



# Reinforced Learning-Based Robust Control Design for Unmanned Aerial Vehicle

Adnan Fayyaz Ud Din<sup>1</sup> · Imran Mir<sup>2</sup> · Faiza Gul<sup>2</sup> · Mohammad Rustom Al Nasar<sup>3</sup> · Laith Abualigah<sup>4,5</sup>

Received: 26 October 2021 / Accepted: 20 February 2022 / Published online: 24 March 2022  
© King Fahd University of Petroleum & Minerals 2022

## Abstract

Innovation in UAV design technologies over the last decade and a half has resulted in capabilities that flourished the development of unique and complex multi-mission capable UAVs. These emerging new distinctive designs of UAVs necessitate development of intelligent and robust Control Laws which are independent of inherent plant variations besides being adaptive to environmental changes for achieving desired design objectives. Current research focuses on development of a control framework which aims to maximize the glide range for an experimental UAV employing reinforcement learning (RL)-based intelligent control architecture. A distinct model-free RL technique, abbreviated as ‘MRL’, is suggested which is capable of handling UAV control complications while keeping the computation cost low. At core, the basic RL DP algorithm has been sensibly modified to cater for the continuous state and control space domains associated with the current problem. Review of the performance characteristics through analysis of the results indicates the prowess of the presented algorithm to dynamically adapt to the changing environment, thereby making it suitable for complex designed UAV applications. Nonlinear simulations carried out under varying environmental conditions illustrated the effectiveness of the proposed methodology and its success over the conventional classical approaches.

**Keywords** Unmanned aerial vehicle · Intelligent technique · Dynamic programming · Reinforced learning

✉ Imran Mir  
imran.mir@aack.au.edu.pk

✉ Laith Abualigah  
aligah.2020@gmail.com

Adnan Fayyaz Ud Din  
adfdin@gmail.com

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

Mohammad Rustom Al Nasar  
mohammed.alnassar@aldar.ac.ae

<sup>1</sup> Institute of Avionics and Aeronautics, Air University, Islamabad, Pakistan

<sup>2</sup> Department of Avionics Engineering, Air University, Aerospace and Aviation, Campus Kamra, Islamabad, Pakistan

<sup>3</sup> School of Engineering and Technology, Department of Information Technology, ALDAR University College, Garhoud, Dubai, UAE

<sup>4</sup> Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

<sup>5</sup> School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia

## List of symbols

$b$	Wing span ( $m$ )
$\tilde{c}$	Mean aerodynamic chord ( $m$ )
$CAD$	Computer-aided design
$CFD$	Computational fluid dynamics
$C_{M_x}$	Coefficient of rolling moment
$C_{M_y}$	Coefficient of pitching moment
$C_{M_z}$	Coefficient of yawing moment
$C_{F_x}$	Force coefficient in the X-direction
$C_{F_y}$	Force coefficient in the Y-direction
$C_{F_z}$	Force coefficient in the Z-direction
$DoF$	Degree of freedom
$DDD$	Dull dirty and dangerous
$g$	Acceleration due to gravity ( $m/sec^2$ )
$h$	Altitude ( $m$ )
$LF$	Left-side control fin
$MRL$	Model-free reinforcement learning
$ML$	Machine learning
$m$	Mass of the vehicle ( $kg$ )
$P_E$	East position vector ( $km$ )
$P_N$	North position vector ( $km$ )
$P$	Roll rate ( $deg/sec$ )



$Q$ :	Pitch rate ( $deg/sec$ )
$R$ :	Yaw rate ( $deg/sec$ )
$RL$ :	Reinforcement learning
$RF$ :	Right-side control fin
$S$ :	Wing area ( $m^2$ )
$UAV$ :	Unmanned aerial vehicle
$V_T$ :	Far stream velocity ( $m/sec$ )
$n$ :	Numerical weights
$xpos$ :	Current X-position( $m$ )
$zpos$ :	Current Z-position( $m$ )
$r$ :	Momentary reward
$R$ :	Total reward
$pny$ :	Penalty

### Greek Symbol

$\alpha$ :	Angle of attack ( $deg$ )
$\beta$ :	Sideslip angle ( $deg$ )
$\gamma$ :	Flight path angle ( $deg$ )
$\psi$ :	Yaw angle ( $deg$ )
$\phi$ :	Roll angle ( $deg$ )
$\theta$ :	Theta angle ( $deg$ )
$\delta_L$ :	LF deflection ( $deg$ )
$\delta_R$ :	RF deflection ( $deg$ )
$\rho$ :	Air density ( $kg/m^3$ )

## 1 Introduction

UAVs are one of the most rapidly expanding and active divisions of the aviation business [1–6]. Unmanned aerial vehicles (UAVs) are useful in a variety of situations, such as search and rescue, monitoring, and exploration. As a result, UAVs need to be able to detect their trajectory quickly and accurately, especially in emergency situations or in a congested environment [7–10]. They are used in non-military applications such as search and rescue/health care, disaster management, journalism, shipping, engineering geology, and so on [11–16]. The demand is enormous and will continue to grow as new technologies become available. UAVs can also be effective with Internet of things (IoTs) components when used to perform sensing activities. UAVs, on the other hand, operate in a dynamic and uncertain environment due to their great mobility and shadowing in air to-ground channels. As a result, UAVs must increase the quality of their sensing and communication services without having compromising comprehensive information; therefore, reinforcement learning is a good fit for the cellular Internet of UAVs [17].

UAV models are now being developed in quite a large number and are acting as an indispensable aid to human operators in a wide range of military and civilian applications [7]. As a result, the fast growing fleet of UAVs, as well as the broadening scope of their applications, poses a

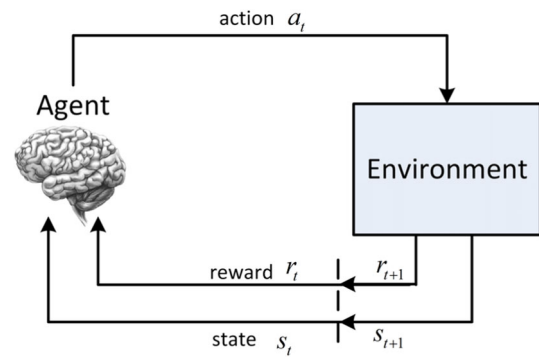


Fig. 1 Basic reinforcement learning framework

severe challenge to designers. The development of hi-fidelity systems was aided by technological improvements in the aviation [18–21] and ground transportation sectors [22–24].

Linear and nonlinear control systems have been utilized to solve a variety of control problems and obtain desired outcomes [4,5,5,9,10,20]. However, a thorough knowledge of these methodologies' inherent limitations became the driving force behind developing an intelligent system capable of making optimal, sequential decisions for a complex control situation.

Intelligent technologies, grouped under the banner of machine learning (ML), have begun to show promising results in resolving previously thought-to-be-impossible domain. Researchers are exploring various algorithms while altering the application of optimal control theory in new and unique ways, thanks to tremendous advancements in computational technology [25–29]. Events and their effects are reinforced by the actions taken in RL inspired by human and animal behavior [30]. At its core, RL [31] has an agent that acquires experience through trial and error as a result of its interactions with a specific environment, thus enhancing its learning curve. The agent is completely unaware of the underlying system and its ability to be controlled [32]. However, it recognizes the concept of a reward signal (as shown in Figure 1 on which the next decision is made). During the training phase, the agent learns about the best actions to take based on the reward function. The trained agent selects actions that result in the biggest rewards in order to attain optimal task performance.

As the system dynamics change or the environment transforms, the reward signal optimizes as well, and the agent alters its action policy to get bigger rewards. RL has baggage connected to the safety of its activities during the exploration phase of its learning, despite the aforementioned facts, indicating that it is a powerful tool to be used in control problems [33–36]. Control system design based on intelligent techniques is deemed most appropriate to cope with the rising complexity of system dynamics and management of

complicated controls for enhancing flexibility with the changing environment [37].

In recent studies, deep RL has been applied employing deep deterministic gradient policy (DDGP), trust region policy optimization (TRPO), and proximal policy optimization (PPO) algorithms for conventional quadcopters only, primarily focusing on controlling some specific phases of flight-like attitude control [38,39] or compensating disturbances, with PPO outperforming others [40]. Further similar studies have been discussed in relevant studies section. However, the goal of the current study is different from these mentioned researches as it aims to provide an RL-based control system for an experimental UAV which has an unusual design and is under-actuated with respect to controls, making its control challenging in the continuous state and action domains.

### 1.1 Relevant Studies

Xiang et al. [41] presented the learning algorithm which is capable of self-learning. The technique is being studied and developed in particular for cases where the reference trajectory is either overly aggressive or incompatible with system dynamics. A numerical analysis is undertaken to confirm the suggested learning algorithm's effectiveness and efficiency, as well as to exhibit improved tracking and learning performance.

Zhang et al. [42] introduced geometric reinforcement learning (GRL), for path planning of UAVs. The authors presented that GRL can make the following contributions: a) For path planning of many UAVs, GRL uses a special reward matrix, which is simple and efficient. The candidate points are chosen from a region along the geometric path connecting the current and target sites. b) The convergence of computing the reward matrix has been theoretically demonstrated, and the path may be estimated in terms of path length and risk measure. c) In GRL, the reward matrix is adaptively updated depending on information shared by other UAVs about geometric distance and risk. Extensive testing has confirmed the usefulness and feasibility of GRL for UAV navigation.

Jingzhi Hu et al. [43] integrated UAV with Internet of things. They presented a distributed sense-and-send mechanism for UAV sensing and transmission coordination. Then, in the cellular Internet of UAVs, an integration of reinforcement learning added to handle crucial challenges like trajectory control and resource management.

For conventional UAVs, onboard flight control system (FCS) based upon linear control strategies with well-designed closed-loop feedback linear controls has yielded satisfactorily results [9,44–47]. Posawat designed cascaded PID controllers [44] with automatic gain scheduling and controller adaption for various operating conditions. However, the control architecture was incapable of adapting to environmental disturbances and was highly dependent on sensor

accuracy. Oualid [46] utilized two different linear control techniques for controlling UAV dynamics. Linear quadratic servo (LQ-Servo) controller based on  $L_2$  and  $L_\infty$  norms was developed. Results, however, showed limited robustness to external disturbances, particularly to wind gusts. Further, Doyle et al. [48] utilized H-1 loop shaping in connection with  $\mu$ -synthesis, while Kulcsár [49] utilized linear quadratic regulator (LQR) architecture for the control of UAV. Both schemes satisfactorily manage the 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 nonlinearity of the system.

