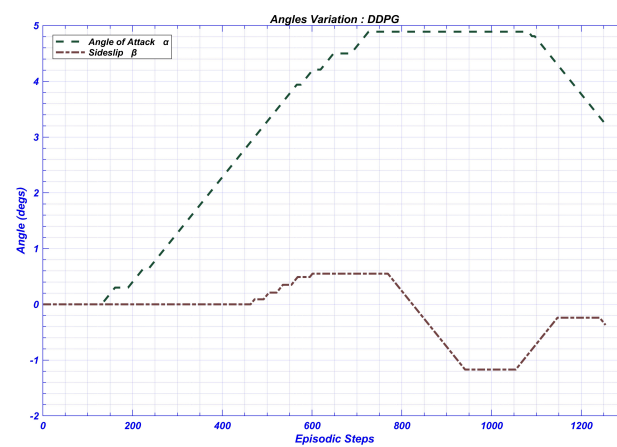


Similarly, Figure 4 depicts the glide path of the UAV which initially covers more distance north wards but with evolving scenario and increasing yaw angle variation the UAV follows the desired east ward direction.



**Figure 4.** Glide path of UAV.

Figure 5 depicts UAVs variation of angle of attack and sideslip angle during the flight. While the sideslip is contained between  $0.5$  to  $-1^\circ$ , the angle of attack initially increases to gain more lift, later maintains it close to  $6^\circ$ . The vehicle finally performs nose down just before culmination of the flight to hit the desired target.



**Figure 5.** Angles variation of UAV.

Velocity profile as shown in Figure 6 decreases smoothly as a result of drag and increase of alpha in the major part of the episode. However, at the later part of the flight it decreases significantly with the increase in yaw angle and thus sideslip, thereby increasing drag profoundly before touching the ground.

Altitude variation as depicted in Figure 7 is smooth and gradual in the initial part where the angle of attack is maintained close to  $5$  degs. However, the altitude shows a steep decline in the later part of the episode primarily for hitting the desired target location.

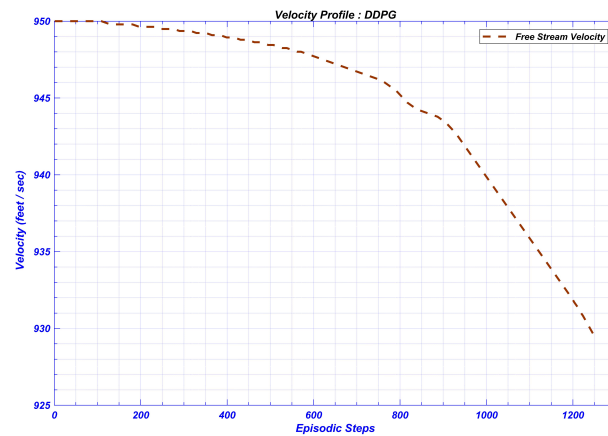


Figure 6. UAV Velocity profile.

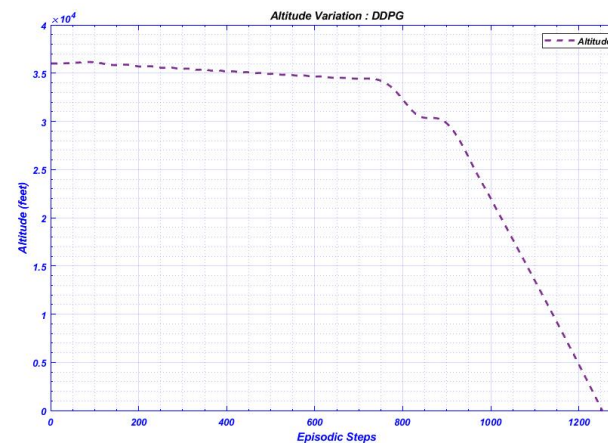


Figure 7. UAV Altitude variation.

