Journal of Ambient Intelligence and Humanized Computing (2023) 14:4005–4022
https://doi.org/10.1007/s12652-022-04467-8

ORIGINAL RESEARCH

Development of reinforced learning based non-linear controller for unmanned aerial vehicle

Adnan Fayyaz Ud Din[1] · Imran Mir[2] · Faiza Gul[3] · Suhail Akhtar[4]

Received: 24 December 2021 / Accepted: 4 October 2022 / Published online: 5 December 2022

Abstract

Design complexities of trending UAVs and the operational harsh environments necessitates ***Control Law*** formulation utilizing intelligent techniques that are both robust, model-free and adaptable. In this research, an intelligent control architecture for an experimental Unmanned Aerial Vehicle (UAV) having an unconventional inverted V-tail design, is presented. Due to unique design of the vehicle strong roll and yaw coupling exists, making the control of vehicle challenging. To handle UAV's inherent control complexities, while keeping them computationally acceptable, a variant of distinct Deep Reinforcement learning (DRL) algorithm, namely Reformed Deep Deterministic Policy Gradient (R-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. The paper illustrates the application of modified DDPG algorithm (namely R-DDPG) towards the design, while the performance of the resulting controller is assessed in simulation using dynamic model of the vehicle. 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 achieved through application of Linear quadratic regulator (LQR) based control strategy. The efficacy of the results and performance characteristics, demonstrated the ability of the presented algorithm to dynamically adapt to the changing environment, thereby making it suitable for UAV applications.

Keywords Control law · Flight dynamics · Linear quadratic regulator · Machine learning · Reinforcement learning · Deep deterministic policy gradient · Optimal reward function · Optimal control theory · Linear quadratic regulator nonlinear simulations

List of symbols

Abbreviation

API	Application programming interface
b	Wing span (m)
$\tilde{c}$	Mean aerodynamic chord (m)
CAD	Computer aided design
CFD	Computational fluid dynamics
C_{M_x}	Rolling moment coefficient
C_{M_y}	Pitching moment coefficient
C_{M_z}	Yawing moment coefficient
C_{F_x}	X-direction force coefficient
C_{F_y}	Y-direction force coefficient
C_{F_z}	Z-direction force coefficient
DDPG	Deep deterministic policy gradient
DoF	Degree of freedom
g	Acceleration due Gravity (m/s^2)
h	Altitude (m)

✉ Imran Mir
imir@cae.nust.edu.pk

Faiza Gul
faiza.gul@aack.au.edu.pk

[1] Department of Mechanical & Aerospace Engineering, Institute of Avionics & Aeronautics, Air University, Islamabad, Pakistan

[2] School of Avionics & Electrical Engineering, College of Aeronautical Engineering, NUST, Risalpur, Pakistan

[3] Department of Electrical Engineering, Air University Aerospace & Aviation, Campus Kamra, Islamabad, Pakistan

[4] Department of Aeronautical Engineering, Institute of Space Technologies, Islamabad, Pakistan

LCF	Left control fin
ML	Machine learning
O-PPO	Optimal proximal policy optimization
POMDP	Partial observable Markov decision process
R-DDPG	Reformed deep deterministic policy gradient
m	Vehicle's mass (kg)
P	Roll rate (deg/s)
P_E	Position vector—east (km)
P_N	Position vector—north (km)
Q	Pitch rate (deg/s)
Parm	Parameter
R	Yaw rate (deg/s)
RL	Reinforcement Learning
RCF	Right control fin
S	Wing area (m^2)
UAV	Unmanned aerial vehicle
V_T	Free stream velocity (m/s)
NNs	Neural networks
wt_i	Numerical weight (ith number)
Xcut	Current X-position (m)
Ycut	Current Y-position (m)
Zcut	Current Z-position (m)
R	Instantaneous reward
TR	Total reward
Py	Penalty

Greek symbols

α	Angle of attack (deg)
β	Sideslip angle (deg)
γ	Flight path angle (deg)
ψ	Yaw angle (deg)
ϕ	Roll angle (deg)
θ	Theta angle (deg)
δ_L	LCF deflection (deg)
δ_R	RCF deflection (deg)
ρ	Density of air (kg/m^3)

1 Introduction

UAVs represent one of the fastest progressing and dynamic segment within the paradigm of aviation industry (Mir et al. 2017b, c, 2018c, d, 2022a, b; Yanushevsky 2011). Although UAVs are being widely used in military applications (Paucar et al. 2018), but their potential for non-military purposes (disaster management, search and rescue/health care, journalism, shipping etc) is enormous (Nikolakopoulos et al. 2017; Nurbani 2018; Winkler et al. 2018), and is continually increasing with the advent of new technologies (Cai et al. 2014; Mir et al. 2019a, b, 2021a). To date, there are over 1000 UAV models being developed in over 52 countries, serving as indispensable assistant for human operators in a broad range of military and civil applications (Elmeseiry et al. 2021) including engineering geology (Cai et al. 2014; Giordan et al. 2020; Mir et al. 2022c). The rapidly increasing fleet of UAVs, along with the widening sphere of their utility, therefore presents a serious challenge for the designers with regards to formulation of unique optimal control strategies. However, technological advancements including development of advanced design tools in the aviation sector (Mir et al. 2017a, 2018a, b, 2021b) and the manufacturing of contemporary ground control vehicles (Gul and Rahiman 2019; Gul et al. 2019, 2020a, b, 2021a, b, c, 2022b, c, d; Szczepanski and Tarczewski 2021; Szczepanski et al. 2019, 2021) marked the requirement for the development of hi-fidelity systems. The rapidly increasing fleet of UAVs, along with the widening sphere of their applications, therefore presents a serious challenge for the UAV designers.

Linear and Non-linear control strategies have been used in the past in diverse ways for solving varying control problems to achieve desired objectives (Din et al. 2022; Mir et al. 2018c, d, 2019a, b). However, in depth understanding of inherent limitations of linear (Araar and Aouf 2014; Poksawat et al. 2017; Rinaldi et al. 2013) and non-linear control techniques (Aboelezz et al. 2021; Adams and Banda 1993; Adams et al. 1994, 2012; Chowdhary et al. 2014; Enomoto et al. 2013; Gul et al. 2022a; Peng 2021) along with the aim of achieving autonomy in controls for complex aerospace systems, provoked researchers, to look for intelligent methods (Zhou 2018) which are self capable of optimal, sequential decision making for a complex control problem. Under the ambit of ML, RL based algorithms (Kaelbling et al. 1996) have emerged as an effective technique for the design of autonomous intelligent control (Bansal et al. 2017; Zhou et al. 2019). Coupled with NNs, RL based algorithms have emerged as a robust methodology in solving complex continuous domain problems (Azar et al. 2021; Lillicrap et al. 2015; Wang et al. 2018) in changing environments which significantly overshadow the contemporary linear and nonlinear control strategies. Further, with the computer's increasing computation power, state of the art RL algorithms have started to exhibit promising results. RL due to its highly adaptive characteristics has increasingly found use in aerospace control applications for platforms like aircraft, missile trajectory control, fixed wing UAVs and so on.

RL inspired by human and animal behavior, can be defined as sequential events and their effects being reinforced by virtue of the actions taken (Thorndike 1898) in an evolving environment. Fundamentally, RL (Sutton and Barto 1998) at its core has an agent, which by virtue of its interaction with a given environment, gains experience through trial and error, thus improving its learning curve. Agent has no explicit knowledge of the underlying system (Verma 2020) and it's controllability. But it understands a notion of reward signal, on the basis of which the next decision is taken. Agent during training phase, learns about optimal action choices on the basis of reward function. In order to achieve optimal task performance, the trained agent selects actions which result in highest rewards. With the evolving system dynamics, dynamic reward signal correspondingly changes, agent thus in pursuit of higher rewards accordingly adapts its action policy. Though above facts declare RL as a strong tool to be utilized in controls problems; however, RL carries a baggage related to safety of its actions during the exploration phase of its learning (Dalal et al. 2018). To cope with increasing complexity of system dynamics and management of complex controls for enhancing adaptability with the changing environment, control system design based upon intelligent techniques is considered most appropriate (Mnih et al. 2015).

Later part of the last decade witnessed an increase in use of RL for aerospace control applications primarily because of the success of Deep Deterministic Policy Gradient (DDPG)—a deep RL algorithm for continuous domain problems like balancing of inverted pendulum or cart-pole system. Some researchers have applied DDPG on Quadcopters (Lin et al. 2020) and for morphing optimization on few other platforms (Xu et al. 2019). Further, application of Deep RL policy gradient algorithms is limited for conventional Quadcopters only, primarily focusing on controlling some specific phases of flight like attitude control (Koch et al. 2019) or compensating disturbances (Pi et al. 2021); with O-PPO outperforming others (Hu and Ql 2020). Further similar studies from 2019 and 2018 also discuss application of Deep RL for aerospace applications yet again in specific phases of the flight regimes and for conventional Quadcopters or fixed wing UAVs but not on any novel platform having unique design characteristics that too on the entire flight regime as in the current research.So to the best of our knowledge current research work differs form other contemporary works by following:-

(a) This study represents one of the pioneering work that applies DRL on controlling a non conventional UAV over its **complete trajectory and flight envelope.**

(b) 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.

(c) 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 Related work

2.1 Implementation of linear controls

For conventional UAVs, onboard Flight Control System **(FCS)** utilizing linear control strategies with well designed closed loop feedback have yielded satisfactorily results (Araar and Aouf 2014; Brière and Traverse 1993; Mir et al. 2019a, 2022a, b; Rinaldi et al. 2013; Rosales et al. 2019). Poksawat et al. (2017) designed cascaded PID controllers with automatic gain scheduling and controller adaption for various operating conditions. However, the control architecture was incapable of adapting to environmental disturbances apart from being highly dependent on sensor accuracy. Araar and Aouf (2014) utilized two different linear control techniques for controlling UAV dynamics. Linear Quadratic Servo (LQ-Servo) controller based on $\boldsymbol{L_2}$ and $\boldsymbol{L_\infty}$ norms were developed. Results however, showed limited robustness to external disturbances particularly to wind gusts. Further, Doyle et al. (1987) utilized H-1 loop shaping in connection with μ-synthesis, while Kulcsar (2000) utilized LQR architecture for the control of UAV. Both schemes satisfactorily manage requisite balance between robustness and performance of the devised controller. But both these linear methods besides being mathematically intricate, lose their effectivity with increasing complexity and non-linearity of the system. Similar work Hussain et al. (2020, 2021) has been done in the field of ground robotics (Gul et al. 2020b, 2021a, b, c, 2022c, d) as well.