Realizing the limitations of linear control and evolving enhanced performance requirements of UAVs, researchers gradually resorted to applying nonlinear techniques to make the controllers more adaptive and responsive to changing scenarios. Methodologies such as back-stepping sliding mode control (SMC), nonlinear dynamic inversion (NDI), and incremental nonlinear dynamic inversion (INDI) 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 the evolving environment. Escareno [50] designed nonlinear 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 [51] implemented a nonlinear control algorithm for a UAV incorporating back-stepping sliding mode technique with adaptive gain. The authors have successfully kept the chattering noise low because of the sign function pronounced in fixed gain controllers. Experimental results of UAV showed acceptable performance with regards to stabilization and tracking. However, the algorithm was computationally expensive.

Understanding of inherent limitations of linear [44–46] and nonlinear control techniques [52] along to achieve autonomy in controls for complex aerospace systems provoked researchers to look for intelligent methods [53]. Under the ambit of ML, RL-based algorithms [54] have emerged as an effective technique for the design of autonomous intelligent control [55,56]. Coupled with neural nets, RL-based algorithms have emerged as a robust methodology in solving complex domain control problems, which significantly overpowers 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. Due to its highly adaptive characteristics, RL has increasingly found use in aerospace control applications for platforms like aircraft, missile trajectory control, fixed wing UAVs, etc.

Kim et al. [57], in their work for flat spin recovery for UAV, utilized RL-based intelligent controllers. Aircraft

nonlinearities 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 [58], in similar work, has highlighted the capability of the RL framework in picking the best solution strategy based on its offline learning, which is especially useful in controlling UAV in harsh environments and during flight-critical phases. Wickenheiser [59] exploited vehicle morphing for optimizing the perching maneuvers to achieve desired objectives. In another study, Novati [60] 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. Krozen in his research [61] has implemented reinforcement learning as an adaptive nonlinear control.

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 limited to complex yet straightforward tasks of balancing inverted pendulums, legged and bipedal robots [62], various board and computer games by effectively implementing a novel mix approach of both supervised and deep RL [63,64]. 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. It has been used only for handling critical flight phases [57] where linear control theory is difficult to implement and for navigation of UAVs [65,66]. Moreover, the in-depth analysis of the results shows slightly better performance by eliminating overshoots besides tracking a reference heading compared to a well-tuned PID controller. However, it still lacked the required accuracy as was anticipated. Further, Rodriguez-Ramos et al. [67] successfully employed deep RL for autonomous landing on a moving platform again, just focusing on the landing phase. Considering the immense potential of RL algorithms and their limited application in entirety for UAV flight control systems development, it is deemed to be mandatory to explore this dimension.

## 1.2 Research Contributions

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

A novel RL-based algorithm named as MRL has been devised. The algorithm has been specifically modified to achieve the desired objective of range enhancement while keeping the computational time required for learning the agent minimal, making it suitable for the practical onboard application. 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 offline based on reward function formulated after each iteration step. Control algorithm in line with the finalized reward function autonomously ascertains the optimum sequence of the available deflections of control surfaces at each time step ( $0.2 \text{ sec}$ ) to maximize UAV range.

Vehicle's six-degree-of-freedom (DoF) model is developed, registering its translational and rotational dynamics. The results from two developed algorithms are compared and analyzed. Simulation results show that apart from improved circular error probable (CEP) of reaching the designated location, the 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 sequential decision-making in uncertain environments.

## 2 Problem Setup

### 2.1 UAV Geometric and Mass parameters

Geometrical parameters of an experimental UAV (refer Figure 2) utilized in this research are selected to meet the mission requirements. The UAV has a mass of **596.7 kg**, wing area of  **$0.865 \text{ m}^2$**  with mean aerodynamic chord  **$0.2677 \text{ m}$** , and a wing span of  **$1.25 \text{ m}$** . The UAV has a wing–tail configuration with unconventional controls which consist of two all-moving inverted V tails to function as ruddervators. These control surfaces can move symmetrically to control pitch motion and differentially for coupled roll and yaw movements. An additional ventral fin is also placed at the bottom side for enhancing lateral stability

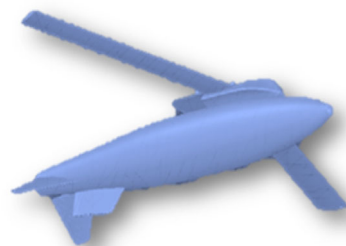


Fig. 2 UAV modal

### 2.2 UAV Mathematical Modeling

In current research, the flight dynamics modeling is carried out utilizing 6-DOF [9] model, which is typically utilized to model the vehicle motion in 3D space [9]. Assuming flat non-rotating Earth, equations are defined as follows:

$$\begin{aligned} \dot{U} &= RV - QW - g \sin \theta + \frac{X_A}{m} \\ \dot{V} &= -RU + PW + g \sin \phi \cos \theta + \frac{Y_A}{m} \\ \dot{W} &= QU - PV + g \cos \phi \cos \theta + \frac{Z_A}{m} \end{aligned} \tag{1}$$

$$\begin{aligned} \Gamma \dot{P} &= J_{XZ}(J_X - J_Y + J_Z)PQ - [J_Z(J_Z - J_Y) \\ &\quad + J_{XZ}^2]QR + J_{XZ}l + J_{XZ}n \\ \Gamma \dot{Q} &= (J_Z - J_X)PR - J_{XZ}(P^2 - R^2) + m \\ \Gamma \dot{R} &= [J_X(J_X - J_Y) + J_{XZ}^2]PQ \\ &\quad - J_{XZ}(J_X - J_Y + J_Z)QR + J_{XZ}l + J_{XZ}n \end{aligned} \tag{2}$$

$$\begin{aligned} \dot{\phi} &= P + \tan \theta (Q \sin \phi + R \cos \phi) \\ \dot{\theta} &= Q \cos \phi - R \sin \phi \\ \dot{\psi} &= \frac{Q \sin \phi + R \cos \phi}{\cos \theta} \end{aligned} \tag{3}$$

$$\begin{aligned} \dot{P}_E &= U \cos \theta \cos \psi + V(-\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) \\ &\quad + W(\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \\ \dot{P}_N &= U \cos \theta \sin \psi + V(\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) \\ &\quad + W(-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \\ \dot{h} &= U \sin \theta - V \sin \phi \cos \theta - W \cos \phi \cos \theta \end{aligned} \tag{4}$$

In the above equations, it is noteworthy that the thrust terms have been removed from the force equations (1) as the UAV has no onboard thrust generating mechanism.  $P, Q, R$  and  $U, V, W$  represent angular velocity and linear components along body x-, y- and z-axes, respectively. Euler angles are defined as  $\phi, \theta$  and  $\psi$  representing orientation of UAV with respect to the inertial frame. Position coordinates along the inertial north and east directions are defined as  $P_n$  and  $P_e$ , whereas vehicle altitude is described by  $h$ .  $X_A, Y_A, Z_A$  are the body axis forces, and moments are represented by  $l, m, n$ . Moment of inertia matrix is given by  $J$ , and  $J_x, J_y, J_z$  are the moments of inertia about the x-, y-, and z-axes, respectively.  $J_{xy}, J_{yz}$ , and  $J_{zx}$  are the cross-products of inertia.

The problem was formulated as a nonlinear system defined as Eq. (5):

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) \tag{5}$$

In the above equation,  $\vec{x} \in \mathbb{R}^{12}$  represents the state vector, control vector is  $\vec{u} \in \mathbb{R}^2$ , and fresh state estimates are represented as  $\vec{\bar{x}} \in \mathbb{R}^{12}$ . The state vector in body and wind axis is defined by Eq. (6) and Eq. (7), respectively.

$$\vec{x} = [U, V, W, \phi, \theta, \psi, P, Q, R, h, P_N, P_E]^T, \vec{x} \in \mathbb{R}^{12} \tag{6}$$

$$\vec{\bar{x}} = [V_T, \alpha, \beta, \phi, \theta, \psi, P, Q, R, h, P_N, P_E]^T, \vec{\bar{x}} \in \mathbb{R}^{12} \tag{7}$$

Control vector with continuous action space is defined in Eq. (8):

$$\vec{u} = [LF, RF]^T, \vec{u} \in \mathbb{R}^2 \tag{8}$$

Fresh state estimates are evaluated at each time step utilizing Eqs. (1-3).

Aerodynamic forces and moments acting on the aerial vehicle during different stages of the flight are governed by Eq. (9) and Eq. (10), respectively.

$$L = q_\infty SC_L, D = q_\infty SC_D, Y = q_\infty SC_Y \tag{9}$$

$$l_w = q_\infty bSC_l, m_w = q_\infty cSC_m, n_w = q_\infty bSC_n \tag{10}$$

where  $L, D, Y$  and  $l_w, m_w, n_w$  represent aerodynamic forces (lift, drag, and side force) and moments (roll, pitch, and yaw) being used in the equations of motions, whereas  $C_L, C_D, C_Y$  and  $C_l, C_m, C_n$  are the dimensionless aerodynamic coefficients in wind axis for calculating forces and moments.  $q_\infty$  is the dynamic pressure, whereas  $S$  is the wing area.