Reward function variation is shown in Figure 8. Initially the agent is taking random actions thus exploring the action space. The moment replay buffer gets filled, the agent based on the learning from replay buffer starts to take desired actions which help achieve set objectives besides giving a rise in reward based on good prediction of actions. Convergence of reward function is also evident as the agent learns with increasing iterations and stabilizes itself after almost 8000 epochs.

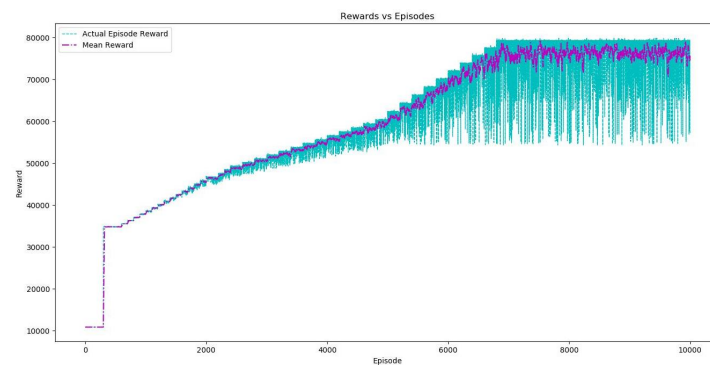


Figure 8. Reward Function.

#### 4. Comparison of Proposed Algorithm with Contemporary PID

After performing extended simulations under different environmental conditions to evaluate the the performance of our proposed DDPG algorithm, now we further extend our results. In this section, we will perform a detailed study and do a comprehensive analysis

by comparing the results achieved from proposed DDPG based control architecture with conventional LQR based control architecture. The analysis was carried out under setting with a higher level of intricacy and introducing complex situation loaded with varying environmental conditions. Results were then examined to draw inferences.

#### *DDPG vs. LQR Control Architecture*

The optimum trajectory for the flight conditions mentioned in Section 2.4 computed utilizing linear LQR based control architecture is depicted in Figure 9. It is evident that in spite of having range enhancement to about 85Kms, the problem of course deviation was encountered. This is primarily because the  $\psi$  dynamics were decoupled and were not included fully in the navigation loop. However, no such problem was encountered in the DDPG based control architecture elaborated in Figure 10. Thus the proposed DDPG algorithm not only significantly enhanced the vehicle range but also posed superior optimization results by reducing the Circular Error Probability (CEP).

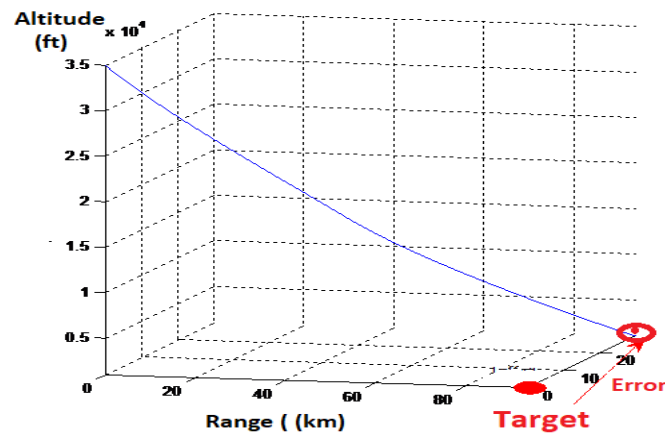


Figure 9. UAV trajectory: LQR based stabilization control architecture.

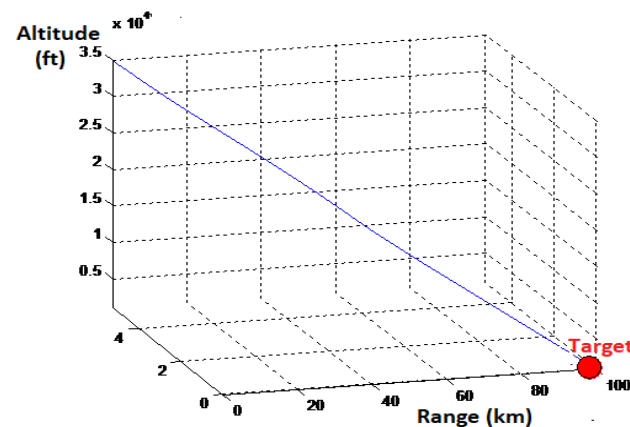


Figure 10. UAV trajectory: DDPG based control architecture.