2.2 Incorporation of non-linear controls

Linear controls being less robust against disturbances along with limitation of operating around varying equilibria with employing gain scheduling provoked researchers gradually resorted to applying non-linear techniques to make the controllers more adaptive and responsive to changing scenarios. Methodologies such as Back-stepping Sliding Mode Control **(SMC)** by Hou et al. (2020), Labbadi and Cherkaoui (2019), Non-linear Dynamic Inversion **(NDI)** by Tal and Karaman (2020) and Incremental Non-linear Dynamic Inversion **(INDI)** (Yang et al. 2020) have emerged to be strong tools in handling uncertainties and nonlinearities satisfactorily; besides having the potential to adapt to changing aircraft dynamics in connection with evolving environment. Escareno et al. (2006) designed non-linear control for attitude control of a quadcopter UAV using nested saturation technique. Results were experimentally verified, however, the control lacked measures for performance control in a harsh environment. In another work, Derafa et al. (2011) implemented a non-linear control algorithm for a UAV incorporating back-stepping sliding mode technique with adaptive gain. He was successful in keeping the chattering noise low because of the sign function which are pronounced in fixed gain controllers. Experimental results of UAV showed acceptable performance with regards to stabilization and tracking, however the algorithm was computationally expensive.

2.3 Research and use of intelligent controls

Realizing the shortcomings of linear and non-linear control strategies besides evolving enhanced performance requirements of UAVs, researchers started to resort to intelligent techniques coupling them with neural networks. Neural networks with varying learning schemes (Pan et al. 2018) are now in use in varying fields like smart networking (Chen et al. 2020; Xiao et al. 2021), satellite image recognition (Pirnazar et al. 2018) using fuzzy algorithms and prediction of certain water based estimates (Golian et al. 2020; Ostad-Ali-Askari et al. 2017) along side flight dynamics and controls domain. In one of the flight control studies, (Novati et al. 2018) employed deep RL for gliding and perching control of a two-dimensional elliptical body and concluded that model-free character and robustness of deep RL suggest a promising framework for developing mechanical devices capable of exploiting complex flow environments. Kroezen (2019) in his research has implemented reinforcement learning as an adaptive nonlinear control. Lei (2021) in their work briefly explain the constituents of DDPG algorithm and further elaborate its usage. Rodriguez-Ramos et al. (2019) successfully employed deep RL for autonomous landing on a moving platform again just focusing on the landing phase.

Bouhamed et al. (2020) employed UAV path planning framework using deep reinforcement learning approach. Dong and Zou (2020) optimized robot path utilizing deep RL techniques. Kim et al. (2017) in their work for Flat Spin Recovery for UAV, utilized RL based intelligent controllers. Aircraft non-linearities were handled near the upset region in two phases as ARA (Angular rate Arrest) and UAR (Unusual Attitude Recovery) using DQN **(Q-learning with ANN (Artificial Neural Network))**. Dutoi et al. (2008) in a similar work has highlighted the capability of RL framework in picking best solution strategy based on its off-line learning which is specially useful in controlling UAV in harsh environments and during flight critical phases. Wickenheiser and Garcia (2008) exploited vehicle morphing for optimizing the perching maneuvers to achieve desired objectives.

2.4 Crux of intelligent controls implementation

Based on our review of the related research and cited papers, it has been assessed that application of RL, especially deep RL for **continuous action and state domains** is mostly limited to simple yet complex tasks of balancing inverted pendulums, legged and bipedal robots (Rastogi 2017) and miscellaneous board and computer games through effective implementation of a novel mix approach of both supervised and deep RL (Silver edt al. 2016; Xenou et al. 2018). Implementation of RL based control strategy with continuous state & action spaces for developing Flight Controls of UAVs have not been applied on the entire flight regime as has been accomplished in the current research. It has been used only for handling critical flight phases (Kim et al. 2017) where linear control theory is difficult to implement and for navigation of UAVs (Kimathi 2017). Moreover, the in-depth analysis of the results show slightly better performance by eliminating overshoots besides tracking a reference heading as compared to well tuned PID controller, however it still lacked the required accuracy as was anticipated. 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.

2.5 Research contributions

In this research, we explore the efficacy of DRL algorithm for an unconventional UAV. DRL based control strategy is formulated for continuous state and control space domains, that encompasses the entire flight regime of the UAV duly incorporating nonlinear dynamical path constraints. To reduce the overall cost, an unconventional UAV with an inverted V-tail is designed with the least number of control surfaces. This distinctive design of UAV resulted in an under actuated system, thus making the stability and control of the UAV prominently challenging.

Effective RL algorithm known as *DDPG* has been carefully employed for the current problem after being **specifically modified in its learning architecture** to achieve the desired objective of UAV range enhancement while keeping the computational time required for learning of the agent, minimal. The designed control framework optimized the range of the UAV without explicit knowledge of the underlying dynamics of the physical system.

Developed RL control algorithm learns off-line, on the basis of a reward function which is formulated and finalized after an iterative process. Control algorithm in line with the reward function autonomously ascertains the optimum sequence of the available deflections of control surfaces at each time step (0.2 s) to maximize UAV range.

An iteratively developed 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.

Vehicle's 6-degree of freedom (DoF) model is developed, registering its transnational and rotational dynamics. The effectiveness of the proposed strategy is further demonstrated by comparing the results with conventional LQR based control strategy. Simulation results show that apart from improved Circular error probable (CEP) of reaching the designated location, range of UAV has also significantly increased with the proposed RL controller. Based on promising results, it is evidently deduced that RL has immense potential in the domain of intelligent controls for future progress because of its capability of adaptive, real time decision making in uncertain environments.

2.6 Proposed R-DDPG algorithm features

The proposed R-DDPG architecture embeds three main techniques in baseline DDPG algorithm which differentiates it from its contemporary DRL algorithms to perform better. These are elaborated below:-

(a) Incorporation of agent's unique learning architecture by tailoring the standard data feeding sequence to replay buffer which results in enhanced episodic learning.
(b) Use of Adam Optimizer for improving the DDPG convergence.
(c) Employment of an optimal reward function which intelligently controls the roll-yaw coupling and also aids in quicker learning of the agent.

All these factors ensure better learning and positive convergence of the proposed algorithm to obtain desired objectives of range enhancement and overall flight stability over the entire flight envelope.

3 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.

3.1 Problem modelling as a partial observable Markov decision process

Formally, MDP is understood as a mathematical based architecture in which sequential actions being taken over time, affect both the immediate rewards and the future states. A Markov process is a tuple of $< S, P, R, \gamma >$ where S is a finite set of states, P is a state transition probability matrix, R is a reward function and γ is a discount factor (usually ranging from 0 to 1) over cumulative rewards of an episode, Figure 1 represents student states and immediate rewards in red for exiting the states (Silver 2015). It is an ideal framework to handle problems that focuses on maximizing longer term return by carrying specific sequence of actions depending on the current state.

Because of adaptive sequential decision making nature of the current problem, it is modelled as a Model-Free Partial Observable Markov Decision Process (POMDP) that formally describes an environment for reinforcement learning like MDP, however the environment is partially observable and is based on the observation function O which becomes a part of the tuple of a standard MDP. It is noteworthy that almost all RL problems can be formalised as MDPs if they exhibit Markov's property where **the future is independent of the past given the present**.

3.2 UAV geometric and mass parameters

The UAV platform utilized in this research is an experimental vehicle whose geometrical parameters are designed to fulfill the desired stability and performance requirements . The UAV has a wing-tail configuration with unconventional

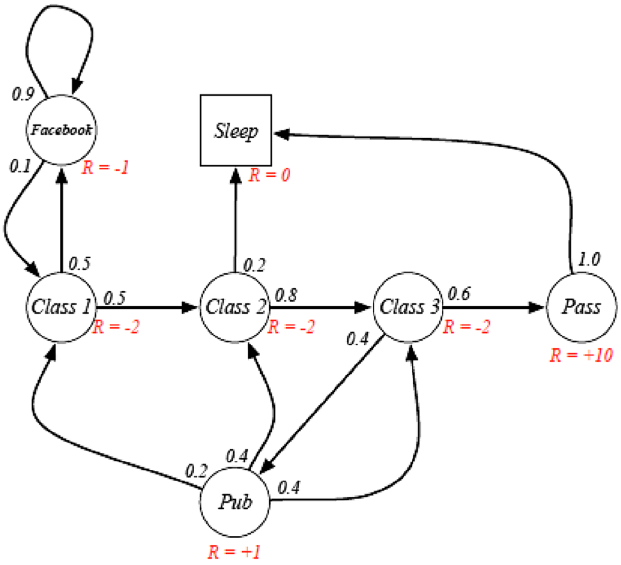

Fig. 1 Markov decision process

controls. The control surfaces are kept to a bear minimum to reduce the structural complexity and cost.

It has two all moving inverted V tails to function as ruddervators. These control surfaces can move both symmetrically for controlling pitch motion and deferentially for coupled roll and yaw movements. An additional ventral fin is also placed at the bottom side for enhancing lateral stability.

3.3 Flight dynamics modeling

Flight dynamics modeling is carried out utilizing 6-DOF (Mir et al. 2019a) model, which is typically utilized to model the vehicle motion in 3D space (Mir et al. 2019a). 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 problem is formulated as a nonlinear system of the form depicted in Eq. (1):

$$\dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u}) \tag{1}$$

where, $\mathbf{x} \in \mathbb{R}^{12}$ is the state vector, $\mathbf{u} \in \mathbb{R}^{2}$ is the control vector, and $\dot{\mathbf{x}} \in \mathbb{R}^{12}$ are the updated state estimates. The state vector in body axis is defined by Eq. (2.

$$\boldsymbol{x} = \left[U,\ V,\ W,\ \phi,\ \theta,\ \psi,\ P,\ Q,\ R,\ h,\ P_N,\ P_E\right]^T,\ \ \boldsymbol{x} \in \mathbb{R}^{12} \tag{2}$$

Control vector with continuous action space is defined in Eq. (3)

$$\boldsymbol{u} = [LCF,\ RCF]^T,\ \ \boldsymbol{u} \in \mathbb{R}^2 \tag{3}$$

3.4 Aerodynamic parameter estimation

The body aerodynamic force and moment coefficients vary with the flight conditions and control settings. A high fidelity aerodynamic model is necessary to accurately determine these aerodynamic coefficients. In this research, both empirical (Napolitano et al. 2000) and non-empirical techniques (such as CFD (Petterson 2006) and USAF DATCOM (Finck et al. 1978)) are utilized to determine these coefficients. The high fidelity model employed for aerodynamic parameter estimation is elaborated in Eq. (4):

$$C_i = C_{i,static}\left(\alpha, \beta, \delta_{control}, M\right) + C_{i,dynamic}\left(\dot{\alpha}, \dot{\beta}, p, q, r\right) \tag{4}$$

where $C_i = C_L, C_D, C_Y, C_l, C_m, C_n$ represents the coefficient of lift, drag, side force, rolling moment, pitching moment, and yawing moment respectively.