### 2.3 Aerodynamic Evaluation

The aerodynamic body force and moment coefficients in Eq. (9) and Eq. (10) vary with the flight conditions and control settings. A high-fidelity aerodynamic model is necessary to determine these aerodynamic coefficients accurately. Current research utilizes both non-empirical (such as CFD [68] and USAF Datcom [69]) and empirical [70]) techniques to determine these coefficients. The generic high-fidelity coefficient model employed for aerodynamic parameter estimation is elaborated in Eq. (11):

$$C_i = C_{i,static} + C_{i,dynamic} \tag{11}$$

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

The non-dimensional coefficients are usually obtained through linear interpolations using data obtained from various sources. Evaluation of static (basic) coefficient data (see Eq. (12)) is achieved utilizing computational fluid dynamics (CFD) [68,71] technique and are conventionally a function of control ( $\delta_{control}$ ), angle of attack ( $\alpha$ ), side slip ( $\beta$ ), and Mach number ( $M$ ).

$$\begin{aligned}
C_{i,static}(\alpha, \beta, \delta_{control}, M) &\Rightarrow C_{D_b}(\alpha, \beta, \delta_{control}, M), \\
&C_{L_b}(\alpha, \beta, \delta_{control}, M), \\
&C_{Y_b}(\alpha, \beta, \delta_{control}, M), \\
&C_{l_b}(\alpha, \beta, \delta_{control}, M), \\
&C_{m_b}(\alpha, \beta, \delta_{control}, M), \\
&C_{n_b}(\alpha, \beta, \delta_{control}, M)
\end{aligned} \quad (12)$$

where  $C_{D_b}$ ,  $C_{L_b}$ ,  $C_{Y_b}$ ,  $C_{l_b}$ ,  $C_{m_b}$ , and  $C_{n_b}$  represent the basic components of the aerodynamic forces and moments as a function of ( $\delta_{control}$ ), angle of attack ( $\alpha$ ), side slip ( $\beta$ ), and Mach number ( $M$ ).

Similarly, dynamic component (Eq. (13)) consists of rate and acceleration derivatives which are evaluated again utilizing empirical [70] and non-empirical ('USAF Stability and Control DATCOM' [69]) techniques.

$$\begin{aligned}
C_{i,dynamic}(\dot{\alpha}, \dot{\beta}, p, q, r) &= \text{Rate derivatives} \\
&+ \text{Acceleration derivatives}
\end{aligned} \quad (13)$$

Rate derivatives are the derivatives due to roll ( $p$ ) rate, 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 Eq. (14) and Eq. (15), respectively.

$$\begin{aligned}
&\text{Rate derivatives} \\
&= (C_{L_q}, C_{D_q}, C_{m_q}) \\
&\quad + (C_{Y_p}, C_{l_p}, C_{n_p}) + (C_{Y_r}, C_{l_r}, C_{n_r})
\end{aligned} \quad (14)$$

$$\begin{aligned}
&\text{Acceleration derivatives} \\
&= (C_{L_{\dot{\alpha}}} + C_{D_{\dot{\alpha}}} + C_{m_{\dot{\alpha}}}) \\
&\quad + (C_{Y_{\dot{\beta}}} + C_{l_{\dot{\beta}}} + C_{m_{\dot{\beta}}}).
\end{aligned} \quad (15)$$

## 3 MRL Framework

### 3.1 Introduction

Basic reinforcement learning algorithms are aimed at finding an optimal state-value function  $\mathcal{V}\pi^*$  or an action-value function  $\mathcal{Q}\pi^*$ , while following a policy  $\pi$  which is a time-dependent distribution over actions given states (16) and guides the choice of action at any given state.

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s] \quad (16)$$

State-value function is the expected return starting from state  $s$ , while following policy  $\pi$  and gathering scalar rewards once transitioning between the states (17). The agent's behavior is carefully controlled during the exploration phase so that maximum states are visited at least once during the course of learning. However, the action-value function is

determined by the return that is accumulated by the agent being in any particular state  $s$  and taking action  $a$  (18).

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] \quad (17)$$

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \quad (18)$$

Total reward of each episode  $\mathcal{R}_s^a$  is defined as expectation of rewards at each step of the episode given state and action and is shown (19)

$$\mathcal{R}_s^a = \mathbb{E}_\pi[R_{t+1} | S_t = s, A_t = a] \quad (19)$$

### 3.2 RL Algorithm Selection Challenge

The development of an appropriate RL algorithm corresponding to any problem is challenging as its implementation varies from the nature of problem in hand [72,73]. Factors such as state ( $s$ ) and action space ( $a$ ) domain type (discrete or continuous), direct policy search ( $\pi$ ) or value function ( $v$ ), model-free or model-based, and requirement for incorporation of neural nets (deep RL) are dictating parameters in formulation/selection of an appropriate algorithm.

Current research work problem is a complex nonlinear 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 in hand due to continuous state and action space [74], a unique approach of MRL is employed which adapts to the desired requirements optimally.

### 3.3 Model-Free Reinforcement Learning (MRL) and RL Dynamic Programming (DP) Architecture

#### 3.3.1 RL Dynamic Programming (DP)

RL DP algorithm employs *Bellman's principle of optimality* [75] at its core. The optimality principle basically works by breaking a bigger complex problem into smaller subproblems and then solving each in a recursive manner, i.e., it optimizes subproblems and combines them to form an optimal solution [76–79]. The RL DP algorithm requires that the environment is a Markov decision process (MDP) whereby the environment model is known along with the state transition matrix. It performs full widths backups at each step (refer Figure 3), where every possible successor state and action is considered at least once. It computes the value of a state based on all possible actions  $a$ , resulting in all possible successor states  $s'$  and all possible rewards. The RL DP algorithm evaluates values and action-value functions using Eq. (20).

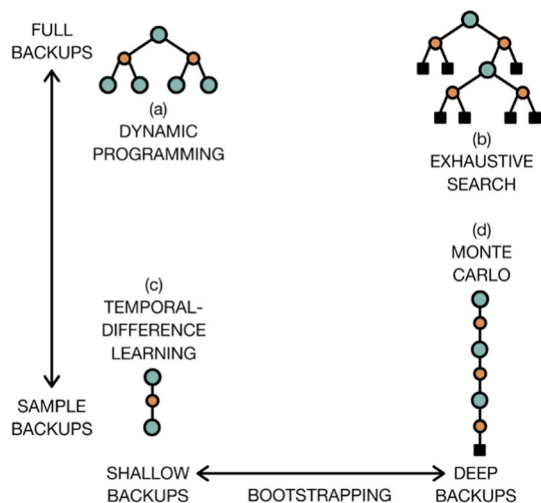


Fig. 3 RL DP implementation

$$v_{\pi}(s) = \sum_{a \in A} \left( \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_{\pi}(s') \right), \tag{20}$$

$$q_{\pi}(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_{\pi}(s', a')$$

The policy (set of good actions) which gives maximum reward as per the defined reward function is known as an *Optimal Policy*  $\pi_*$  and is defined in Eq. (21):

$$v_*(s) = \max_{\pi} v_{\pi}(s), \tag{21}$$

$$q_*(s, a) = \max_{\pi} q_{\pi}(s, a)$$

RL DP once configured optimally is ideally suited in situations where the state of the system is changing continuously over time and sequential decisions are required [80]. It sequentially improves the policy because every action being selected at each step maximizes the overall return.

### 3.3.2 MRL Framework

Devised new MRL algorithm in this research is a derivative of RL DP algorithm. However, in MRL a priori knowledge of the model parameters is not required. This effectively makes the proposed algorithm **model-free**. Further, the process of policy optimization is managed through the iterative development of an **optimal reward function** instead of a value function or action-value function only. This ensures that from the ab initio, optimal action is chosen at each time step [81,82].

After the development of an optimal reward function, the proposed MRL algorithm, which is the improved and

a model-free variant of RL DP, takes all the available actions into account one by one, while calculating reward for actions taken at every step of the algorithm. Then, among all the rewards accumulated for each action taken in a particular state, it characterizes the action with maximum reward as the optimal action as shown in Eq. (22). MRL algorithm is elaborated at Algorithm 1.

$$V(S_t) \leftarrow \max [R_{t+1} + \gamma V(S_{t+1})] \tag{22}$$

### Algorithm 1 MRL Policy Iteration Algorithm

- 1: Initialize states  $s$
- 2: Embed iterative reward function
- 3: Initialize action policy  $\pi$
- 4: Evaluate reward ( $\mathcal{R}$ ) for entire action space: (a)  $\pi(s) = \arg \max_{a \in A} q(s, a) = \arg \max_{a \in A} (\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_{\pi}(s'))$  (b) Using synchronous updates, update each optimal state and action pair
- 5: **repeat**
- 6: Step 4
- 7:
- 8: Terminal state is reached

This process of identifying optimal action at each step continues to ensure optimizing the entire trajectory starting from the initial launch conditions to the terminal stage.

Configuring the trajectory optimization problem in the MRL environment was challenging as it was difficult to accurately formulate the reward function, which fulfills the desired objectives optimally. Erroneously developed reward functions drive the agent to achieve non-priority goals and non-converging solution. Another manifested problem was the fact that the optimization process is inherently iterative. Arriving at the desired final reward function takes considerable time, which must be minimized. Lastly, the application of MRL for a complex problem based upon the continuous domain requires accurate discretization of the constituent domains. These need to be curtailed to ensure that the algorithm remains computationally viable.

### 3.3.3 MRL Controller Development Architecture

To make the above-stated MRL controller algorithm efficient, the associated action space was analyzed. With two actions  $u \in \mathbf{R}^2$  (i.e.,  $LF$  and  $RF$ ), the search space was segregated corresponding to deflection range of  $\pm 10^\circ$ . The action space of each control was then discretized into **50** equal spaces, making a total of **2500** actions. This was primarily done to make the algorithm computationally acceptable. Then, scalar reward function was formulated for maximizing the glide range of the experimental glide UAV. An inherent penalization was introduced in the reward function. This ensured that if the platform sets of course from the desired state values

**Table 1** Initial launch conditions

No.	Launch parameter	Value
1	Altitude	30000ft
2	Mach No.	0.7
3	Angle of attack ( $\alpha$ )	$0^\circ$ & $3^\circ$

during the learning phase, the reward will decrease as the penalty is deducted from the reward function.

Starting from initial conditions (launch conditions), the entire discretized action space was swept. A scalar reward for each action pair was calculated based on the finalized reward function. Action pair, which resulted in the best compensation for a specific given set of states, is chosen as optimal value of conditions and activities.

Subsequently, at the next step, for the chosen set of states, the same sequence of **2500** actions is applied, and again an optimal action pair based on the highest reward is selected and stored along with the new set of states. This optimization process at every step of the process continues until the terminal state (when the experimental glide vehicle hits the ground with the employed condition of  $z$  is less than or equal to zero in the algorithm) is reached. It is noteworthy that the optimal action corresponding to maximum reward was being taken at every step, so the entire trajectory was optimal. The results are discussed in Sect. 4.