## 5. Conclusions

In this research, a Reinforcement Learning based non-linear intelligent controller for an experimental UAV was developed. Results indicate efficacy of the control architecture as RL based intelligent controller keeps the platform dynamically stable throughout the flight envelope while satisfying vehicles design constraints. Performance of proposed RL controller extensively evaluated through non-linear 6-DOF simulations, exhibited stable flight. After ascertaining the performance characteristics, a detailed comprehensive analysis by comparing the results achieved from proposed DDPG based control architecture with conventional LQR based control architecture. The analysis was carried out under setting

with a higher level of intricacy and introducing complex situation loaded with varying environmental conditions. Authors believe that the investigations made in this research provides a mathematical-based analysis for designing a preliminary guidance and control system for the aerial vehicles utilizing intelligent controls. This research is expected to open avenues for researchers for designing intelligent control systems for aircraft, UAVs and the autonomous control of missile trajectories for both powered and unpowered configurations.

**Author Contributions:** Conceptualization, A.F.u.D. and F.G.; methodology, A.F.u.D. and I.M.; software, A.F.u.D.; validation, I.M. and N.S.; formal analysis, A.F.u.D., I.M. and F.G.; investigation, I.M., F.G. and S.M.A. resources, N.S. and T.A.; data curation, A.F.u.D., S.M.A. and I.M.; writing—original draft preparation, F.G. and S.M.; writing—review and editing, F.G., I.M., N.S. and L.A.; visualization, S.M.A., I.M., N.S. and L.A.; supervision, I.M.; project administration, I.M. and S.M.A.; funding acquisition, N.S. and T.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education on Saudia Arabia for funding this research work through the project number “IF\_2020\_NBU\_438”.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Din, A.F.U.; Akhtar, S.; Maqsood, A.; Habib, M.; Mir, I. Modified model free dynamic programming: An augmented approach for unmanned aerial vehicle. *Appl. Intell.* **2022**, 1–21. [\[CrossRef\]](#)
- Kim, D.; Park, S.; Kim, J.; Bang, J.Y.; Jung, S. Stabilized adaptive sampling control for reliable real-time learning-based surveillance systems. *J. Commun. Netw.* **2021**, *23*, 129–137. [\[CrossRef\]](#)
- Fatima, S.K.; Abbas, M.; Mir, I.; Gul, F.; Mir, S.; Saeed, N.; Alotaibi, A.A.; Althobaiti, T.; Abualigah, L. Data Driven Model Estimation for Aerial Vehicles: A Perspective Analysis. *Processes* **2022**, *10*, 1236. [\[CrossRef\]](#)
- Din, A.F.U.; Mir, I.; Gul, F.; Nasar, A.; Rustom, M.; Abualigah, L. Reinforced Learning-Based Robust Control Design for Unmanned Aerial Vehicle. *Arab. J. Sci. Eng.* **2022**, 1–16. [\[CrossRef\]](#)
- Mir, I.; Eisa, S.; Taha, H.E.; Gul, F. On the Stability of Dynamic Soaring: Floquet-based Investigation. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022; p. 0882.
- Mir, I.; Eisa, S.; Maqsood, A.; Gul, F. Contraction Analysis of Dynamic Soaring. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022; p. 0881.
- Mir, I.; Taha, H.; Eisa, S.A.; Maqsood, A. A controllability perspective of dynamic soaring. *Nonlinear Dyn.* **2018**, *94*, 2347–2362. [\[CrossRef\]](#)
- Mir, I.; Maqsood, A.; Akhtar, S. Dynamic modeling & stability analysis of a generic UAV in glide phase. *Proc. Matec Web Conf.* **2017**, *114*, 01007.
- Mir, I.; Eisa, S.A.; Taha, H.; Maqsood, A.; Akhtar, S.; Islam, T.U. A stability perspective of bioinspired unmanned aerial vehicles performing optimal dynamic soaring. *Bioinspiration Biomim.* **2021**, *16*, 066010. [\[CrossRef\]](#)
- Huang, H.; Savkin, A.V. An algorithm of reactive collision free 3-D deployment of networked unmanned aerial vehicles for surveillance and monitoring. *IEEE Trans. Ind. Inform.* **2019**, *16*, 132–140. [\[CrossRef\]](#)
- Nawaratne, R.; Alahakoon, D.; De Silva, D.; Yu, X. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Trans. Ind. Inform.* **2019**, *16*, 393–402. [\[CrossRef\]](#)
- Gul, F.; Mir, I.; Abualigah, L.; Mir, S.; Altalhi, M. Cooperative multi-function approach: A new strategy for autonomous ground robotics. *Future Gener. Comput. Syst.* **2022**, *134*, 361–373. [\[CrossRef\]](#)
- Gul, F.; Mir, S.; Mir, I. Coordinated Multi-Robot Exploration: Hybrid Stochastic Optimization Approach. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022; p. 1414.
- Gul, F.; Mir, S.; Mir, I. Multi Robot Space Exploration: A Modified Frequency Whale Optimization Approach. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022; p. 1416.
- Gul, F.; Mir, S.; Mir, I. Reinforced Whale Optimizer for Multi-Robot Application. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022; p. 1416.
- Gul, F.; Mir, I.; Abualigah, L.; Sumari, P. Multi-Robot Space Exploration: An Augmented Arithmetic Approach. *IEEE Access* **2021**, *9*, 107738–107750. [\[CrossRef\]](#)