Evaluation of static (basic) coefficient data (see Eq. (5)) is achieved utilizing Computational Fluid Dynamics (CFD) (Buning et al. 2004; Petterson 2006) technique and are conventionally a function of control ($\delta_{control}$), angle of attack (α), side slip (β) and mach number *(M)*.

$$\begin{aligned} &C_{i,static}\left(\alpha, \beta, \delta_{control}, M\right) \Rightarrow C_{D_b}(\alpha, \beta, \delta_{control}, M), \\ &C_{L_b}(\alpha, \beta, \delta_{control}, M), C_{Y_b}(\alpha, \beta, \delta_{control}, M), \\ &C_{l_b}(\alpha, \beta, \delta_{control}, M), C_{m_b}(\alpha, \beta, \delta_{control}, M), C_{n_b}(\alpha, \beta, \delta_{control}, M), \end{aligned} \tag{5}$$

where C_{D_b}, C_{L_b}, C_{Y_b}, C_{l_b} C_{m_b}, C_{n_b} represents the basic components of the aerodynamic forces and moments as a function of ($\delta_{control}$), angle of attack (α), side slip (β) and mach number *(M)*.

Similarly dynamic component (Eq. (6)) consists of rate and acceleration derivatives and are evaluated utilizing empirical (Napolitano et al. 2000) and non-empirical [‘USAF Stability and Control DATCOM’ (Finck et al. 1978)] techniques.

$$\begin{aligned} &C_{i,dynamic}\left(\dot{\alpha}, \dot{\beta}, p, q, r\right) = Rate\ derivatives + \\ &Acceleration\ derivatives \end{aligned} \tag{6}$$

Rate derivatives are the derivatives due to roll rate (p), pitch rate (q) and yaw rate (r) while acceleration derivatives are the derivatives due to change in the aerodynamic angles ($\dot{\alpha}, \dot{\beta}$). They are shown in Eqs. (7) and (8) respectively.

$$\begin{aligned} &Rate\ derivatives = \left(C_{L_q}, C_{D_q}, C_{m_q}\right) \\ &\quad + \left(C_{Y_p}, C_{l_p}, C_{n_p}\right) + \left(C_{Y_r}, C_{l_r}, C_{n_r}\right) \end{aligned} \tag{7}$$

$$\begin{aligned} &Acceleration\ derivatives \\ &= \left(C_{L_{\dot{\alpha}}} + C_{D_{\dot{\alpha}}} + C_{m_{\dot{\alpha}}}\right) + \left(C_{Y_{\dot{\beta}}} + C_{l_{\dot{\beta}}} + C_{n_{\dot{\beta}}}\right) \end{aligned} \tag{8}$$

The aerodynamic coefficients modeling method employed in this research utilize curve-fitting to generate polynomial forms of these coefficients. For the platform under study, a detailed CFD analysis provided the numerical data for the coefficients in different flight regimes (Buning et al. 2004; Petterson 2006; Roaskam 2001).

The flight conditions were based on aerodynamic angles (α, β) variation along with the control fins varying deflections. A linear model consisting of the four basic parameters (α, β, LCF and RCF) are used to model the coefficients. The results were then refined using regression techniques and the computed coefficients provided a 95% confidence.

4 Formulation of RL algorithm

4.1 DRL algorithms and appropriate Selection

Reinforcement learning algorithms for discrete domains are aimed at finding an optimal state-value function $\mathcal{V}\pi^*$ or an action-value function $Q\pi^*$, while following a policy π which is a time dependent distribution over actions given states and guides the choice of action at any given state.

$$\boldsymbol{\pi(a|s)} = \mathbb{P}\left[\boldsymbol{A_t = a|S_t = s}\right] \tag{9}$$

State-value function (10) is the expected return starting from state s, while following policy π and gathers scalar rewards once transitioning between the states. The behavior of the agent is carefully controlled so that all states have been visited at least once during the course of learning. However, the action-value function (11) is determined by the return that is accumulated by the agent being in any particular state s and taking an action a.

$$\boldsymbol{v_\pi(s)} = \mathbb{E}_\pi\left[\boldsymbol{G_t|S_t = s}\right] \tag{10}$$

$$\boldsymbol{q_\pi(s,a)} = \mathbb{E}_\pi\left[\boldsymbol{G_t|S_t = s, A_t = a}\right] \tag{11}$$

Selection of appropriate RL algorithm is challenging as its implementation varies from problem to problem in terms of complexity of states and actions (Hafner and Riedmiller 2011; Laroche and Feraud 2017). Factors such as state (s) and action space (a) domain type (discrete or continuous), policy search (π) or value function (v), model free or model based, requirement for incorporation of NNs (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 (Henderson et al. 2018) in DRL algorithms

As our problem has a complex continuous state and action space so policy gradient methods incorporating NNs 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) (Schulman et al. 2015), Optimized Proximal Policy Optimization (O-PPO) (Schulman et al. 2017), Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al. 2015), 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) (Wu et al. 2017). TRPO and O-PPO use constraints and advantage estimation to perform network update.

TRPO uses conjugate gradient descent as the optimization method with a KL constraint while O-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 Kronecketer-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, O-PPO, DDPG and A3C only. Few of issues with said algorithms are as follows that lead to selection of a simpler and easily implementable DDPG:-

(a) Major disadvantage of TRPO is that it's computationally expensive and is difficult to model.
(b) O-PPO though gives a better convergence and performance rate than other techniques but is sensitive to changes.
(c) A3C though lacks the convergence rate but is very useful when large computation power is available and concept of transfer learning on similar environments is required.

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.

The current research work problem is a complex non-linear problem with mixed coupled controls. The problem has a **12** dimensional state space and a **2** dimensional action space, both of which are continuous. Realizing the complexity of the problem at hand due to continuous state and action space (Kingma and Ba 2014), DRL algorithms were studied with regards to practical implementation. Deep Deterministic Policy Gradient algorithm is employed which has been specifically modified and is named as Reformed Deep Deterministic Policy Gradient (R-DDPG) to adapt to the desired problem requirements optimally. DDPG is an effective policy gradient based RL algorithm (Heess et al. 2015), that can be configured for problems involving high dimensional continuous state space domain (Luo et al. 2019). It is an off-policy algorithm (refer Algorithm 1), 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 (Werbos et al. 1990). It utilizes the concept of replay buffer and then use experience replay to break up the temporal correlations (Silver et al. 2014).

4.2 Structure, hyper-paramaterization and optimization strategy of implemented R-DDPG algorithm

Based on the basic architecture of the DDPG algorithm as articulated in Lillicrap et al. (2015) and Chen et al. (2021), Pan et al. (2018), actor and critic NNs along with their target networks were established in Python. TFlearn (Tang 2016), a modular higher-level API to TensorFlow deep learning library (Tang 2016) has been utilized during the research and exhibits remarkable performance. Designed NNs 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 NNs. **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.

Adam optimizer which is an extension to stochastic gradient descent as explained by Kingma and Ba (2014) was used for ensuring efficient learning of all the four actor critic and their target networks. Empirical results retrieved from the analysis of Kinga (2015) 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 (Chen et al. 2021). As per CS231n (2017); Ruder (2016) among the modern optimization methods, Adam possess inherent advantages over the two other extensions of stochastic gradient descent namely Adaptive gradient algorithm (AdaGrad) and Root mean square propagation (RMSProp). 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^{-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** γ =0.95 and the **soft update of the target NNs** was selected as τ = 0.005. To allow exploration a simple **Gaussian noise with** σ = 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 NNs for efficient training of NNs. Additionally, the data being generated during simulated episodes was fed to the NNs in chunks with an aim to speed up the learning curve.

Algorithm 1 DDPG Algorithm

```
Input: initial policy parameters $\theta$, Q-function parameters $\phi$, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\theta_{targ} \leftarrow \theta$, $\phi_{targ} \leftarrow \phi$
repeat
    Observe state $s$ and select action $a = clip(\mu_\theta(s)) + \epsilon, a_{Low}, a_{High}), where \epsilon \sim \mathcal{N}$
    Execute $a$ in the environment and observe next state $s'$, reward $Ir$, and done signal $d$ to indicate whether $s'$ is terminal. Store $(s, a, r, s', d)$ in replay buffer $\mathcal{N}$
    if $s'$ is terminal then
        rest environment state
        if it's time to update then
            for however many updates do
                Randomly sample a batch of transitions, $B = (s, a, r, s', d)$ from $\mathcal{D}$
                Compute targets
                $y(r, s', d) = r + \gamma(1 - d)Q_{\phi_{targ}}(s', \mu_{\theta_{targ}}(s'))$
                Update Q-function by one step of gradient descent using
                $\nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s',d) \in \mathcal{B}} (Q_\phi(s, a) - y(r, s', d))^2$
                Update policy by one step of gradient ascent using
                $\nabla_\phi \frac{1}{|B|} \sum_{(s) \in \mathcal{B}} (Q_\phi(s, \mu_\theta(s))$
                Update target networks with
                $\phi_{targ} \leftarrow \rho\phi_{targ} + (1 - \rho)\phi$
$\theta_{targ} \leftarrow \rho\theta_{targ} + (1 - \rho)\theta$
until convergence
```

In order to improve the efficacy of conventional DDPG algorithm, optimal penalty and reward function developed after an iterative process was utilized, refer equation 12 & 13. After incorporation of the final reward function in the control algorithm, final results corresponding to all states of *R-DDPG controller*, plotted against sequential episodic time steps for the glide vehicle are presented in the results section. Selection of optimal control deflections by the controller during the flight regime amidst changing scenario can be appreciated from the results of states and the gliding range achieved.