## 4 Results and Discussion

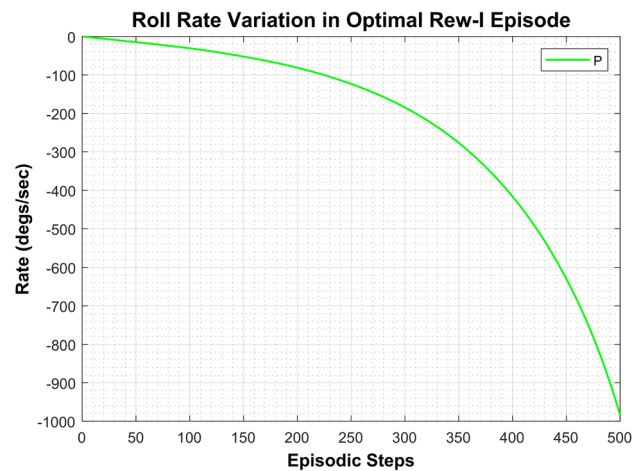
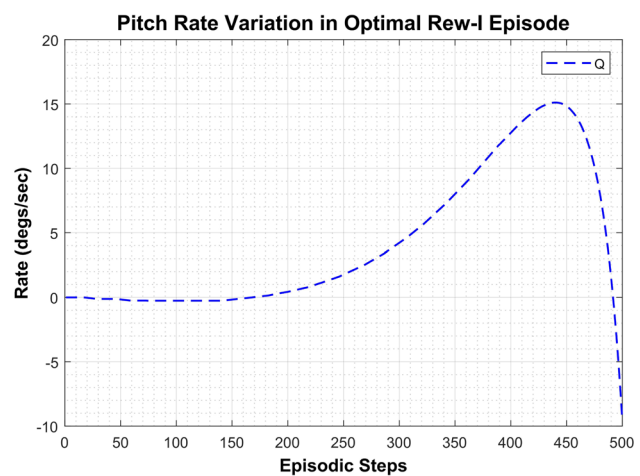
Results obtained from the suggested MRL algorithm which is a variant of RL DP algorithm are discussed here. Variation of all the 12 states during the glide phase of an experimental UAV as mentioned in Eq. (7) has been plotted against the episodic steps. The simulation time step after test and trial is kept as 0.1 secs as it adequately captures the quantum of change of states yielding optimum results for the entire state space. The initial launch conditions for the gliding vehicle are specified in Table 1

### 4.1 MRL Controller Results

The initial reward function formulated for the MRL controller is depicted in Equation (23).

$$\begin{aligned}
 pny &= |P| + |Q| + |R| \\
 r &= xpos \\
 Rew &= r - pny
 \end{aligned} \quad (23)$$

where  $pny$  represents the penalty defined at each step of the simulation;  $r$  is a scalar value based on increasing  $xpos$

**Fig. 4** Roll rate variation reward function I**Fig. 5** Pitch rate variation reward function I

which is the incremental current  $x$  value or the gliding distance covered. At first, only three states corresponding to body rates were included in the cost function. Simulation carried out utilizing this initially formulated reward function showed body rates exploding just after **300** episodic steps (refer Figures 4, 5, 6) while only achieving approximately **19 kms** of range, as shown in Figure 7. It is noteworthy that roll and yaw rates are excessively high, thus showing *platforms instability in the roll and yaw dynamics along with their inherent strong coupling due to unconventional design of the UAV.*

Analysis of the previous results necessitated for more stringent control of the body rates. Therefore, next iteration focused on adding variable weightages to the rates in order to control them (Eq. (24) efficiently. The addition of the weights primarily aimed at keeping the penalty low. This focused effort resulted in increasing the reward also which is evident through the increase in glide range. However, once again, the rates started to grow, surpassing the anticipated



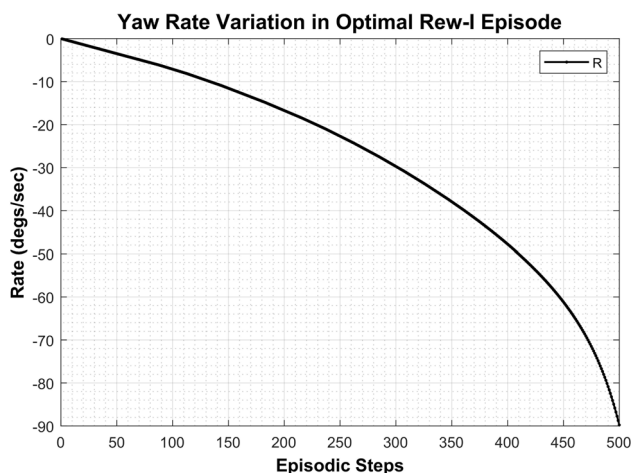


Fig. 6 Yaw rate variation reward function I



Fig. 8 Rates variation reward function III

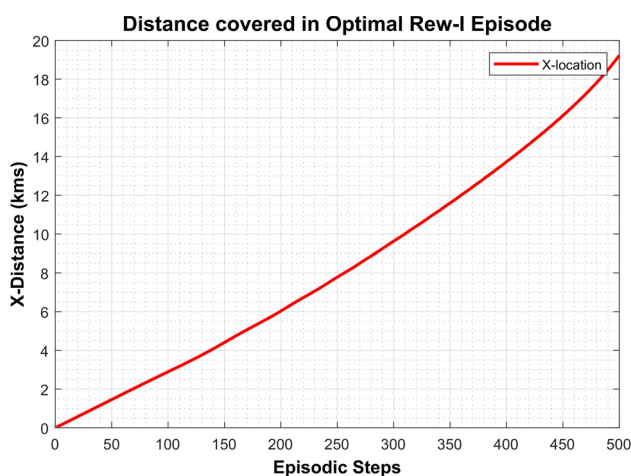


Fig. 7 Glide range of UAV reward function I

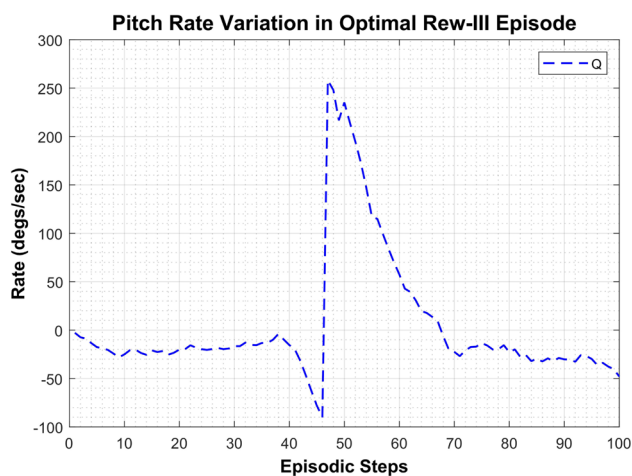


Fig. 9 Rates variation reward function III

tolerance range and causing instability.

$$\begin{aligned}
 pny &= n_1|P| + n_2|Q| + n_3|R| \\
 r &= xpos \\
 Rew &= r - pny
 \end{aligned}
 \tag{24}$$

After continuous thought process, besides the rates, quantum of change in rates was now targeted and a new reward function was formulated as mentioned in Eq. (25).

$$pny = n_1|P| + n_2|Q| + n_3|R| + \Delta P + \Delta Q + \Delta R \tag{25}$$

where  $\Delta$  in the reward function represents the state change.

Analysis of the initial results of this new structure reveals that the rates remained controlled for increased time steps and the range slightly enhanced to **22 kms** as evident in Figure 16; however, rates blew up in between shown (refer Figures 8, 9, 10)

Next, once again previous weightages were re-tuned and variable weightages were added to the change in rates of reward function. It is meaningful to highlight here that because of the excessive nonlinearity associated with the experimental vehicle based on its peculiar design, roll and yaw rates were specially focused as shown in Eq. (26).

$$\begin{aligned}
 pny &= n_1|P| + n_2|Q| + n_3|R| + n_4\Delta P \\
 &+ n_5\Delta Q + n_6\Delta R
 \end{aligned}
 \tag{26}$$

Although the rates controllability was achieved for a longer duration (refer Figures 12, 13, 14), the vehicle remained unstable (Figure 15) with range enhancement to about **30 kms** (Figure 16).

The increasing range, precision, and rates of controllability over the increased number of steps built confidence toward iterative re-tuning of the reward function.

It is critical to understand that a random increase in the weights would increase  $pny$ , thus sharply decreasing the reward for each step. Therefore, thorough analysis is required during formulation of the reward function as an ill developed reward function would result in an instability and non-convergence of the MRL algorithm. Keeping same concern in focus, next, the difference of rates with their corresponding desired absolute values was also included in the reward function as elaborated in Eq. (27).

$$pny = n_1|P| + n_2|Q| + n_3|R| + n_4\Delta P + n_5\Delta Q + n_6\Delta R + n_7\delta P + n_8\delta Q + n_9\delta R \quad (27)$$

where  $\delta$  represents the difference from the desired reference value in the penalty part of the reward function. Interim results on the basis of reward function mentioned as Eq. (27)