17. Gul, F.; Rahiman, W.; Alhady, S.N.; Ali, A.; Mir, I.; Jalil, A. Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO–GWO optimization algorithm with evolutionary programming. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 7873–7890. [[CrossRef](#)]
18. Gul, F.; Mir, I.; Rahiman, W.; Islam, T.U. Novel Implementation of Multi-Robot Space Exploration Utilizing Coordinated Multi-Robot Exploration and Frequency Modified Whale Optimization Algorithm. *IEEE Access* **2021**, *9*, 22774–22787. [[CrossRef](#)]
19. Gul, F.; Mir, I.; Abualigah, L.; Sumari, P.; Forestiero, A. A Consolidated Review of Path Planning and Optimization Techniques: Technical Perspectives and Future Directions. *Electronics* **2021**, *10*, 2250. [[CrossRef](#)]
20. Martinez, C.; Sampedro, C.; Chauhan, A.; Campoy, P. Towards autonomous detection and tracking of electric towers for aerial power line inspection. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 284–295.
21. Olivares-Mendez, M.A.; Fu, C.; Ludvig, P.; Bissyandé, T.F.; Kannan, S.; Zurad, M.; Annaiyan, A.; Voos, H.; Campoy, P. Towards an autonomous vision-based unmanned aerial system against wildlife poachers. *Sensors* **2015**, *15*, 31362–31391. [[CrossRef](#)] [[PubMed](#)]
22. Carrio, A.; Pestana, J.; Sanchez-Lopez, J.L.; Suarez-Fernandez, R.; Campoy, P.; Tendero, R.; García-De-Viedma, M.; González-Rodrigo, B.; Bonatti, J.; Rejas-Ayuga, J.G.; et al. UBRISTES: UAV-based building rehabilitation with visible and thermal infrared remote sensing. In Proceedings of the Robot 2015: Second Iberian Robotics Conference, Lisbon, Portugal, 19–21 November 2015; Springer: Berlin/Heidelberg, Germany, 2016; pp. 245–256.
23. Li, L.; Fan, Y.; Huang, X.; Tian, L. Real-time UAV weed scout for selective weed control by adaptive robust control and machine learning algorithm. In Proceedings of the 2016 ASABE Annual International Meeting. American Society of Agricultural and Biological Engineers, Orlando, FL, USA, 17–20 July 2016; p. 1.
24. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [[CrossRef](#)]
25. Dao, P.N.; Liu, Y.C. Adaptive reinforcement learning strategy with sliding mode control for unknown and disturbed wheeled inverted pendulum. *Int. J. Control. Autom. Syst.* **2021**, *19*, 1139–1150. [[CrossRef](#)]
26. Dao, P.N.; Liu, Y.C. Adaptive reinforcement learning in control design for cooperating manipulator systems. *Asian J. Control* **2022**, *24*, 1088–1103. [[CrossRef](#)]
27. Vu, V.T.; Pham, T.L.; Dao, P.N. Disturbance observer-based adaptive reinforcement learning for perturbed uncertain surface vessels. *ISA Trans.* **2022**, *in press*.