Table 1 Parametric conditions

No	Parameters	Value
1	Altitude	39,000 ft
2	Mach No	0.9
3	Angle of Attack (α)	0° & 4 °

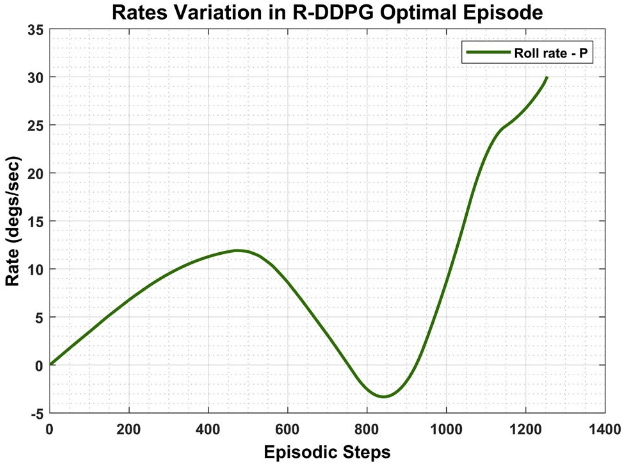

Fig. 2 Episodic variation of UAV roll rate

$$\begin{aligned} Py = {} & wt_1|P| + wt_2|Q| + wt_3|R| \\ & + wt_4\Delta P + wt_5\Delta Q + wt_6\Delta R \\ & + wt_7\delta P + wt_8\delta Q + wt_9\delta R + \\ & wt_{10}Ycut \end{aligned} \tag{12}$$

$$\begin{aligned} IR &= 1e^{-3} \times Xcut^2 + \left(36e^3 - Zcut\right) \\ TR &= IR - Py \end{aligned} \tag{13}$$

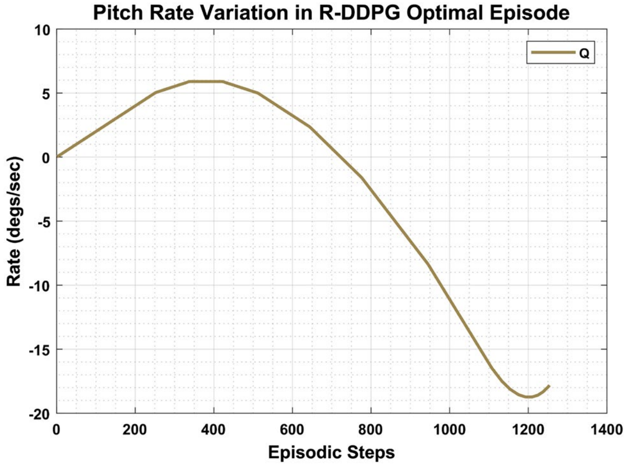

Fig. 3 Episodic variation of UAV pitch rate

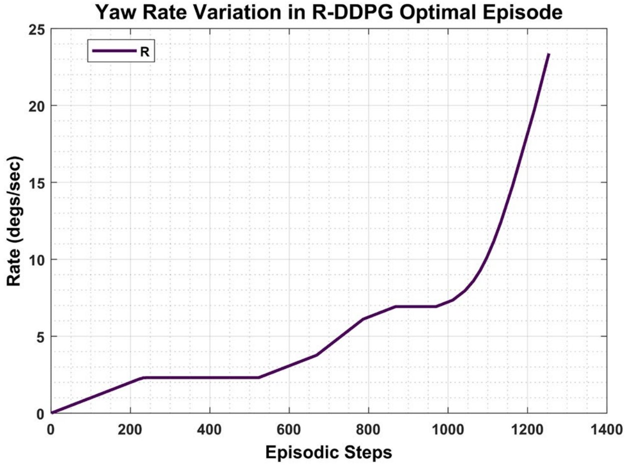

Fig. 4 Episodic variation of UAV yaw rate

5 Results insight and analysis

This section deliberates on the results achieved by implementing R-DDPG controller. Two different initial launch conditions for the Gliding Vehicle are specified in Table 1. It is pertinent to highlight that these results pertain to one of the best episodes saved over numerous episodes run after the agent completes its learning.

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".**

5.1 Results of R-DDPG controller

Figures 2, 3 and 4 depict the angular rate dynamics of the UAV exhibiting variation of body rates **P**, **Q** and **R**, along the episode. During the UAV launch, initially all the body rates are zero, subsequently agent selects random actions during the exploration phase while all the states resulting from the actions are continuously stored in the replay buffer. Based on the embedded optimal reward function and the learning from the replay buffer, agent gradually starts to make optimal trade-off among all three rates. Enhanced learning of R-DDPG agent based on NNs can be appreciated from the smoothness of graphs with . Though the rates are contained in the major part of the episode, however, strong coupling between roll and yaw dynamics due to UAVs complex controls, the roll and yaw rates show an increasing trend just before the culmination of one of the optimal episodes and thus validate the strong coupling behaviour. This behaviour of the agent gives us a peak into its exploration behaviour that is being managed through the added noise in

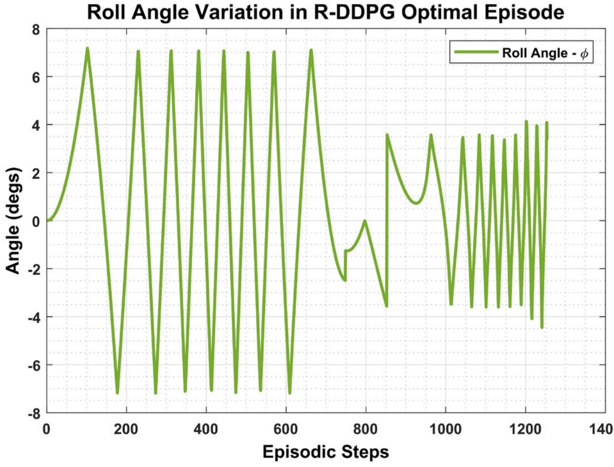

Fig. 5 Episodic variation of UAV roll angle

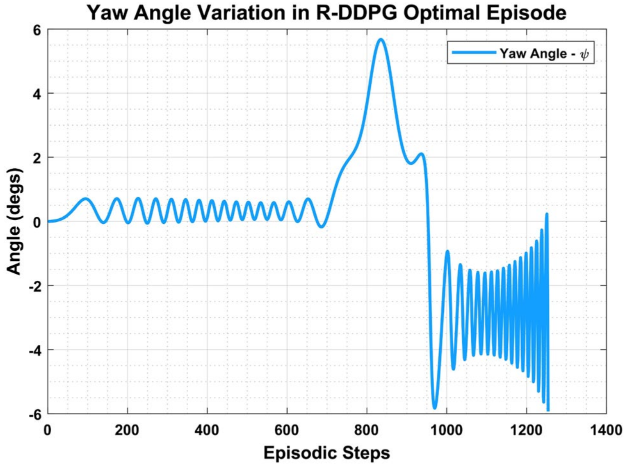

Fig. 7 Episodic variation of UAV yaw angle

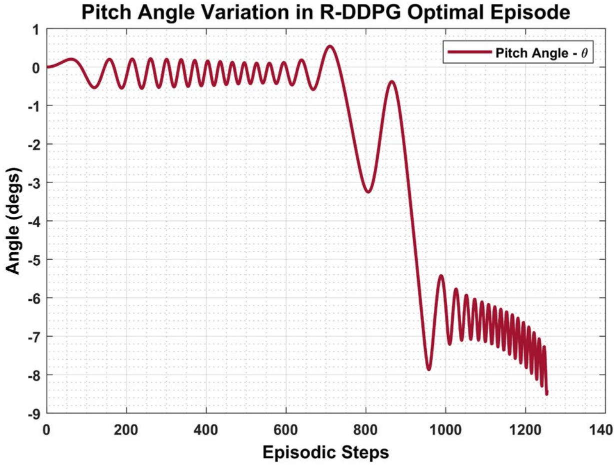

Fig. 6 Episodic variation of UAV pitch angle

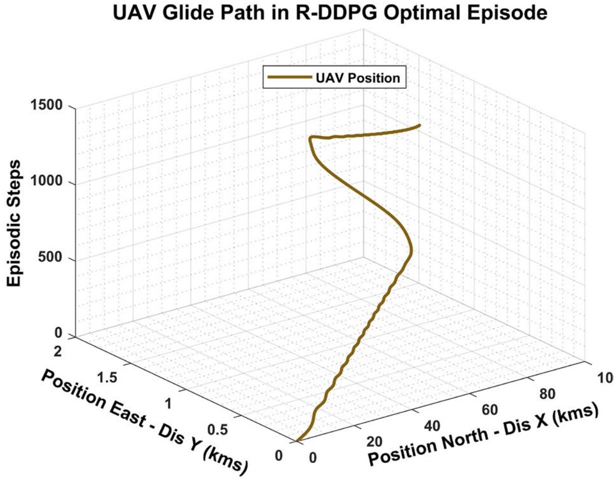

Fig. 8 UAV episodic glide path

the action policy after initially following learnt policy for a good reward.

Variation in the attitude of the UAV as observed in the Earth's frame during the complete episodic flight is shown in Figs. 5, 6 and 7. During half of the initial flight phase, pitch and yaw angles are restricted close to 1°. Due to sinusoidal behavior in the roll rate, UAV exhibits a wing rocking behaviour for initial part of the episode. Owing to the inherent complex geometry of the platform the coupled roll and yaw dynamics are complex. With agent learning to optimally trade-off rates with time, it gradually decreases the roll angle variation, however, pitch and yaw angles continue to show variation. 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.

Similarly, Fig. 8 depicts the glide path of the UAV which initially covers more distance north wards but with increasing yaw angle variation the UAV follows the desired east ward direction.

Figures 9 and 10 exhibit variation of UAV's aerodynamic angles alpha and beta respectively, during the entire episodic flight. While the sideslip is contained between $0.5^\circ \ to - 1^\circ$, the angle of attack initially increases to gain more lift, later maintains it close to 6° for achieving the desired objective of range enhancement.

Figure. 11 gives a complete overview of the variation of UAV velocity profile. Velocity of the platform decreases gradually and smoothly as a result of increased drag due to increase in alpha in the major part of the episode. However, during the later part of the flight, velocity decreases significantly with the increase in yaw angle and thus sideslip, profoundly increasing drag.