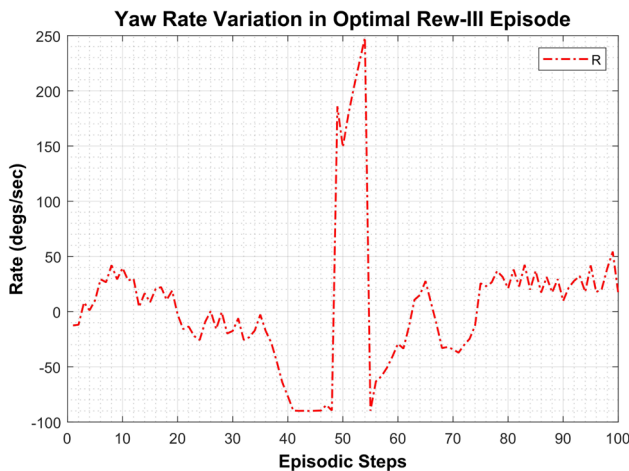


Fig. 10 Rates variation reward function III

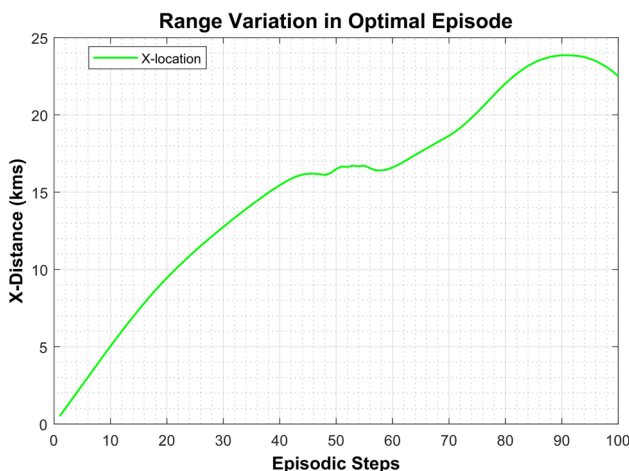


Fig. 11 Glide range of UAV reward function III

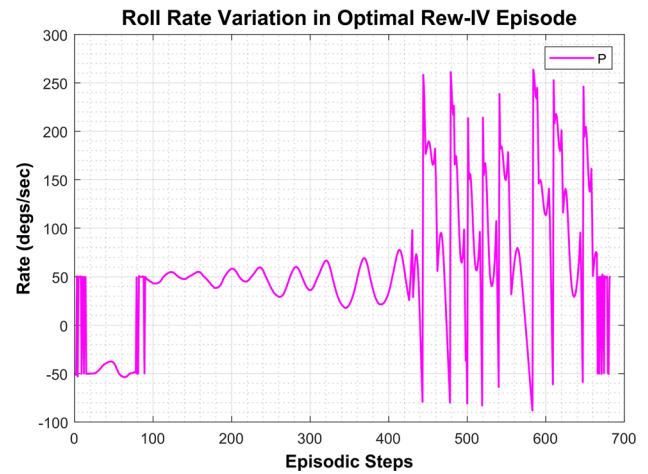


Fig. 12 Roll rate variation reward function IV

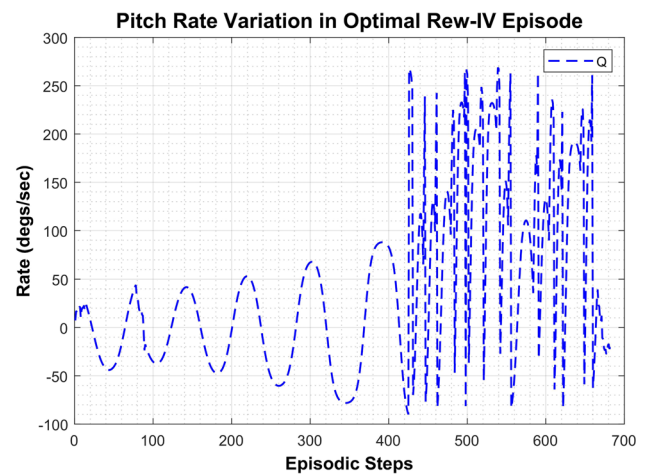


Fig. 13 Pitch rate variation reward function IV

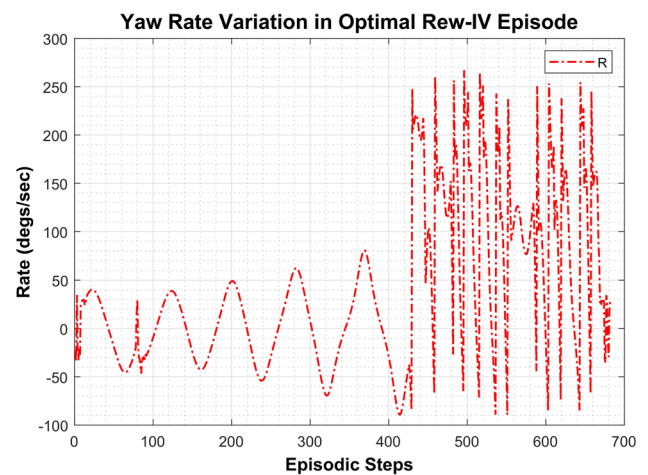


Fig. 14 Yaw rate variation reward function IV

show improvement in controlling the rates as shown in Figures 17, 18, 19. The reward started to increase with each step of the episode, as shown in Figure 20. Similarly, the range, lateral distance, and altitude showed considerably improved results as depicted in Figure 21 and Figure 22. A gliding range of around **63kms** was achieved.

The iterative process of formulating an optimal reward function continued clearly focusing on arresting the rates variation. To improve the control of states, additional dynamic weights  $n_7, n_8, n_9,$  and  $n_{10}$  were also added to the already finalized structure Eq. (27) for gaining an effective control of the changing rates with each step of the episode. Subsequently,  $y_{dis}$  parameter was also added in the penalty to restrict platforms lateral movement in the Y-direction. Additionally, the attribute of altitude decrease was also included in the  $r$ , i.e.,  $z_{pos}$ , to contribute positively with every step. The final reward function is shown as a set of Eq. (28).