28. Vu, V.T.; Tran, Q.H.; Pham, T.L.; Dao, P.N. Online Actor-critic Reinforcement Learning Control for Uncertain Surface Vessel Systems with External Disturbances. *Int. J. Control. Autom. Syst.* **2022**, *20*, 1029–1040. [[CrossRef](#)]
29. Hussain, A.; Hussain, I.; Mir, I.; Afzal, W.; Anjum, U.; Channa, B.A. Target Parameter Estimation in Reduced Dimension STAP for Airborne Phased Array Radar. In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6.
30. Hussain, A.; Anjum, U.; Channa, B.A.; Afzal, W.; Hussain, I.; Mir, I. Displaced Phase Center Antenna Processing For Airborne Phased Array Radar. In Proceedings of the 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 12–16 January 2021; pp. 988–992.
31. Szczepanski, R.; Tarczewski, T.; Grzesiak, L.M. Adaptive state feedback speed controller for PMSM based on Artificial Bee Colony algorithm. *Appl. Soft Comput.* **2019**, *83*, 105644. [[CrossRef](#)]
32. Szczepanski, R.; Bereit, A.; Tarczewski, T. Efficient Local Path Planning Algorithm Using Artificial Potential Field Supported by Augmented Reality. *Energies* **2021**, *14*, 6642. [[CrossRef](#)]
33. Szczepanski, R.; Tarczewski, T. Global path planning for mobile robot based on Artificial Bee Colony and Dijkstra’s algorithms. In Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 25–29 April 2021; pp. 724–730.
34. Kim, D.; Oh, G.; Seo, Y.; Kim, Y. Reinforcement learning-based optimal flat spin recovery for unmanned aerial vehicle. *J. Guid. Control. Dyn.* **2017**, *40*, 1076–1084. [[CrossRef](#)]
35. Pham, H.X.; La, H.M.; Feil-Seifer, D.; Nguyen, L.V. Autonomous uav navigation using reinforcement learning. *arXiv* **2018**, arXiv:1801.05086.
36. Mir, I.; Maqsood, A.; Eisa, S.A.; Taha, H.; Akhtar, S. Optimal morphing–augmented dynamic soaring maneuvers for unmanned air vehicle capable of span and sweep morphologies. *Aerosp. Sci. Technol.* **2018**, *79*, 17–36. [[CrossRef](#)]
37. Mir, I.; Maqsood, A.; Akhtar, S. Optimization of dynamic soaring maneuvers to enhance endurance of a versatile UAV. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Bangkok, Thailand, 21–23 April 2017; Volume 211, p. 012010.
38. Mir, I.; Eisa, S.A.; Taha, H.; Maqsood, A.; Akhtar, S.; Islam, T.U. A stability perspective of bio-inspired UAVs performing dynamic soaring optimally. *Bioinspiration Biomim.* **2021**, *16*, 066010. [[CrossRef](#)] [[PubMed](#)]
39. Mir, I.; Akhtar, S.; Eisa, S.; Maqsood, A. Guidance and control of standoff air-to-surface carrier vehicle. *Aeronaut. J.* **2019**, *123*, 283–309. [[CrossRef](#)]
40. Mir, I.; Maqsood, A.; Taha, H.E.; Eisa, S.A. Soaring Energetics for a Nature Inspired Unmanned Aerial Vehicle. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019; p. 1622.