Altitude variation as depicted in Fig. 12 is smooth and gradual in the initial part where the angle of attack is

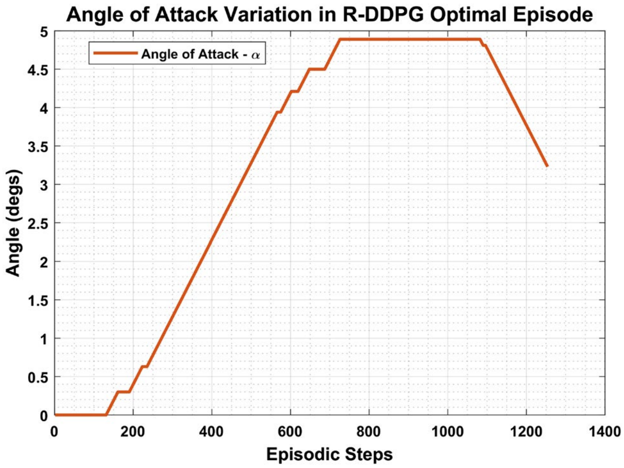

Fig. 9 Episodic variation of UAV alpha angle

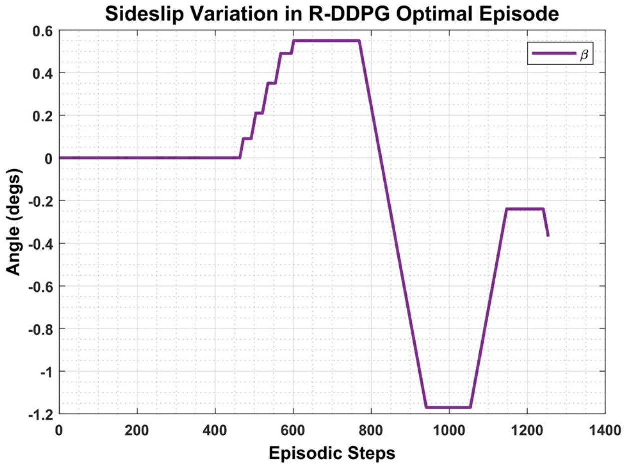

Fig. 10 Episodic variation of UAV beta angle

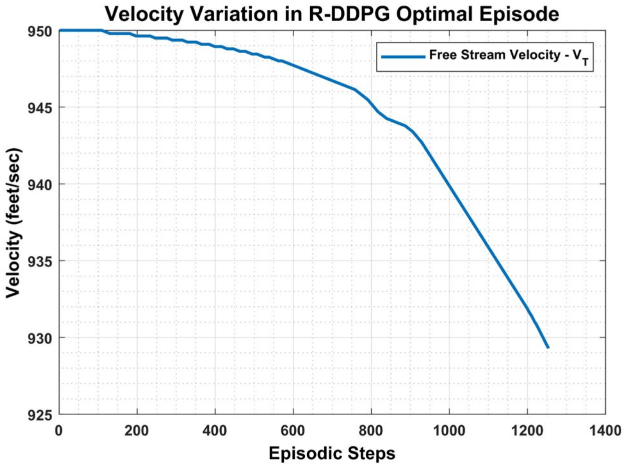

Fig. 11 Episodic variation of UAV velocity

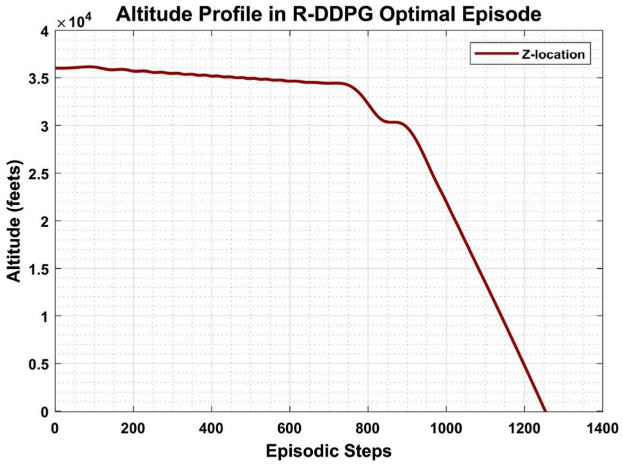

Fig. 12 Episodic variation of UAV altitude

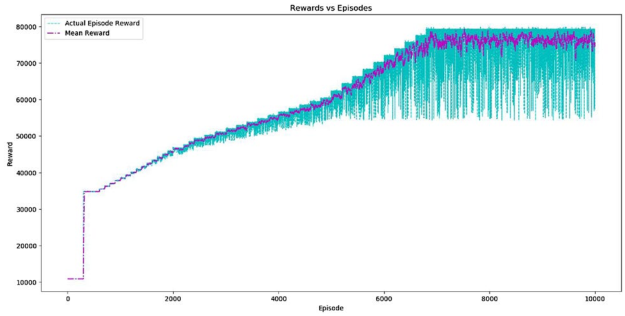

Fig. 13 Reward function of R-DDPG agent

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.

Reward function variation is shown in Fig. 13. 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.

All the 12 states of the gliding vehicle have been plotted against episodic steps. The simulation step is kept at 0.1 as it yields optimum results for all the states keeping in view the quantum of change of states. Besides states depiction, the reward function is also plotted which is a measure of the learning performance of NNs for R-DDPG.

6 Comparative analysis: proposed R-DDPG controller vs conventional LQR control & DRL optimal proximal policy optimization (O-PPO) control architecture

In this section, we preform the comparison of the results achieved from the proposed R-DDPG non-linear algorithm with conventional PID based control architecture and another DRL method named Optimized proximal policy optimization (O-PPO). Results of experimental UAV's optimized trajectory and enhanced range being controlled with the help of devised DRL based intelligent R-DDPG controller were compared with that of a linear classical LQR based controller, to compare the performance of an intelligent control with that of the classical one.

6.1 Formulation of LQR based control architecture

In case of linear LQR based control architecture, steady state values for the state and control variables are determined at each point of a pre-specified trajectory by computing an equilibria point of the differential equations. The key idea includes linearizing the nonlinear system along the trajectory, then using the resulting time-varying linearization to obtain a time-varying state feedback controller that locally stabilizes the system along the trajectory. The optimization techniques similar to Gul et al. (2021a), Mir et al. (2017b, 2017c, 2018d, 2021a) enhanced UAV range to about 85 km with high accuracy. The problem is configured as a constrained optimization problem (Gul et al. 2021c, b, d; Mir et al. 2019a), with an objective to determine, an open loop control that optimizes the specified performance index, subject to certain constraints (Mir et al. 2017b, 2018a, b, d). For optimization, Matlab$^{\text{registered}}$ nonlinear constrained optimization technique, based on Sequential Quadratic Programming (SQP) and quasi-Newton methods is utilized. Steady states for the optimized trajectory were obtained for a coordinated turn flight at different turn rates as shown in Eq. (14):

$$Turn\ rate\,(\dot{\psi}) = [-10, -7, -5, -3, -2, 1, 0, 1, 2, 3, 4, 5, 10]^T \\ \text{deg}\,/sec. \tag{14}$$

In order to generate the optimized trajectories at the desired turn rates (as specified in Eq. (14)), the performance measure that is minimized is defined in Eq. (15):

$$J_{min} = w_1\dot{V}_T + w_2\dot{\alpha} + w_3\dot{\beta} + w_4\dot{p} + w_5\dot{q} + w_6\dot{r} \tag{15}$$

where $w_1 ... w_6 = 1$ and $\dot{\alpha}$, $\dot{\beta}$, $\dot{p}$, $\dot{q}$, $\dot{r}$ are the rate derivatives of velocity, angle of attack, side slip angle and roll,pitch and yaw rates respectively. This cost function is minimized by the optimization algorithm at each equilibria point of the pre-specified trajectory governed by the differential equations.

The state and control variables utilized during the optimization process are defined in Eq. (16):

$$\begin{aligned} \boldsymbol{x} &= \left[V_T, \alpha, \beta, \phi, \theta, \psi, p, q, r, P_n, P_e, h\right]^T, \quad \boldsymbol{x} \in \mathbb{R}^{12} \\ \boldsymbol{u} &= [LCF, RCF]^T, \quad \boldsymbol{u} \in \mathbb{R}^2 \end{aligned} \tag{16}$$

where LCF and RCF are the control variables.

Path constraints along with the bounds on the state and control variables are defined in Eq. (17):

$$\begin{aligned} &0 \le h \le 35000ft,\ -3^\circ < \alpha < 6^\circ,\ -6^\circ < \beta < 6^\circ, \\ &\quad Mach \le 0.75,\ 100\ km \le Range \le 120\ km. \end{aligned} \tag{17}$$

Terminal constraints are defined in Eqs. (18):

$$\begin{aligned} &P_n(te) - P_{n_{(te)}} \le \Delta P_n, \\ &P_e(te) - P_{e_{(te)}} \le \Delta P_e, \\ &h(te) - h_{te} \le \Delta h, \end{aligned} \tag{18}$$

where $P_n(te), P_e(te), h(te)$ are the UAV coordinates at the terminal point, $P_{n_{(te)}}, P_{e_{(te)}}, h_{te}$ are the target coordinates, and $\Delta P_n, \Delta P_e, \Delta h$ are the permissible tolerances.

According to the model assumptions, the orientation of UAV at any point *'b'* can be described in terms of earlier point *'a'*, by Eq. (19):

$$\boldsymbol{x}(t_a) = \boldsymbol{x}(t_a) + \int_{t_a}^{t_b} \dot{\boldsymbol{x}}(t)\,dt, \tag{19}$$

where $\boldsymbol{x}(t_b)$ and $\boldsymbol{x}(t_a)$ represents the state variables at time t_b and t_a respectively.

An LQR control based optimization framework was therefore formulated which utilizes a set of dynamic constraints, path constraints and terminal constraints (Eq. (18)), while minimizing the performance measure represented by Eq. (15). As a result of the optimization process, steady state values for the state and control variables along various optimal trajectories (governed by Eq. 14) were obtained. The optimization process utilized the optimal flight profile parameters (velocity, dynamic pressure, and mach number).