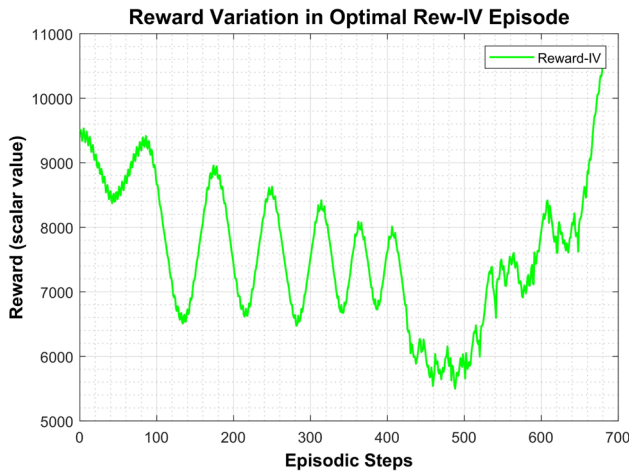


Fig. 15 Variation of reward in reward function IV

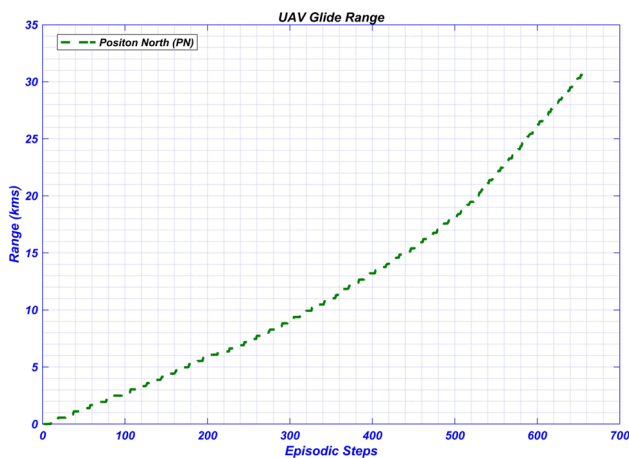


Fig. 16 Glide range of UAV reward function IV

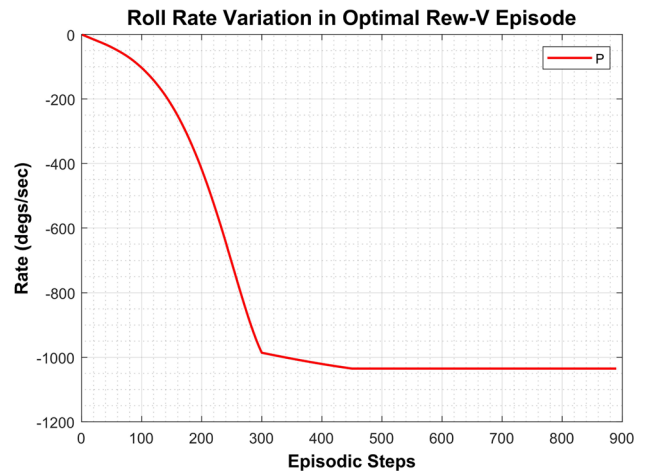


Fig. 17 Roll rate variation reward function V

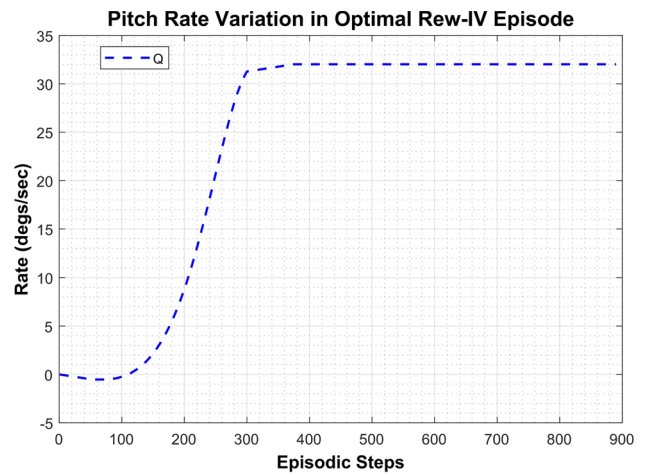


Fig. 18 Pitch rate variation reward function V

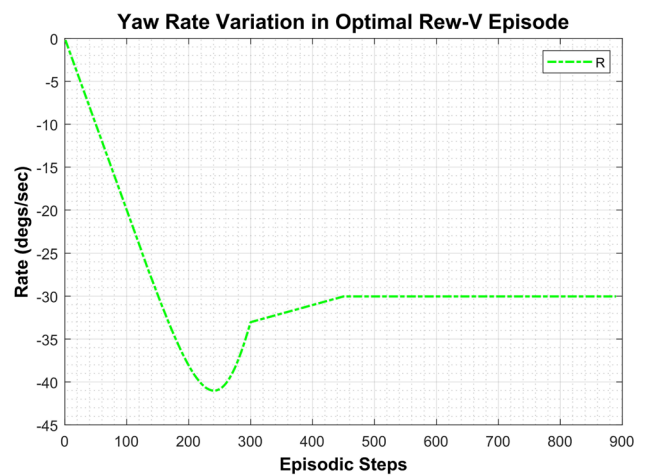


Fig. 19 Yaw rate variation reward function V



Fig. 20 Reward function V

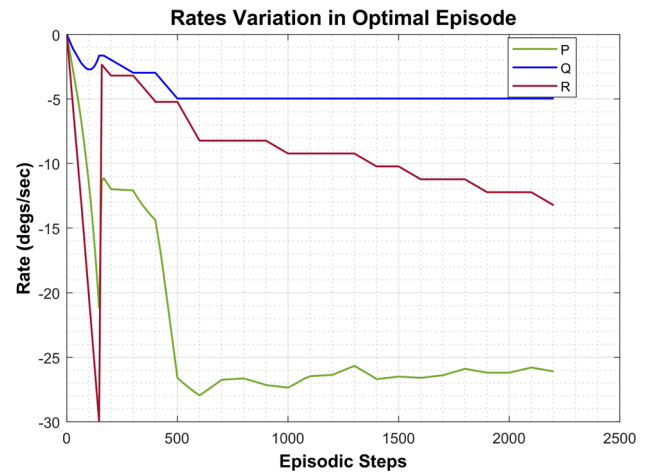


Fig. 23 UAV rates for reward function V

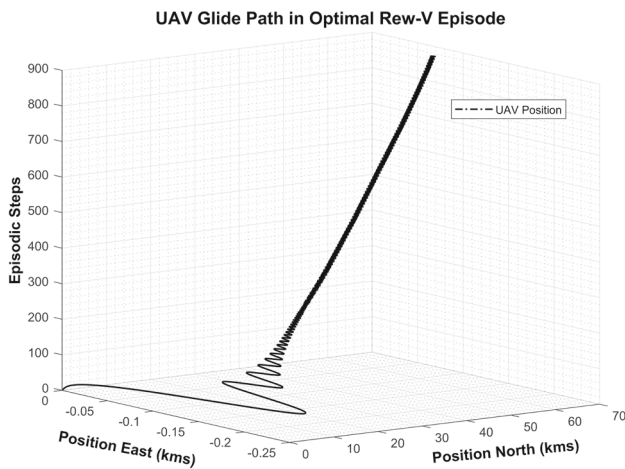


Fig. 21 UAV glide path of UAV reward function V

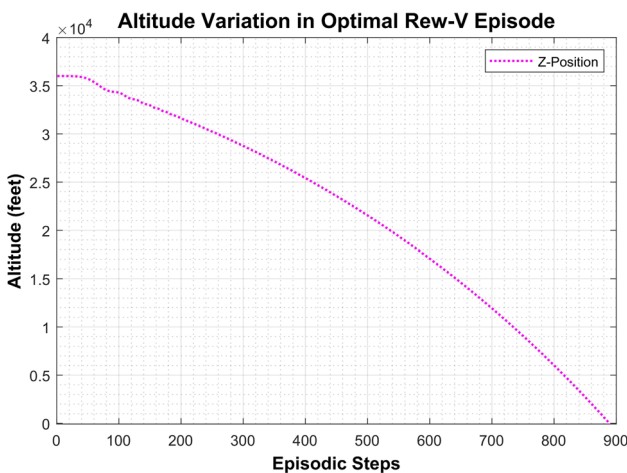


Fig. 22 Altitude profile of UAV reward function V

$$\begin{aligned}
 pny &= n_1|P| + n_2|Q| + n_3|R| + n_4\Delta P + n_5\Delta Q \\
 &\quad + n_6\Delta R + n_7\delta P + n_8\delta Q + n_9\delta R + n_{10}ydis \quad (28) \\
 r &= 10^{-3} \times xpos^2 + (36000 - zpos) \\
 rew &= r - pny
 \end{aligned}$$

After incorporation of the final reward function as a set of Eq. (28) in the control algorithm, final results corresponding to all states of *MRL-based controller*, plotted against sequential episodic time steps for the glide vehicle are presented in ensuing paragraphs. The selection of optimal control deflections by the controller during the flight regime amidst changing scenarios can be appreciated from the states' results and the gliding range achieved.

Variation of rates during the flight of UAV are depicted in Figure 23. Initial negative spike in roll and yaw rates highlights the exploration phase of the agent where it learns to select best control deflections trying to arrest the increasing roll and yaw rates. The graph also validates the vital role and yaw coupling because of the unconventional design of the UAV. After 500 episodic steps, an optimal trade-off among the rates achieves the maximum glide range.

Figures 24, 25, and 26 explain Euler angles variation during the flight. Considerable variation in roll angle (around  $\pm 3^\circ$ ) is initially experienced until the time rates settle. Later, it determines to  $\pm 0.8^\circ$  which indicates loss of negligible energy. The pitch angle variation in an episode is initially large (around 0 to -4 degs) until the time rates are conserved. Later, it is close to -2 degs but shows a slight diverging behavior at the culmination of the episode, which is not desirable but acceptable. The variation of yaw angle in an attack is initially considerable (around +/- 2 degs) until the time optimal rates trade-off is achieved. Later, it's close to +1 deg because the UAV is covering eastwards lateral distance. The initial variation (up to 500 episodic steps) in roll and pitch rate can also be connected with the roll angle variation.

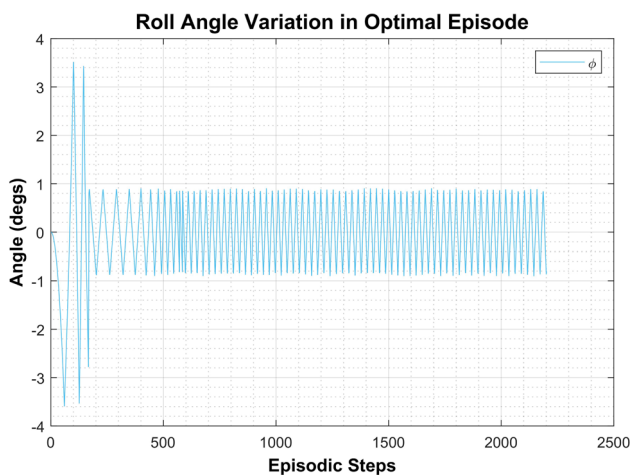


Fig. 24 UAV roll angle variation

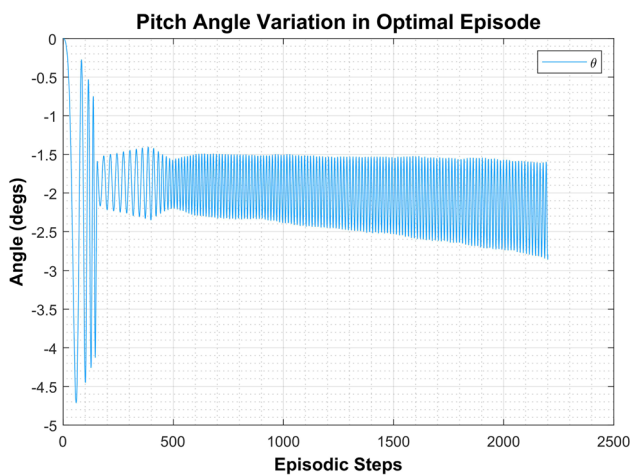


Fig. 25 UAV pitch angle variation

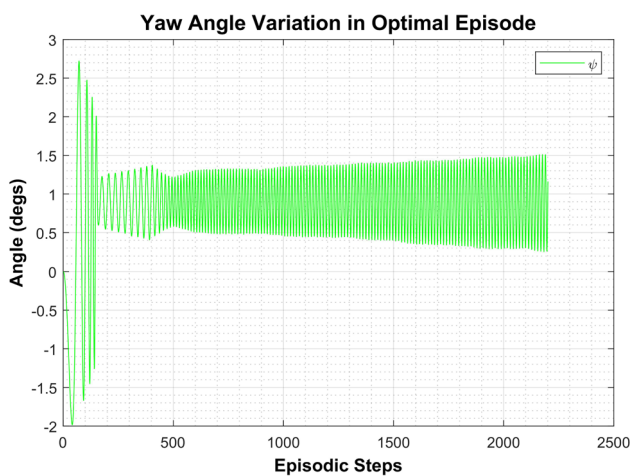


Fig. 26 UAV yaw angle variation

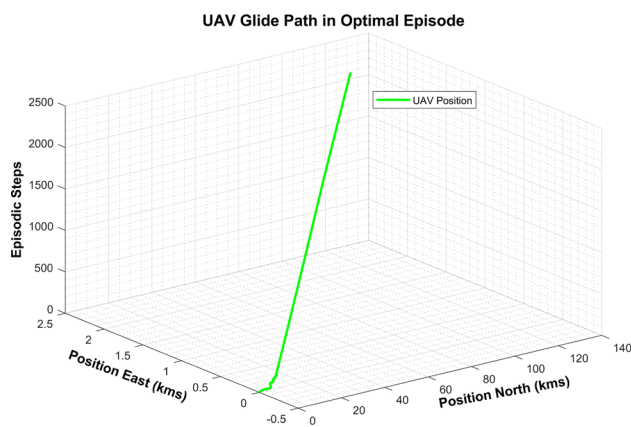


Fig. 27 Optimal glide path of UAV

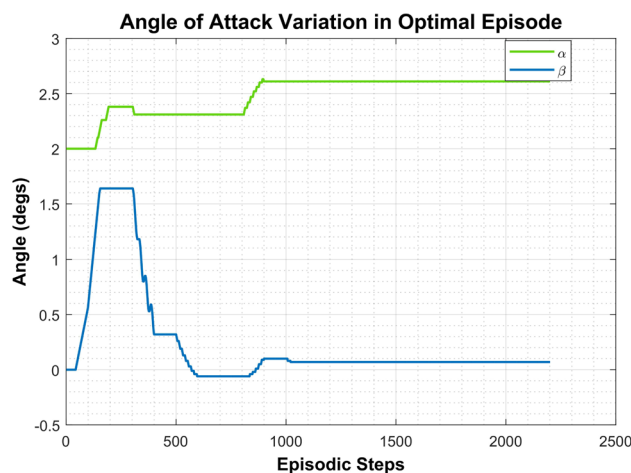


Fig. 28 Aerodynamic angles of UAV

Figure 27 shows the glide path of the UAV. Platform achieved an optimal range of more than **120 kms**. While maintaining smooth descent, UAV maintains a constant yaw angle of around 1 deg, and the total lateral distance covered in the entire gliding flight is approximately **2.4 kms**.

Figure 28 depicts the variation of aerodynamic angles during the flight. Angle of attack launched from initial  $2^\circ$  is maintained around  $2.6^\circ$ , after controlling the initial fluctuation of the body rates. Side slip angle is adjusted during the flight to achieve maximum range.