41. Mir, I.; Eisa, S.A.; Maqsood, A. Review of dynamic soaring: Technical aspects, nonlinear modeling perspectives and future directions. *Nonlinear Dyn.* **2018**, *94*, 3117–3144. [CrossRef]
42. Mir, I.; Maqsood, A.; Akhtar, S. Biologically inspired dynamic soaring maneuvers for an unmanned air vehicle capable of sweep morphing. *Int. J. Aeronaut. Space Sci.* **2018**, *19*, 1006–1016. [CrossRef]
43. Hafner, R.; Riedmiller, M. Reinforcement learning in feedback control. *Mach. Learn.* **2011**, *84*, 137–169. [CrossRef]
44. Laroche, R.; Feraud, R. Reinforcement learning algorithm selection. *arXiv* **2017**, arXiv:1701.08810.
45. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep reinforcement learning that matters. *arXiv* **2018**, arXiv:1709.06560.
46. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning. PMLR, Lille, France, 7–9 July 2015; pp. 1889–1897.
47. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
48. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
49. Wu, Y.; Mansimov, E.; Grosse, R.B.; Liao, S.; Ba, J. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–14.
50. Heess, N.; Hunt, J.J.; Lillicrap, T.P.; Silver, D. Memory-based control with recurrent neural networks. *arXiv* **2015**, arXiv:1512.04455.
51. Luo, X.; Zhang, Y.; He, Z.; Yang, G.; Ji, Z. A two-step environment-learning-based method for optimal UAV deployment. *IEEE Access* **2019**, *7*, 149328–149340. [CrossRef]
52. Stooke, A.; Abbeel, P. rlypt: A research code base for deep reinforcement learning in pytorch. *arXiv* **2019**, arXiv:1909.01500.
53. Werbos, P.J.; Miller, W.; Sutton, R. A menu of designs for reinforcement learning over time. *Neural Netw. Control* **1990**, *3*, 67–95.
54. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic Policy Gradient Algorithms. 2014. Available online: <https://ieeexplore.ieee.org/document/6300641> (accessed on 17 June 2022).
55. Chen, J.; Xing, H.; Xiao, Z.; Xu, L.; Tao, T. A DRL agent for jointly optimizing computation offloading and resource allocation in MEC. *IEEE Internet Things J.* **2021**, *8*, 17508–17524. [CrossRef]
56. Pan, J.; Wang, X.; Cheng, Y.; Yu, Q. Multisource transfer double DQN based on actor learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2227–2238. [CrossRef] [PubMed]
57. Tflearn. 2016. Available online: <https://ieeexplore.ieee.org/document/8310951> (accessed on 17 June 2022).
58. Tang, Y. TF Learn: TensorFlow’s high-level module for distributed machine learning. *arXiv* **2016**, arXiv:1612.04251.
59. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
60. Kingma, D.P.; Ba, J. A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
61. Zhao, J.; Gao, Z.M.; Chen, H.F. The Simplified Aquila Optimization Algorithm. *IEEE Access* **2022**, *10*, 22487–22515. [CrossRef]
62. Zhang, Y.J.; Yan, Y.X.; Zhao, J.; Gao, Z.M. AOAAO: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer. *IEEE Access* **2022**, *10*, 10907–10933. [CrossRef]
63. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
64. CS231n. Convolutional Neural Networks for Visual Recognition. 2017. Available online: <https://cs231n.github.io/> (accessed on 17 June 2022).