Computers, Materials & Continua
DOI: 10.32604/cmc.2022.024927
Article

Tech Science Press

Deep Reinforcement Learning Based Unmanned Aerial Vehicle (UAV) Control Using 3D Hand Gestures

Fawad Salam Khan[1,4], Mohd Norzali Haji Mohd[1,*], Saiful Azrin B. M. Zulkifli[2], Ghulam E Mustafa Abro[2], Suhail Kazi[3] and Dur Muhammad Soomro[1]

[1]Faculty of Electrical and Electronics (FKEE), Universiti Tun Hussein Onn Malaysia, Parit Raja, 81756, Malaysia
[2]Department of Electrical & Electronic Engineering, Universiti Teknologi PETRONAS, Seri Iskandar, 32610, Malaysia
[3]Faculty of Engineering Science and Technology, Isra University, Hyderabad, 71000, Pakistan
[4]Department of Innovation, CONVSYS (Pvt) Ltd., 44000, Islamabad, Pakistan
*Corresponding Author: Mohd Norzali Haji Mohd. Email: norzali@uthm.edu.my
Received: 04 November 2021; Accepted: 19 January 2022

Abstract: The evident change in the design of the autopilot system produced massive help for the aviation industry and it required frequent upgrades. Reinforcement learning delivers appropriate outcomes when considering a continuous environment where the controlling Unmanned Aerial Vehicle (UAV) required maximum accuracy. In this paper, we designed a hybrid framework, which is based on Reinforcement Learning and Deep Learning where the traditional electronic flight controller is replaced by using 3D hand gestures. The algorithm is designed to take the input from 3D hand gestures and integrate with the Deep Deterministic Policy Gradient (DDPG) to receive the best reward and take actions according to 3D hand gestures input. The UAV consist of a Jetson Nano embedded testbed, Global Positioning System (GPS) sensor module, and Intel depth camera. The collision avoidance system based on the polar mask segmentation technique detects the obstacles and decides the best path according to the designed reward function. The analysis of the results has been observed providing best accuracy and computational time using novel design framework when compared with traditional Proportional Integral Derivatives (PID) flight controller. There are six reward functions estimated for 2500, 5000, 7500, and 10000 episodes of training, which have been normalized between 0 to −4000. The best observation has been captured on 2500 episodes where the rewards are calculated for maximum value. The achieved training accuracy of polar mask segmentation for collision avoidance is 86.36%.

Keywords: Deep reinforcement learning; UAV; 3D hand gestures; obstacle detection; polar mask

1 Introduction

The eminent establishment of 3D hand gestures recognition systems provides massive applications with the advent of artificial intelligence. To control a UAV having its own decision to fly by taking input from 3D hand gestures provides for the user with the operator less mechanism by replacing electronic remote with 3D hand gestures. A state of art segmentation and classification method [1] describes the input image segments and classifies them with the designed optimized algorithm. This technique is utilized in designing the recognition of 3D hand gestures classification from six different types (Upward, backward, downward, upward, left, and right), which is used as the input for the designed framework.

The hardware of any UAV is the most important part due to its design to hover and control during flight. Proportional Integral Derivates (PID) and fuzzy controllers help the aviation industry to design these technologies but with certain limitation such as professional knowledge for control, electronic noise from the remote controllers, sensor-based collision avoidance, etc. The hardware design mentioned in [2] utilizes the Inertial Measurement Unit (IMU) with the sensors for yaw, pitch and roll to provide the values to formulate the reward functions for the decision to mobilize the UAV. The best reward is estimated where after each episode a reset function is defined to learn the best path during training.

There are many approaches when considering reinforcement learning such as policy-based, model-based and value-based. A model-free approach where a policy is designed for UAV controller is used for path planning, navigation and control [3]. Different path smoothening methodologies using Grey Wolf [4] are used to provide the best path planning during the flight. There are various simulators available like GymFC [5] for the implementation of reinforcement learning, especially for UAV attitude control. LSTM based hand gestures recognition system may be useful for UAV control system [6]. Generally, ResNet101 and Inception V3 are utilized as backbone network during the selection of best features for segmentation of the images [7].

Sensor-free obstacle detection using only a camera for detection and recognition has overcome the cost and maintenance of UAVs. The image instance segmentation technique [8], which is used in this work for UAV obstacle detection during its flight provides a prediction mechanism, which utilizes the contours of the object present for the UAV, a polar coordinates system where the rays are constructed from the edge of the object to calculate the Intersection Over Union (IoU) between different bounding boxes.