During optimization process, certain states and variables were kept fixed along the trim points of the optimal ballistic trajectories while other states and control variables were kept free for optimization (Eq. (20)) within the permissible ranges.

$$\begin{aligned} Fixed\_states &= \left[V_T, h, \dot{\phi}, \dot{\theta}, \dot{\psi}\right] \\ Free\_states\&Controls &= [\alpha, \beta, \gamma, \phi, \theta, p, q, r, LCF, RCF] \end{aligned} \tag{20}$$

As a result of the optimization, optimal trim values of the state and control variables were ascertained along trim points for the entire flight envelope (35,000-ground level). This included determining optimal trajectories for different turn rates governed by Eq. (14).

6.2 Comparison of the results: R-DDPG Vs LQR control architecture

In the designed state feed-back LQR based control architecture, problem of deviation from the desired course was encountered. In spite of having stable system response with theoretical range enhancement to about 85 km, the UAV has a considerable off shoot from the target location. This is shown in Fig. 14. The deviation from desired course was encountered as the designed control law did not utilize the directional information (ψ) provided by navigational equation. Both the longitudinal and lateral dynamics were independent from directional orientation provided by navigational equation. Accurate guidance to the target, therefore, requires incorporation of an additional control loop which minimizes the drift in the heading.

However, no such problem of accuracy was encountered in the DDPG based control architecture. The UAV was successfully guided to the desired target location with reduced CEP. This is shown in Figure. 15.

It is evident that R-DDPG flight controller after incorporation of the final reward function and by virtue of its off-line neural networks based learning, intelligently selects optimal control deflections at every episodic step thus resulting in optimal glide range of **83 *kms***. Further, R-DDPG controller smoothly covered the entire flight envelope while classical optimal controller was applicable for certain equilibrium points and gain scheduling was manually performed to handle the complex dynamics of the UAV besides causing controllability problems in case of environmental disturbances while R-DDPG controller intelligently rejected the external disturbances. Due same, accuracy problem was encountered in LQR based architecture with much higher Circular Error Probabilities (CEPs).

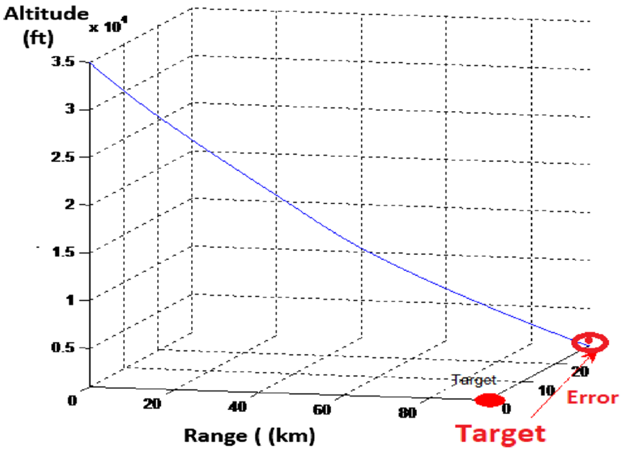

Fig. 14 UAV trajectory: LQR based stabilization control architecture

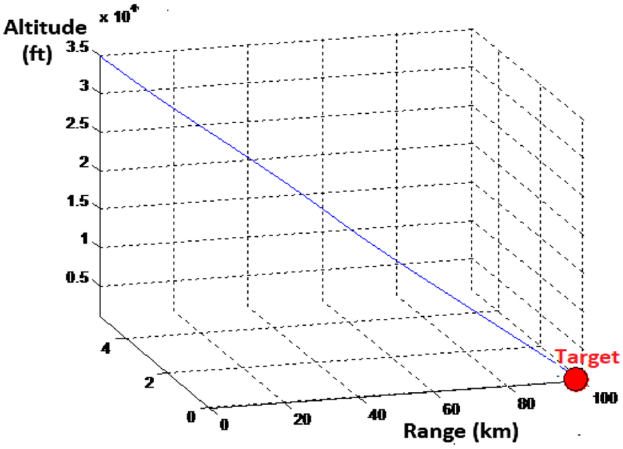

Fig. 15 UAV trajectory: R-DDPG based control architecture

6.3 Optimized proximal policy optimization (O-PPO) Implementation and Comparative Analysis

Besides comparing the performance of the implemented R-DDPG algorithm in the previous section with LQR based controller, an effort was made to compare it with one of widely utilized DRL based methods so as to gauge and mark true comparison of the efficacy of DDPG. Keeping in view the nature of problem in hand, two **On Policy** algorithms namely **TRPO** and **O-PPO** were also studied. In this research, O-PPO was selected in comparison to TRPO as O-PPO is faster and more sample efficient besides it is much simpler to implement. In O-PPO with its two variants **O-PPO penalty** and **O-PPO Clipping**, O-PPO Clipping was selected as it ignores the KL-divergence term in the objective function and doesn't have any constraint. Alternatively, it relies on specialized clipping in the objective function to eradicate incentives for the new policy θ_{new} to get far from the old policy θ_{old} and this is primarily for the exploration phase. Clipping was manged by setting the upper and lower limits as $1+\epsilon$ and $1-\epsilon$, respectively with ϵ as 0.99. For optimization Adam was again utilized because of its lowest training cost. Learning rates for both actor and critic were set as 0.001. Batch size as 64 and Trajectory memory as 200.

Performance of O-PPO (refer Algorithm 2) in convergence is slightly better than R-DDPG. However, O-PPO being more susceptible to changes in the environment as discussed in Section 3 above, was not able to get good return for the current problem as evident from the roll, pitch and yaw rate graphs (refer Figs. 16, 17 and 18). As evident from these results the rates are initially controlled, but the

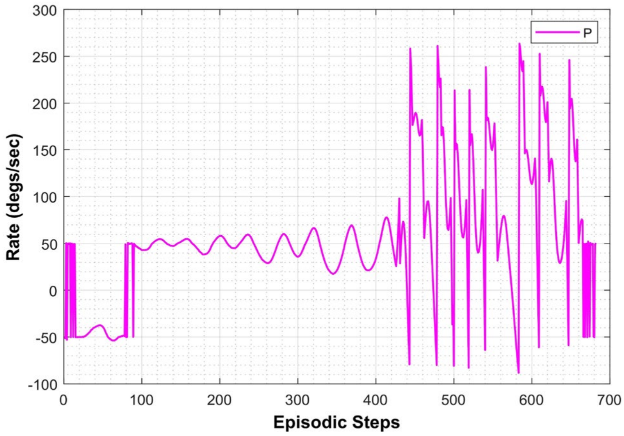

Fig. 16 UAV Roll Rate: O-PPO based controller

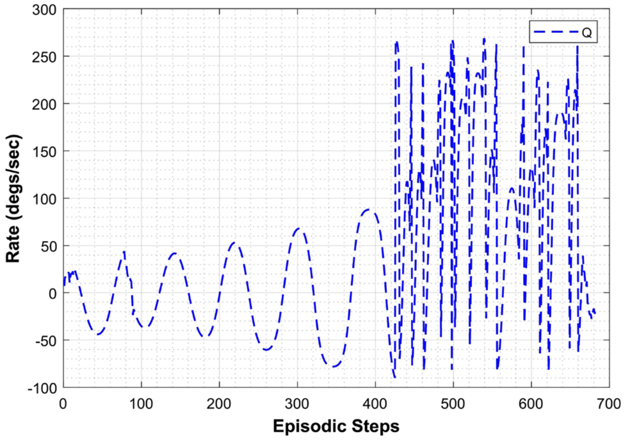

Fig. 17 UAV Pitch Rate: O-PPO based controller

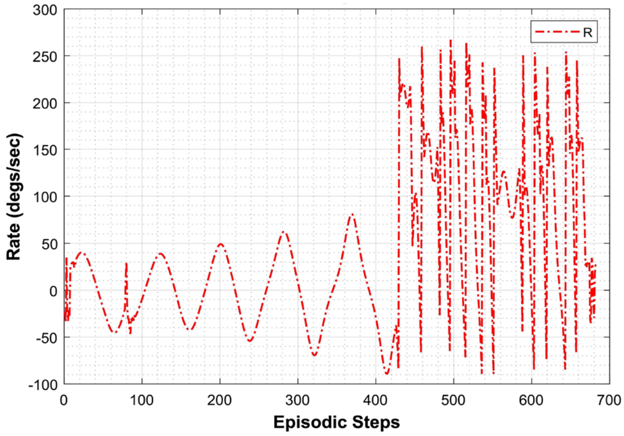

Fig. 18 UAV Yaw Rate : O-PPO based controller

moment O-PPO tries to clip the surrogate objective function to ensure that the θ_{new} and θ_{old} are markedly different that is to have an increased step size for enhanced learning, the problem of low return commences as the agents learning deviates from intended objectives and vehicles inherent roll and yaw coupling starts to get more prominent.

Algorithm 2 O-PPO Clip Algorithm

```
Input: initial policy parameters $\theta_0$, initial value function parameters $\phi_0$
for k = 0,1,2,.... do
    Collect set of trajectories $D_k = \tau_i$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
    Compute rewards-to-go $\hat{R}_t$
    Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function $V_{\phi_k}$
    Update the policy by maximizing the O-PPO Clip objective: $\theta_{k+1} = \arg\max_\theta \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} min \left( \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t)(a_t), g(\in, A^{\pi_{\theta_k}})(s_t)(a_t)) \right)$ typically via stochastic gradient ascent with Adam.
    Fit value function by regression on mean-squared error: $\phi_{k+1} = \arg\min_\theta \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \left( V_\phi(s_t) - \hat{R}_t \right)^2$ typically via some gradient descent algorithm.
```

7 Conclusion

In current research, DRL based intelligent non-linear controller for an experimental glide UAV was proposed utilizing R-DDPG (modified version of DDPG algorithm) while incorporating an optimal reward function. Implemented control algorithm showed promising results in achieving the primary objective of maximizing the range while keeping the platform stable within its design constraints throughout the flight regime. R-DDPG approach gave the optimal range of around **83** ***kms***; while handling the non-linearity (controlling the roll, pitch and yaw rates in an optimal trade-off) of vehicle through effective control deflections which were being controlled based on the off-line learning of the agent based on the optimized incorporated reward function.

Devised RL algorithm proved to be computationally acceptable, wherein the agent was successfully trained for large continuous state and action space. The performance of the controller was evaluated in a **6-DoF** simulation developed with the help of Matlab and Flight gear software. R-DDPG controller outperformed the LQR based optimal controller as being effective in the entire flight regime of the vehicle thus disregarding the conventional approaches of calculating various equilibrium's during the trajectory, and then trying to keep the vehicle stable within the ambit of these equilibria utilizing linear / non-linear approaches.