Velocity decreases smoothly as a result of drag and slight increase of alpha as shown in Figure 29.

Altitude variation is smooth along the trajectory, and the vehicle descent is controlled optimally to maximize the range as shown in Figure 30.

It is evident from the results that the autonomous MRL controller continuously arrests the rate through the reward function while keeping them within limits in pursuit of optimal performance. The reward function graph gradually grows while increasing reward, thus indicating optimal actions being taken at every step of the episode.

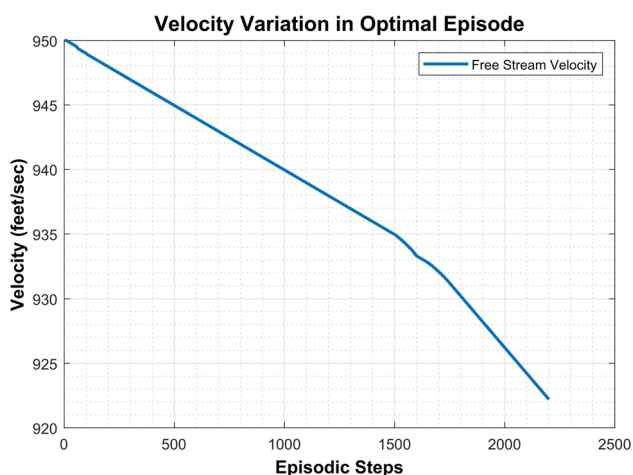


Fig. 29 Velocity profile of UAV

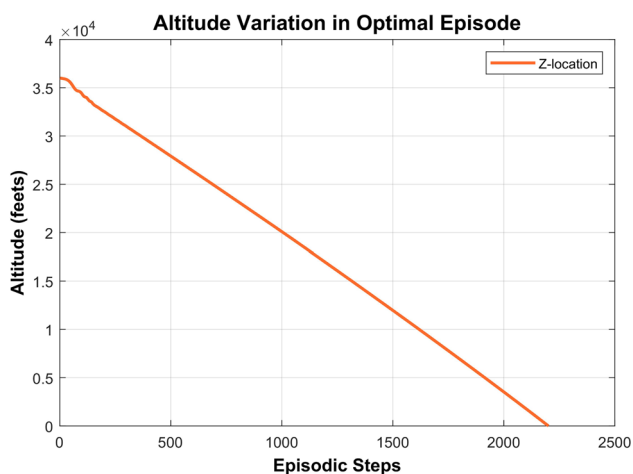


Fig. 30 Altitude variation of UAV

## 5 Conclusion

In this research, RL-based intelligent nonlinear controller for an experimental glide UAV was proposed utilizing the MRL algorithm. 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. MRL approach gave the optimal range of around **120 kms**, while handling the nonlinearity of vehicle (controlling the roll, pitch, and yaw rates in a trade-off) through effective control deflections, which were being monitored by the changing reward function. Devised RL algorithm is proved to be computationally acceptable, wherein the agent was successfully trained for large state and action space.

The performance of the controller was evaluated in a **6-DoF** simulation developed with the help of MATLAB and FlightGear software. RL-based controller outperformed the classical 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/nonlinear methods. The investigations made in this research provide a mathematical-based analysis for designing a preliminary guidance and control system for the aerial vehicles using intelligent controls. This research must open 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.

**Data Availability Statement** Data are available from the authors upon reasonable request.

## Declarations

**Conflict of interest** The authors declare that there is no conflict of interest regarding the publication of this paper.

**Ethical Approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed Consent** Informed consent was obtained from all individual participants included in the study.

## References

1. Yanushevsky, R.: Guidance of Unmanned Aerial Vehicles. CRC Press (2011)
2. Mir, I.; Eisa, S.; Taha, H.E.; Gul, F.: On the stability of dynamic soaring: Floquet-based investigation. In *AIAA SCITECH 2022 Forum*, page 0882, (2022)
3. Mir, I.; Eisa, S.; Maqsood, A.; Gul, F.: Contraction analysis of dynamic soaring. In *AIAA SCITECH 2022 Forum*, page 0881, (2022)
4. Mir, I.; Taha, H.; Eisa, S.A.; Maqsood, A.: A controllability perspective of dynamic soaring. *Nonlinear Dyn.* **94**(4), 2347–2362 (2018)
5. 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.* **79**, 17–36 (2018)
6. Mir, I.; Maqsood, A.; Akhtar, S.: Optimization of dynamic soaring maneuvers to enhance endurance of a versatile uav. In *IOP Conference Series: Materials Science and Engineering*, volume 211, page 012010. IOP Publishing, (2017)
7. Cai, G.; Dias, J.; Seneviratne, L.: A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends. *Unmanned Syst.* **2**(02), 175–199 (2014)
8. Mir, I.; Eisa, S.A.; Taha, H.E.; Maqsood, A.; Akhtar, S.; Islam, T.U.: A stability perspective of bio-inspired uavs performing dynamic soaring optimally. *Bioinspir. Biomim* (2021)
9. Mir, I.; Akhtar, S.; Eisa, S.A.; Maqsood, A.: Guidance and control of standoff air-to-surface carrier vehicle. *Aeronaut. J.* **123**(1261), 283–309 (2019)
10. Mir, I.; Maqsood, A.; Taha, H.E.; Eisa, S.A.: Soaring energetics for a nature inspired unmanned aerial vehicle. In *AIAA Scitech 2019 Forum*, page 1622, (2019)



11. Elmeseiry, N.; Alshaer, N.; Ismail, T.: A detailed survey and future directions of unmanned aerial vehicles (uavs) with potential applications. *Aerospace* **8**(12), 363 (2021)
12. Giordan, Daniele; Adams, Marc S.; Aicardi, Irene; Alicandro, Maria; Allasia, Paolo; Baldo, Marco; De Berardinis, Pierluigi; Dominici, Donatella; Godone, Danilo; Hobbs, Peter; et al.: The use of unmanned aerial vehicles (uavs) for engineering geology applications. *Bulletin of Engineering Geology and the Environment* **79**(7), 3437–3481 (2020)
13. Winkler, Stephanie; Zeadally, Sherali; Evans, Katrine: Privacy and civilian drone use: The need for further regulation. *IEEE Security & Privacy* **16**(5), 72–80 (2018)
14. Nurbani, Erlies Septiana: Environmental protection in international humanitarian law. *Unram Law Review*, 2(1), (2018)
15. Giordan, Daniele; Hayakawa, Yuichi; Nex, Francesco; Remondino, Fabio; Tarolli, Paolo: The use of remotely piloted aircraft systems (rpas) for natural hazards monitoring and management. *Natural hazards and earth system sciences* **18**(4), 1079–1096 (2018)
16. Nikolakopoulos, Konstantinos G.; Soura, Konstantina; Koukouvelas, Ioannis K.; Argyropoulos, Nikolaos G.: Uav vs classical aerial photogrammetry for archaeological studies. *Journal of Archaeological Science: Reports* **14**, 758–773 (2017)
17. Abualigah, Laith; Diabat, Ali; Sumari, Putra; Gandomi, Amir H.: Applications, deployments, and integration of internet of drones (iod): a review. *IEEE Sensors Journal*, (2021)
18. Mir, Imran; Eisa, Sameh A.; Maqsood, Adnan: Review of dynamic soaring: technical aspects, nonlinear modeling perspectives and future directions. *Nonlinear Dynamics* **94**(4), 3117–3144 (2018)
19. Mir, Imran; Maqsood, Adnan; Akhtar, Suhail: Biologically inspired dynamic soaring maneuvers for an unmanned air vehicle capable of sweep morphing. *International Journal of Aeronautical and Space Sciences* **19**(4), 1006–1016 (2018)
20. Mir, Imran; Maqsood, Adnan; Akhtar, Suhail: Dynamic modeling & stability analysis of a generic uav in glide phase. In *MATEC Web of Conferences*, volume 114, page 01007. EDP Sciences, (2017)
21. Mir, Imran; Eisa, Sameh A.; Taha, Haithem; Maqsood, Adnan; Akhtar, Suhail; Islam, Tauqeer Ul: A stability perspective of bioinspired unmanned aerial vehicles performing optimal dynamic soaring. *Bioinspiration & Biomimetics* **16**(6), 066010 (2021)
22. Gul, Faiza; Alhady, Syed Sahal Nazli.; Rahiman, Wan: A review of controller approach for autonomous guided vehicle system. *Indonesian Journal of Electrical Engineering and Computer Science* **20**(1), 552–562 (2020)
23. Gul, Faiza; Rahiman, Wan: An integrated approach for path planning for mobile robot using bi-rrt. In *IOP Conference Series: Materials Science and Engineering*, volume 697, page 012022. IOP Publishing, (2019)
24. Gul, F.; Rahiman, W.; Alhady, S.S.; Nazli: A comprehensive study for robot navigation techniques. *Cogent Eng.* **6**(1), 1632046 (2019)
25. Agushaka, Jeffrey O.; Ezugwu, Absalom E.; Abualigah, Laith: Dwarf mongoose optimization algorithm. *Computer Methods in Applied Mechanics and Engineering* **391**, 114570 (2022)
26. Abualigah, Laith; Yousri, Dalia; Elaziz, Mohamed Abd; Ewees, Ahmed A.; Al-Qaness, Mohammed AA.; Gandomi, Amir H.: Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering* **157**, 107250 (2021)
27. Abualigah, Laith; Elaziz, Mohamed Abd; Sumari, Putra; Geem, Zong Woo; Gandomi, Amir H.: Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications* **191**, 116158 (2022)
28. Abualigah, Laith; Diabat, Ali; Mirjalili, Seyedali; Elaziz, Mohamed Abd; Gandomi, Amir H.: The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering* **376**, 113609 (2021)
29. Oyelade, Olaide N.; Ezugwu, Absalom E.; Mohamed, Tehnan IA.; Abualigah, Laith: Ebola optimization search algorithm: A new nature-inspired metaheuristic algorithm with application in medical image classification problem. *IEEE Access*, (2022)
30. Thorndike, EL: Animal intelligence, darien, ct, (1911)
31. Sutton, Richard S; Barto, Andrew G: Planning and learning. In *Reinforcement Learning: An Introduction., ser. Adaptive Computation and Machine Learning*, pages 227–254. A Bradford Book, (1998)
32. Verma, Sagar: A survey on machine learning applied to dynamic physical systems. *arXiv preprint arXiv:2009.09719*, (2020).
33. Dalal, Gal; Dvijotham, Krishnamurthy; Vecerik, Matej; Hester, Todd; Paduraru, Cosmin; Tassa, Yuval: Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, (2018)
34. Garcia, Javier; Fernández, Fernando: A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* **16**(1), 1437–1480 (2015)
35. Matthew Kretchmar, R.; Young, Peter M.; Anderson, Charles W.; Hittle, Douglas C.; Anderson, Michael L.; Delnero, Christopher C.: Robust reinforcement learning control with static and dynamic stability. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* **11**(15), 1469–1500 (2001)
36. Mannucci, Tommaso; van Kampen, Erik-Jan.; de Visser, Cornelis; Chu, Qiping: Safe exploration algorithms for reinforcement learning controllers. *IEEE transactions on neural networks and learning systems* **29**(4), 1069–1081 (2017)
37. Mnih, Volodymyr; Kavukcuoglu, Koray; Silver, David; Rusu, Andrei A.; Veness, Joel; Bellemare, Marc G.; Graves, Alex; Riedmiller, Martin; Fidjeland, Andreas K.; Ostrovski, Georg; et al.: Human-level control through deep reinforcement learning. *nature* **518**(7540), 529–533 (2015)
38. Koch, Wil; Mancuso, Renato; West, Richard; Bestavros, Azer: Reinforcement learning for uav attitude control. *ACM Transactions on Cyber-Physical Systems* **3**, 04 (2018)
39. Nurten, EMER; Özbek, Necdet Sinan: Control of attitude dynamics of an unmanned aerial vehicle with reinforcement learning algorithms. *Avrupa Bilim ve Teknoloji Dergisi*, (29):351–357.
40. Pi, Chen-Huan.; Ye, Wei-Yuan.; Cheng, Stone: Robust quadrotor control through reinforcement learning with disturbance compensation. *Applied Sciences* **11**(7), 3257 (2021)
41. Xiang, Shuiying; Ren, Zhenxing; Zhang, Yahui; Song, Ziwei; Guo, Xingxing; Han, Genquan; Hao, Yue: Training a multi-layer photonic spiking neural network with modified supervised learning algorithm based on photonic stdp. *IEEE Journal of Selected Topics in Quantum Electronics* **27**(2), 1–9 (2020)
42. Zhang, Baochang; Mao, Zhili; Liu, Wanquan; Liu, Jianzhuang: Geometric reinforcement learning for path planning of uavs. *Journal of Intelligent & Robotic Systems* **77**(2), 391–409 (2015)
43. Jingzhi, Hu.; Zhang, Hongliang; Di, Boya; Li, Lianlin; Bian, Kaigui; Song, Lingyang; Li, Yonghui; Han, Zhu; Vincent Poor, H.: Reconfigurable intelligent surface based rf sensing: Design, optimization, and implementation. *IEEE Journal on Selected Areas in Communications* **38**(11), 2700–2716 (2020)
44. Poksawat, Pakorn; Wang, Liuping; Mohamed, Abdulghani: Gain scheduled attitude control of fixed-wing uav with automatic controller tuning. *IEEE Transactions on Control Systems Technology* **26**(4), 1192–1203 (2017)
45. Rinaldi, F.; Chiesa, S.; Quagliotti, Fulvia: Linear quadratic control for quadrotors uavs dynamics and formation flight. *Journal of Intelligent & Robotic Systems* **70**(1–4), 203–220 (2013)
46. Araar, Oualid; Aouf, Nabil: Full linear control of a quadrotor uav, lq vs hinf. In *2014 UKACC International Conference on Control (CONTROL)*, pages 133–138. IEEE, (2014)
47. Brière, Dominique; Traverse, Pascal: 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*, pages 616–623. IEEE, (1993)