The study in the subject of UAV control is intriguing, not only because of several improved or proposed new DRL algorithms, but a wide range of its applications and also resolving for control issues that were previously virtually difficult to solve. The DRL algorithm's process of learning was built on knowledge collected from images in [9,10]. As a result, improvements in single and multi-UAV control utilizing various communications and sensing techniques clear the doors to widespread real-world application of these techniques in a variety of activities such as monitoring, first rescuers in disasters, transportation and agricultural, etc. Each of these studies shows that selecting a reward function is just as essential as selecting a DRL algorithm. Every study presents, a unique reward function depending on the study's scenario as well as the control algorithm's goal. This necessitates a comprehensive examination of reward functions, and testing alternative reward functions underneath the similar control algorithm can be fruitful resulting in greater control efficacy.

1.1 UAV Navigation with RL

A test is carried on a UAV utilizing various reinforcement learning algorithms with the goal of classifying the algorithms into two groupings: discrete action space and continuous action space. Reinforcement learning is predicated on the agents being educated via tests and mistakes to navigate and avoid obstacles. This feature is advantageous since the agent will begin learning on its own as quickly as the training environment is complete. The research began with RL, which was used to derive equations for sequential decision making, wherein the agent engages with the surrounding visual world by splitting it into discrete time steps. Some parameters are tuned in the state form to receive the best action provided by the Actor-network where the resultant Temporal Difference (TD) errors are normalized by Critic-network for the control of UAV.

The suggested agent in discrete action space selects to implement a strategy in the manner of greedy learning by selecting the best action depending on the provided state value. A deep Q network may be used to determine this value in high-dimensional data, such as photographs (DQN). To address these concerns, a new approach is developed where the suggested algorithm, dubbed Double Dueling (DQN), integrates the Double DQN with the Dueling DQN (D3QN). In tests, the algorithm shows a strong capacity to remove correlation and enhance the standard of the states. The study utilizes a simulation platform named AirSim, which creates images using the Unreal Engine, to assist in constructing a realistic simulation environment through using discrete action space. The simulation, while providing certain constraints in the environment, does not give intricate pathways for the UAV because all of the obstacles are situated on plain terrain. To address this problem, the researchers designed a new habitat that comprises a variety of impediments such as solid surfaces in cubes and spheres, among other things. RGB and depth sensors, as well as CNN as inputs to the RL network, are utilized to calculate the best route for the drone [3]. The system is compatible with SAC off policy and all other RL algorithms. There are various datasets containing multiple types of 3D hand gestures available but in our case study, only six types of hand gestures are required to control the UAV in six different directions.

1.2 Research Contributions

1. The design of the framework based on hybrid modules consists of 3D hand gestures recognition using deep learning and reinforcement learning to control the UAV.
2. Development of an algorithm for an embedded platform to recognize 3D hand gestures for activation of reward functions for the control of UAVs.
3. Design of the collision avoidance system for the UAV using polar mask techniques, which calculates the least distance from the center of the obstacle for collision avoidance.

1.3 Research Objective

The research objective of this study is to design a novel framework to control the UAVs with 3D hand gestures and a state of art collision avoidance system without using sensors.

1.4 Structure of the Article

The article is divided into (2) Related Work, (3) Proposed Framework (4) Results (5) Analysis and Discussion, and (6) Conclusions.

2 Related Work

Deep Reinforcement learning has changed the traditional design of flight controllers. There are two versatile adaptive controllers for unmanned aerial vehicles (UAVs). The first controller was a fuzzy logic-based robust adaptive PID. The second was based on an intelligent flight controller built on ANFIS. The results showed that the built-in controllers are robust. Similarly, the findings showed that in presence of external wind disruptions and UAV parametric uncertainties, the intelligent flight controller based on ANFIS outperformed the stable adaptive PID controller based on fuzzy inference [11]. An exhaustive study of open-source flight controllers that are freely accessible and can be used for research purposes. The drone's central feature is the flight controller, which is an integrated electronic component. Its aim is to carry out the main functions of the drone, such as autonomous control and navigation. There are many categories of flight controllers, each with its own set of characteristics and functions. The paper proposes the fundamentals of the UAV design and its elements. It investigates and contrasts the processing capacities, sensor composition, interfaces, and other features of open-source UAV platforms. It also illustrates the discontinued open-source UAV platforms [12]. Few flight controllers where timing assurances are critical in embedded and cyber-physical systems that would limit the time between sensing, encoding, and actuation. This research discusses a modular pipe model for sensor data processing and actuation. The pipe model was used to investigate two end-to-end semantics: freshness and response time. The paper provides a statistical method for calculating feasible assignment cycles and budgets that met both schedulability and end-to-end timing criteria. It shows the applicability of the design strategy by porting the CleanFlight flight controller firmware to Quest, the in-house real-time operating system. Experiments demonstrated that CleanFlight on Quest can attain end-to-end latencies even within time bounds expected by observation [13].