Besides LQR, another DRL algorithm best known for its simple implementation and efficient convergence O-PPO

Clipping was also evaluated in the research. It is noteworthy that O-PPO as expected gave better and quicker convergence but was unable to handle most uncertainties of the environment in the form of varying launch conditions. Further probe into this behaviour of O-PPO agent, led to two main conclusions. First, R-DDPG is able to benefit from old data while O-PPO cannot. This is because O-PPO uses a ratio of the policies to limit the step size, while R-DDPG uses the policy to predict the action for the value computed by the critic. It is because of this that the ratio of the policies in O-PPO, limits the distance between the policies and therefor needs fresh data. Second important conclusion inferred from the comparison of two DRL methods and which has been practically observed, that behaviour of deep policy gradient algorithms, often deviates from the prediction of their original motivating framework and therefore their success varies for different problems primarily because of different environments and changing success objectives.

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. It is noteworthy that this research will open the avenues for researchers for designing intelligent control systems for aircraft, UAVs and the autonomous control of missile trajectories for both powered and un-powered configurations.

References

Aboelezz A, Mohamady O, Hassanalian M, Elhadidi B (2021) Nonlinear flight dynamics and control of a fixed-wing micro air vehicle: numerical, system identification and experimental investigations. J Intell Robot Syst 101(3):1–18

Adams RJ, Banda SS (1993) Robust flight control design using dynamic inversion and structured singular value synthesis. IEEE Trans Control Syst Technol 1(2):80–92

Adams RJ, Buffington JM, Banda SS (1994) Design of nonlinear control laws for high-angle-of-attack flight. J Guid Control Dyn 17(4):737–746

Adams RJ, Buffington JM, Sparks AG, Banda SS (2012) Robust multivariable flight control. Springer Science & Business Media, Berlin

Araar O, Aouf N (2014) Full linear control of a quadrotor uav, lq vs hinf. In: 2014 UKACC International Conference on control (CONTROL), IEEE, pp 133–138

Azar AT, Koubaa A, Ali Mohamed N, Ibrahim HA, Ibrahim ZF, Kazim M, Ammar A, Benjdira B, Khamis AM, Hameed IA et al (2021) Drone deep reinforcement learning: a review. Electronics 10(9):999

Bansal T, Pachocki J, Sidor S, Sutskever I, Mordatch I (2017) Emergent complexity via multi-agent competition. arXiv preprint arXiv:1710.03748

Bouhamed O, Ghazzai H, Besbes H, Massoud Y (2020) Autonomous UAV navigation: a DDPG-based deep reinforcement learning approach. In: 2020 IEEE International Symposium on circuits and systems (ISCAS), vol 1. IEEE, pp 1–5

Brière D, Traverse P (1993) Airbus a320/a330/a340 electrical flight controls-a family of fault-tolerant systems. In: FTCS-23 The Twenty-Third International Symposium on fault-tolerant computing, vol 1. IEEE, pp 616–623

Buning PG, Gomez RJ, Scallion WI (2004) Cfd approaches for simulation of wing-body stage separation. AIAA Paper 4838:2004

Cai G, Dias J, Seneviratne L (2014) A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends. Unmanned Syst 2(02):175–199

Chen J, Xiao Z, Xing H, Dai P, Luo S, Iqbal MA (2020) Stdpg: a spatio-temporal deterministic policy gradient agent for dynamic routing in sdn. In: ICC 2020-2020 IEEE International Conference on communications (ICC), vol 1. IEEE, pp 1–6

Chen J, Xing H, Xiao Z, Xu L, Tao T (2021) A drl agent for jointly optimizing computation offloading and resource allocation in mec. IEEE Internet Things J 8(24):17508–17524

Chowdhary G, Frazzoli E, How J, Liu H (2014) Nonlinear flight control techniques for unmanned aerial vehicles. In: Handbook of unmanned aerial vehicles, Springer, Houten

CS231n S (2017) Convolutional neural networks for visual recognition. https://www.cs231ngithubio/neural-networks-3/#baby. Accessed 1 Sept 2020

Dalal G, Dvijotham K, Vecerik M, Hester T, Paduraru C, Tassa Y (2018) Safe exploration in continuous action spaces. arXiv preprint arXiv:1801.08757

Derafa L, Ouldali A, Madani T, Benallegue A (2011) Non-linear control algorithm for the four rotors uav attitude tracking problem. Aeronaut J 115(1165):175–185

Din AFU, Mir I, Gul F, Mir S, Alhady SSN, Nasar A, Rustom M, Alkhazaleh HA, Abualigah L (2022) Robust flight control system design of a fixed wing uav using optimal dynamic programming. Soft Comput. https://doi.org/10.1007/s00500-022-07484-z

Dong Y, Zou X (2020) Mobile robot path planning based on improved ddpg reinforcement learning algorithm. In: 2020 IEEE 11th International Conference on software engineering and service science (ICSESS), vol 1. IEEE, pp 52–56

Doyle J, Lenz K, Packard A (1987) Design examples using μ-synthesis: space shuttle lateral axis fcs during reentry. In: Modelling, robustness and sensitivity reduction in control systems. Springer, pp 127–154

Dutoi B, Richards N, Gandhi N, Ward D, Leonard J (2008) Hybrid robust control and reinforcement learning for optimal upset recovery. In: AIAA Guidance, Navigation and Control Conference and Exhibit, vol 1. p 6502

Elmeseiry N, Alshaer N, Ismail T (2021) A detailed survey and future directions of unmanned aerial vehicles (uavs) with potential applications. Aerospace 8(12):363

Enomoto K, Yamasaki T, Takano H, Baba Y (2013) Guidance and control system design for chase uav. In: AIAA Guidance, Navigation and Control Conference and exhibit, vol 1. p 6842

Escareno J, Salazar-Cruz S, Lozano R (2006) Embedded control of a four-rotor uav. In: 2006 American Control Conference, IEEE, vol 1. pp 6–pp

Finck R, (US) AFFDL, Hoak D (1978) USAF stability and control DATCOM. Eng Doc

Giordan D, Adams MS, Aicardi I, Alicandro M, Allasia P, Baldo M, De Berardinis P, Dominici D, Godone D, Hobbs P et al (2020) The use of unmanned aerial vehicles (uavs) for engineering geology applications. Bull Eng Geol Env 79(7):3437–3481

Golian M, Katibeh H, Singh VP, Ostad-Ali-Askari K, Rostami HT (2020) Prediction of tunnelling impact on flow rates of adjacent extraction water wells. Q J Eng Geol Hydrogeol 53(2):236–251

Gul F, Rahiman W (2019) An integrated approach for path planning for mobile robot using bi-rrt. In: IOP Conference Series: materials science and engineering, IOP Publishing, vol 697, p 012022

Gul F, Rahiman W, Nazli Alhady SS (2019) A comprehensive study for robot navigation techniques. Cogent Eng 6(1):1632046

Gul F, Alhady SSN, Rahiman W (2020a) A review of controller approach for autonomous guided vehicle system. Indones J Electr Eng Comput Sci 20(1):552–562

Gul F, Rahiman W, Alhady SN, Ali A, Mir I, Jalil A (2020b) 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 pp 1–18

Gul F, Mir I, Abualigah L, Sumari P (2021a) Multi-robot space exploration: an augmented arithmetic approach. IEEE Access 9:107738–107750

Gul F, Mir I, Abualigah L, Sumari P, Forestiero A (2021b) A consolidated review of path planning and optimization techniques: technical perspectives and future directions. Electronics 10(18):2250

Gul F, Mir I, Rahiman W, Islam TU (2021c) Novel implementation of multi-robot space exploration utilizing coordinated multi-robot exploration and frequency modified whale optimization algorithm. IEEE Access 9:22774–22787

Gul F, Rahiman W, Alhady S, Ali A, Mir I, Jalil A (2021d) 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 12(7):7873–7890

Gul F, Mir I, Abualigah L, Mir S, Altalhi M (2022a) Cooperative multi-function approach: a new strategy for autonomous ground robotics. Future Gener Comput Syst 134:361–373

Gul F, Mir I, Alarabiat D, Alabool HM, Abualigah L, Mir S (2022b) Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning. J Parallel Distrib Comput 169:171–184

Gul F, Mir S, Mir I (2022c) Coordinated multi-robot exploration: hybrid stochastic optimization approach. In: AIAA SCITECH 2022 Forum, p 1414

Gul F, Mir S, Mir I (2022d) Multi robot space exploration: a modified frequency whale optimization approach. In: AIAA SCITECH 2022 Forum, p 1416

Hafner R, Riedmiller M (2011) Reinforcement learning in feedback control. Mach Learn 84(1–2):137–169

Heess N, Hunt JJ, Lillicrap TP, Silver D (2015) Memory-based control with recurrent neural networks. arXiv preprint arXiv:1512.04455

Henderson P, Islam R, Bachman P, Pineau J, Precup D, Meger D (2018) Deep reinforcement learning that matters. In: Proceedings of the AAAI Conference on artificial intelligence, vol 32. pp 1–12

Hou Z, Lu P, Tu Z (2020) Nonsingular terminal sliding mode control for a quadrotor uav with a total rotor failure. Aerosp Sci Technol 98:105716

Hu H, Ql Wang (2020) Proximal policy optimization with an integral compensator for quadrotor control. Front Inf Technol Electron Eng 21(5):777–795

Hussain A, Hussain I, Mir I, Afzal W, Anjum U, Channa BA (2020) Target parameter estimation in reduced dimension stap for airborne phased array radar. In: 2020 IEEE 23rd International Multitopic Conference (INMIC), IEEE, pp 1–6

Hussain A, Anjum U, Channa BA, Afzal W, Hussain I, Mir I (2021) Displaced phase center antenna processing for airborne phased array radar. In: 2021 International Bhurban Conference on applied sciences and technologies (IBCAST), IEEE, pp 988–992

Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285

Kim D, Oh G, Seo Y, Kim Y (2017) Reinforcement learning-based optimal flat spin recovery for unmanned aerial vehicle. J Guid Control Dyn 40(4):1076–1084

Kimathi S (2017) Application of reinforcement learning in heading control of a fixed wing uav using x-plane platform

Kinga DA (2015) A method for stochastic optimization. In: Anon International Conference on learning representations, San Diego: ICLR