48. Doyle, John; Lenz, Kathryn; Packard, Andy: Design examples using  $\mu$ -synthesis: Space shuttle lateral axis fcs during reentry. In *Modelling, Robustness and Sensitivity Reduction in Control Systems*, pages 127–154. Springer, (1987)
49. Kulcsar, Balazs: Lqg/ltr controller design for an aircraft model. *Periodica Polytechnica Transportation Engineering* **28**(1–2), 131–142 (2000)
50. Escareno, Juan; Salazar-Cruz, S; Lozano, R.: Embedded control of a four-rotor uav. In *2006 American Control Conference*, pages 6–pp. IEEE, (2006)
51. Derafa, L.; Ouldali, A.; Madani, T.; Benallegue, A.: Non-linear control algorithm for the four rotors uav attitude tracking problem. *The Aeronautical Journal* **115**(1165), 175–185 (2011)
52. Adams, Richard J.; Banda, Siva S.: Robust flight control design using dynamic inversion and structured singular value synthesis. *IEEE Transactions on control systems technology* **1**(2), 80–92 (1993)
53. Zhou, Y.: Online reinforcement learning control for aerospace systems. (2018).
54. Kaelbling, Leslie Pack; Littman, Michael L.; Moore, Andrew W.: Reinforcement learning: A survey. *Journal of artificial intelligence research* **4**, 237–285 (1996)
55. Zhou, Conghao; He, Hongli; Yang, Peng; Lyu, Feng; Wu, Wen; Cheng, Nan; Shen, Xuemin: Deep rl-based trajectory planning for aoi minimization in uav-assisted iot. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, (2019)
56. Bansal, Trapit; Pachocki, Jakub; Sidor, Szymon; Sutskever, Ilya; Mordatch, Igor: Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*, (2017)
57. Kim, Donghae; Gyeongtaek, Oh.; Seo, Yongjun; Kim, Youdan: Reinforcement learning-based optimal flat spin recovery for unmanned aerial vehicle. *Journal of Guidance, Control, and Dynamics* **40**(4), 1076–1084 (2017)
58. Dutoi, Brian; Richards, Nathan; Gandhi, Neha; Ward, David; Leonard, John: Hybrid robust control and reinforcement learning for optimal upset recovery. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6502, (2008)
59. Wickenheiser, Adam M.; Garcia, Ephraim: Optimization of perching maneuvers through vehicle morphing. *Journal of Guidance Control and Dynamics* **31**(4), 815–823 (2008)
60. Novati, Guido; Mahadevan, Lakshminarayanan; Koumoutsakos, Petros: Deep-reinforcement-learning for gliding and perching bodies. *arXiv preprint arXiv:1807.03671*, (2018)
61. Kroezen, Dave: Online reinforcement learning for flight control: An adaptive critic design without prior model knowledge. (2019)
62. Haamoja, T.; Zhou, A.; Ha, S.; Tan, J.; Tucker, G.; Levine, S.; Dec, L.G.: Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*.
63. Silver, David; Huang, Aja; Maddison, Chris J.; Guez, Arthur; Sifre, Laurent; Van Den Driessche, George; Schrittwieser, Julian; Antonoglou, Ioannis; Panneershelvam, Veda; Lanctot, Marc; et al.: Mastering the game of go with deep neural networks and tree search. *nature* **529**(7587), 484–489 (2016)
64. Xenou, Konstantia; Chalkiadakis, Georgios; Afantenos, Stergos: Deep reinforcement learning in strategic board game environments. In *European Conference on Multi-Agent Systems*, pages 233–248. Springer, (2018)
65. Kimathi, Stephen: Application of reinforcement learning in heading control of a fixed wing uav using x-plane platform. (2017)
66. Pham, Huy X.; La, Hung M.; Feil-Seifer, David; Nguyen, Luan V.: Autonomous uav navigation using reinforcement learning. *arXiv preprint arXiv:1801.05086*, (2018)
67. Rodriguez-Ramos, Alejandro; Sampedro, Carlos; Bavl, Hriday; De La Puente, Paloma; Pascual, Campoy: A deep reinforcement learning strategy for uav autonomous landing on a moving platform. *Journal of Intelligent & Robotic Systems* **93**(1–2), 351–366 (2019)
68. Petterson, Kristian: Cfd analysis of the low-speed aerodynamic characteristics of a ucav. *AIAA Paper* **1259**, 2006 (2006)
69. Finck, R.D.: Air Force Flight Dynamics Laboratory (US), and DE Hoak. *USAF stability and control DATCOM, Engineering Documents* (1978)
70. Roskam, J.: *Airplane design* 8vol. (1985)
71. Buning, P.G.; Gomez, R.J.; Scallion, W.I.: Cfd approaches for simulation of wing-body stage separation. *AIAA Paper* **4838**, 2004 (2004)
72. Hafner, R.; Riedmiller, M.: Reinforcement learning in feedback control. *Mach. Learn.* **84**(1–2), 137–169 (2011)
73. Laroche, R.; Feraud, R.: Reinforcement learning algorithm selection. *arXiv preprint arXiv:1701.08810*, (2017)
74. Kingma, D.P.; Adam, J.B.: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014)
75. Bellman, R.: Dynamic programming. *Science* **153**(3731), 34–37 (1966)
76. Bellman, R.E.; Dreyfus, S.E.: *Applied Dynamic Programming*. Princeton university press (2015)
77. Liu, D.; Wei, Q.; Wang, D.; Yang, X.; Li, H.: *Adaptive Dynamic Programming with Applications in Optimal Control*. Springer (2017)
78. Luo, B.; Liu, D.; Huai-Ning, W.; Wang, D.; Lewis, F.L.: Policy gradient adaptive dynamic programming for data-based optimal control. *IEEE Trans. Cybern.* **47**(10), 3341–3354 (2016)
79. Bouman, P.; Agatz, N.; Schmidt, M.: Dynamic programming approaches for the traveling salesman problem with drone. *Networks* **72**(4), 528–542 (2018)
80. Silver, D.; Lever, G.; Nicolas, H.; Daan, W., Martin, R.: *Deterministic policy gradient algorithms*, Thomas Degris (2014)
81. Matignon, L.; Laurent, G.J; Le Fort-Piat, N.: Reward function and initial values: better choices for accelerated goal-directed reinforcement learning. In *International Conference on Artificial Neural Networks*, pages 840–849. Springer, (2006)
82. Gleave, A.; Dennis, M.; Legg, S.; Russell, S.; Leike, J.: Quantifying differences in reward functions. *arXiv preprint arXiv:2006.13900*, (2020)