A new framework concept for 3D flight path tracking control of (UAVs) in windy conditions. The new design paradigm simultaneously met the following three goals: (i) 3D path tracking error device representation in wind environments using the Serret-Frenet frame, (ii) assured cost management, and (iii) simultaneous stabilization via a single controller for various 3D paths with a similar interval parameter configuration in the Serret-Frenet frame. In the Serret-Frenet frame, a path tracking error scheme based on a 3D kinematic model of UAVs in wind conditions was built to realize the three points. Inside the considered operation domains, the Takagi-Sugeno (T-S) fuzzy model accurately represented the path tracking error system. It examined a guaranteed cost controller design that reduced the upper bound of a provided output function as a benefit of the T-S fuzzy model construction. The problem of the guaranteed cost controller model was expressed in terms of Linear Matrix Inequalities (LMIs). As a result, the developed controller ensured not only path stability but also cost management and path tracking control for a suitable value 3D flight path in wind environments. Also, a simultaneous stabilization issue in terms of finding a common solution in a series of LMIs was considered. The simulation findings demonstrated the effectiveness of the proposed 3D flight path tracking control in windy conditions [14].

A monitoring flight control scheme for a quadrotor with external disturbances dependent on a disturbance observer. It was believed to include certain harmonic disturbances to aid in the processing of potential time-varying disturbances. Then, to quantify the uncertain disturbance, a disturbance observer was proposed. A quadrotor flight controller was designed using the output of the disturbance observer to monitor the provided signals produced by the reference model. Finally, a proposed control system was used to control the flight of the quadrotor Quanser Qball 2. The experimental findings were presented to illustrate the efficacy of the control technique produced [15]. The design and development of a quadrotor utilizing low-cost components and a Proportional Integral Derivative (PID) control system itself as controller. This paper also explained the PID control system similar to a flight controller. To explain the expense of developing this quadrotor, a basic economic analysis was provided. According to the results of the experimental trials, the quadrotor could fly stably with a PID controller, but there was still an overshoot at attitude responses [16]. A new full-duplex (FD) confidentiality communication system for UAVs was used, which explored its optimum configuration to maximize the UAV's Energy Efficiency (EE). Particularly, the UAV collected sensitive information from a ground channel while also sending jamming signals to disrupt a possible ground eavesdropper. This research intended to optimize the EE for their secrecy contacts by jointly optimizing the UAV trajectory as well as the source/UAV transmits/jamming forces over a finite flight time, since the UAV has minimal onboard energy in practice. Despite the fact that the formulated problem was difficult to solve optimally due to its non-convexity, the study also proposed an effective iterative algorithm to reach a good suboptimal solution. The simulation results demonstrated that major EE changes could be obtained by joint optimization, and the EE benefits were strongly dependent on the capacity of the UAV's self-interference cancelation [17].

A novel Integral Sliding Mode Control (ISMC) technique for quadrotor waypoint tracking control in the existence of model inconsistencies and external disturbances. The inner-outer loop configuration was included in the proposed controller: The outer loop generated the reference signals for the roll and pitch angles, whereas the inner loop was equipped for the quadrotor to monitor the desired x, y positions, as well as the roll and pitch angles, using the ISMC technique. The Lyapunov stability study was used to demonstrate how the detriments affected the bounded model uncertainty and external disturbances could be greatly reduced. To solve the consensus challenge, the engineered controller was applied to a heterogeneous Multi-Agent System (MAS) comprised of quadrotors and two-wheeled mobile robots (2WMRs). The control algorithms for 2WMRs and quadrotors were presented. If the switching graphs still had a spanning tree, the heterogeneous MAS would achieve consensus. Finally, laboratory experiments were carried out to validate the efficacy of the proposed control methods [18].