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980

Koch W, Mancuso R, West R, Bestavros A (2019) Reinforcement learning for uav attitude control. ACM Trans Cyber-Phys Syst 3(2):1–21

Kroezen D (2019) Online reinforcement learning for flight control: an adaptive critic design without prior model knowledge

Kulcsar B (2000) Lqg/ltr controller design for an aircraft model. Period Polytech Transp Eng 28(1–2):131–142

Labbadi M, Cherkaoui M (2019) Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor uav. Aerosp Sci Technol 93:105306

Laroche R, Feraud R (2017) Reinforcement learning algorithm selection. arXiv preprint arXiv:1701.08810

Lei C (2021) Deep reinforcement learning. In: Deep learning and practice with MindSpore, Springer, pp 217–243

Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971

Lin X, Liu J, Yu Y, Sun C (2020) Event-triggered reinforcement learning control for the quadrotor uav with actuator saturation. Neurocomputing 415:135–145

Luo X, Zhang Y, He Z, Yang G, Ji Z (2019) A two-step environment-learning-based method for optimal uav deployment. IEEE Access 7:149328–149340

Mir I, Maqsood A, Akhtar S (2017a) Dynamic modeling & stability analysis of a generic uav in glide phase. In: MATEC Web of Conferences, EDP Sciences, vol 114, p 01007

Mir I, Maqsood A, Akhtar S (2017b) Optimization of dynamic soaring maneuvers to enhance endurance of a versatile uav. In: IOP Conference series: materials science and engineering, IOP Publishing, vol 211, p 012010

Mir I, Maqsood A, Akhtar S (2017c) Optimization of dynamic soaring maneuvers to enhance endurance of a versatile uav. In: IOP Conference series: materials science and engineering, IOP Publishing, vol 211, p 012010

Mir I, Eisa SA, Maqsood A (2018a) Review of dynamic soaring: technical aspects, nonlinear modeling perspectives and future directions. Nonlinear Dyn 94(4):3117–3144

Mir I, Maqsood A, Akhtar S (2018b) Biologically inspired dynamic soaring maneuvers for an unmanned air vehicle capable of sweep morphing. Int J Aeronaut Sp Sci 19(4):1006–1016

Mir I, Maqsood A, Eisa SA, Taha H, Akhtar S (2018c) Optimal morphing-augmented dynamic soaring maneuvers for unmanned air vehicle capable of span and sweep morphologies. Aerosp Sci Technol 79:17–36

Mir I, Taha H, Eisa SA, Maqsood A (2018d) A controllability perspective of dynamic soaring. Nonlinear Dyn 94(4):2347–2362

Mir I, Akhtar S, Eisa S, Maqsood A (2019a) Guidance and control of standoff air-to-surface carrier vehicle. Aeronaut J 123(1261):283–309

Mir I, Maqsood A, Taha HE, Eisa SA (2019b) Soaring energetics for a nature inspired unmanned aerial vehicle. In: AIAA Scitech 2019 Forum, p 1622

Mir I, Eisa SA, Taha H, Maqsood A, Akhtar S, Islam TU (2021a) A stability perspective of bio-inspired uavs performing dynamic soaring optimally. Bioinspir Biomim 16(6):066010

Mir I, Eisa SA, Taha H, Maqsood A, Akhtar S, Islam TU (2021b) A stability perspective of bioinspired unmanned aerial vehicles performing optimal dynamic soaring. Bioinspir Biomim 16(6):066010

Mir I, Eisa S, Maqsood A, Gul F (2022a) Contraction analysis of dynamic soaring. In: AIAA SCITECH 2022 Forum, p 0881

Mir I, Eisa S, Taha HE, Gul F (2022b) On the stability of dynamic soaring: Floquet-based investigation. In: AIAA SCITECH 2022 Forum, p 0882

Mir I, Gul F, Mir S, Khan MA, Saeed N, Abualigah L, Abuhaija B, Gandomi AH (2022c) A survey of trajectory planning techniques for autonomous systems. Electronics 11(18):2801

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533

Napolitano MR, An Y, Seanor BA (2000) A fault tolerant flight control system for sensor and actuator failures using neural networks. Aircraft Design 3(2):103–128

Nikolakopoulos KG, Soura K, Koukouvelas IK, Argyropoulos NG (2017) Uav vs classical aerial photogrammetry for archaeological studies. J Archaeol Sci Rep 14:758–773

Novati G, Mahadevan L, Koumoutsakos P (2018) Deep-reinforcement-learning for gliding and perching bodies. arXiv preprint arXiv:1807.03671

Nurbani ES (2018) Environmental protection in international humanitarian law. Unram Law Rev 2(1):1–12

Ostad-Ali-Askari K, Shayannejad M, Ghorbanizadeh-Kharazi H (2017) Artificial neural network for modeling nitrate pollution of groundwater in marginal area of zayandeh-rood river, isfahan, iran. KSCE J Civ Eng 21(1):134–140

Pan J, Wang X, Cheng Y, Yu Q (2018) Multisource transfer double dqn based on actor learning. IEEE Trans Neural Netw Learn Syst 29(6):2227–2238

Paucar C, Morales L, Pinto K, Sánchez M, Rodríguez R, Gutierrez M, Palacios L (2018) Use of drones for surveillance and reconnaissance of military areas. In: International Conference of research applied to defense and security, Springer, pp 119–132

Peng K (2021) Autonomous mission management based nonlinear flight control design for a class of hybrid unmanned aerial vehicles. Guid Navig Control 1(02):2150009

Petterson K (2006) Cfd analysis of the low-speed aerodynamic characteristics of a ucav. AIAA Paper 1259:2006

Pi CH, Ye WY, Cheng S (2021) Robust quadrotor control through reinforcement learning with disturbance compensation. Appl Sci 11(7):3257

Pirnazar M, Hasheminasab H, Karimi AZ, Ostad-Ali-Askari K, Ghasemi Z, Haeri-Hamedani M, Mohri-Esfahani E, Eslamian S (2018) The evaluation of the usage of the fuzzy algorithms in increasing the accuracy of the extracted land use maps. Int J Glob Environ Issues 17(4):307–321

Poksawat P, Wang L, Mohamed A (2017) Gain scheduled attitude control of fixed-wing uav with automatic controller tuning. IEEE Trans Control Syst Technol 26(4):1192–1203

Rastogi D (2017) Deep reinforcement learning for bipedal robots, Student Thesis

Rinaldi F, Chiesa S, Quagliotti F (2013) Linear quadratic control for quadrotors uavs dynamics and formation flight. J Intell Robot Syst 70(1–4):203–220

Roaskam J (2001) Airplane flight dynamics and automatic flight controls. vol

Rodriguez-Ramos A, Sampedro C, Bavle H, De La Puente P, Campoy P (2019) A deep reinforcement learning strategy for uav autonomous landing on a moving platform. J Intell Robot Syst 93(1–2):351–366

Rosales C, Soria CM, Rossomando FG (2019) Identification and adaptive pid control of a hexacopter uav based on neural networks. Int J Adapt Control Signal Process 33(1):74–91

Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747

Schulman J, Levine S, Abbeel P, Jordan M, Moritz P (2015) Trust region policy optimization. In: International Conference on machine learning, PMLR, pp 1889–1897

Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347

Silver D (2015) Lecture 2: Markov decision processes. UCL Retrieved from www.0csuclacuk/staff/dsilver/web/Teaching_files/MDPpdf

Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M (2014) Deterministic policy gradient algorithms

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489

Sutton RS, Barto AG (1998) Planning and learning. In: Reinforcement learning: an introduction., ser. Adaptive computation and machine learning. A Bradford Book, pp 227–254

Szczepanski R, Tarczewski T, Grzesiak LM (2019) Adaptive state feedback speed controller for pmsm based on artificial bee colony algorithm. Appl Soft Comput 83:105644

Szczepanski R, Bereit A, Tarczewski T (2021) Efficient local path planning algorithm using artificial potential field supported by augmented reality. Energies 14(20):6642

Szczepanski R, Tarczewski T (2021) Global path planning for mobile robot based on artificial bee colony and dijkstra's algorithms. In: 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), IEEE, pp 724–730

Tal E, Karaman S (2020) Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. IEEE Trans Control Syst Technol 29(3):1203–1218

Tang Y (2016) Tf. learn: tensorflow's high-level module for distributed machine learning. arXiv preprint arXiv:1612.04251

Thorndike EL (1898) Animal intelligence. Nature 58(1504):390

Verma S (2020) A survey on machine learning applied to dynamic physical systems. arXiv preprint arXiv:2009.09719

Wang S, Jia D, Weng X (2018) Deep reinforcement learning for autonomous driving. arXiv preprint arXiv:1811.11329

Werbos PJ, Miller W, Sutton R (1990) A menu of designs for reinforcement learning over time. In: Neural networks for control, pp 67–95

Wickenheiser AM, Garcia E (2008) Optimization of perching maneuvers through vehicle morphing. J Guid Control Dyn 31(4):815–823

Winkler S, Zeadally S, Evans K (2018) Privacy and civilian drone use: The need for further regulation. IEEE Secur Privacy 16(5):72–80

Wu Y, Mansimov E, Grosse RB, Liao S, Ba J (2017) Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In: Advances in neural information processing systems, p 30

Xenou K, Chalkiadakis G, Afantenos S (2018) Deep reinforcement learning in strategic board game environments. In: European Conference on multi-agent systems, Springer, pp 233–248

Xiao Z, Xu X, Xing H, Luo S, Dai P, Zhan D (2021) Rtfn: a robust temporal feature network for time series classification. Inf Sci 571:65–86

Xu D, Hui Z, Liu Y, Chen G (2019) Morphing control of a new bionic morphing uav with deep reinforcement learning. Aerosp Sci Technol 92:232–243

Yang Y, Zhu J, Yang J (2020) Indi-based transitional flight control and stability analysis of a tail-sitter uav. In: 2020 IEEE International Conference on systems, man, and cybernetics (SMC), IEEE, pp 1420–1426

Yanushevsky R (2011) Guidance of unmanned aerial vehicles. CRC Press, Boca Raton

Zhou Y (2018) Online reinforcement learning control for aerospace systems

Zhou C, He H, Yang P, Lyu F, Wu W, Cheng N, Shen X (2019) Deep rl-based trajectory planning for aoi minimization in uav-assisted iot. In: 2019 11th International Conference on wireless communications and signal processing (WCSP), IEEE, pp 1–6

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.