A collision avoidance problem involving multiple Unmanned Aerial Vehicles (UAVs) in high-speed flight, allowing UAV cooperative formation flight and mission completion. The key contribution was the development of a collision avoidance control algorithm for a multi-UAV system using a bi-directional network connection structure. To efficiently prevent collisions between UAVs as well as between the UAVs and obstacles, a consensus-based algorithm "leader-follower" control technique was used in tandem for UAV formation control to ensure formation convergence. In the horizontal plane, each UAV had the same forward velocity and heading angle, and they held a constant relative distance throughout the vertical direction. Centered on an enhanced artificial potential field method, this paper proposed a consensus-based collision avoidance algorithm for multiple UAVs. To verify the proposed control algorithm as well as provide a guide for engineering applications, simulation tests including several UAVs were conducted [19].

Because of their long-range connectivity, fast maneuverability, versatile operation, and low latency, unmanned aerial vehicle (UAV) communications play a significant role in developing the space air-ground network and achieving seamless wide-area coverage. Unlike conventional ground-only communications, control methods have a direct effect on UAV communications and may be developed collaboratively to improve data transmission efficiency. In this paper, the benefits and drawbacks of integrating communications and control in UAV systems were looked at. A new frequency-dependent 3D channel model was presented for single-UAV scenarios. Channel monitoring was then demonstrated with a flight control system, and also mechanical and electronic transmission beam formulation. New strategies were proposed for multi-UAV scenarios such as cooperative interactions, self-positioning, trajectory planning, resource distribution, and seamless coverage. Finally, connectivity protocols, confidentiality, 3D complex topology heterogeneous networks, and low-cost model for realistic UAV applications were explored [20].

A hybrid vertical takeoff and landing (VTOL) unmanned aerial vehicle (UAV) of the kind known as dual system or extra propulsion VTOL UAV in this paper [21]. This research covered the entire system construction of such VTOL UAVs, covering aircraft model and implementation, onboard computer- integration, ground station service, and long-distance communication. Aerodynamics, mechanical design, and controller creation were also explored. Finally, a hybrid VTOL UAV was tested to ensure that this had the necessary aerodynamic efficiency, flight stability, durability, and range. Furthermore, with the built-in flight controller, the VTOL UAV could fly fully autonomously in a real-world outdoor environment. It provided an excellent foundation for future research in areas such as vision-based precise landing, motion planning, and fast 3-D imaging, as well as service applications like medication delivery [22].

The design of using a motion controller to control the motion of a drone utilizing basic human movements in this research. For this implementation, the Leap Motion Controller and the Parrot AR DRONE 2.0 were used. The AR-DRONE communicated with the ground station through Wi-Fi, while the Leap communicated with the ground station via a USB connection [23]. The hand signals were recognized by the LEAP motion controller and relayed to the ground station. The ground station operated ROS (Robot Operating System) in Linux that served as the implementation's base. Python was used to communicate with the AR DRONE and express basic hand signals. In execution, Python codes were written to decode the LEAP-captured hand gestures and relay them in order to control the motion of the AR-DRONE using these gestures [24].

The gesture-sensing system leap motion to control a drone in a simulated world created by the game engine Unity. Four swiping movements and two static gestures were checked, like face up and face down. According to the findings of the experiments, static movements were more identifiable than dynamic gestures [25]. Between different users, the drone responded to gesture control with an average accuracy of more than 90% [26]. Due to their basic mechanical structure and propulsion philosophy, quadrotor UAVs are among the most common types of small unmanned aerial vehicles. Also, due to the nonlinear dynamic behavior of these vehicles, specialized stabilizing control is needed. The use of a learning algorithm that makes the training of appropriate control behavior is one potential approach in easing the tough challenge of nonlinear control design [27].

Reinforcement learning was used as a form of unsupervised learning in this case study. A nonlinear autopilot was first suggested for quadrotor UAVs based on feedback linearization. This controller then was comparable to an autopilot learned by reinforcement learning with fitted value repetition in terms of design commitment and efficiency. The effect of this comparison was highlighted by the first simulation and experimental finding [28]. They compared the performance and accuracy of the inner control loop that provides attitude control by using intelligent flight control systems trained with cutting-edge RL algorithms such as Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization, and Proximal Policy Optimization. To explore these unknown parameters, an open-source high-fidelity simulation system was created first for training a quadrotor flight controller's attitude control using RL. The environment was therefore used to equate their output to a PID controller in order to determine whether or not using RL is sufficient in high-precision, time-critical flight control [5].

3 Proposed Framework

The framework consists of a hybrid module based on deep learning and deep reinforcement learning. The deep learning module is responsible for 3D hand gestures recognition, segmentation, and classification. A private dataset contains 4200 images of 3D hand gestures of six types (up, down, back, forward, left, and right) trained deep learning module is used as output, which fed into the deep reinforcement learning module. The DRL agent (UAV) takes the state information from the environment and calculated the reward function depending upon the gestures output and sensor data from the environment. The hand gestures once segmented and classified with higher accuracy with skeletal information converted into the required signals. In Fig. 1 shows the hybrid modules where the Deep Reinforcement Learning (DRL) agent (UAV) activates the DRL algorithm after receiving the state values from (pitch, yaw, roll) and best the reward functions to identify which action to be performed.

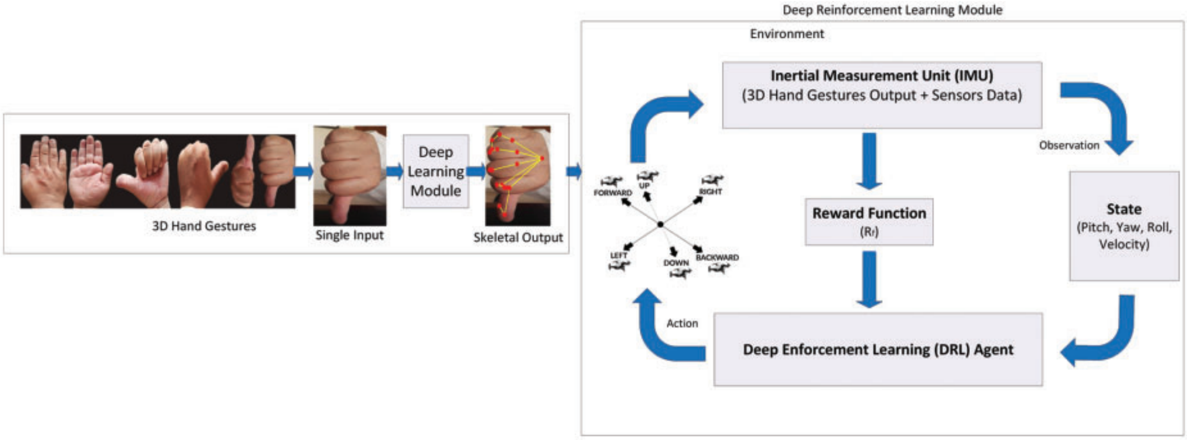
3D Hand Gestures
Single Input
Deep Learning Module
Skeletal Output
Deep Reinforcement Learning Module
Environment
Inertial Measurement Unit (IMU)
(3D Hand Gestures Output + Sensors Data)
Observation
State
(Pitch, Yaw, Roll, Velocity)
Reward Function
(Rr)
FORWARD
UP
RIGHT
LEFT
DOWN
BACKWARD
Action
Deep Enforcement Learning (DRL) Agent

Figure 1: A framework based on RL to control UAVs using 3D hand gestures

3.1 Reward Function

The framework based on deep reinforcement learning calculates the maximum reward during the flight for its decision to move from left to right or right to left, down to up or up to down, backward to forward, or forward to backward direction. The reward functions are the mathematical formation from the different values of velocity, yaw, pitch and roll. The hand gestures input which is included with these reward functions to be initialized for the UAV to take its decision according to the given hand gesture. These reward functions can mathematically describe as:

$$R_{upward} = Vy - k\sum[(x^2 + z^2) - k'\left(\theta_r^{\ 2} + \varphi_p^{\ 2} + \delta_y^{\ 2}\right) + k''] \tag{1}$$

$$R_{downward} = Vy' - k\sum[(x^2 + z^2) - k'\left(\theta_r^{\ 2} + \varphi_p^{\ 2} + \delta_y^{\ 2}\right) + k''] \tag{2}$$

$$R_{left} = Vx - k\sum[\left(y^2 + z^2\right) - k'\left(\theta_r^{\ 2} + \varphi_p^{\ 2} + \delta_y^{\ 2}\right) + k''] \tag{3}$$

$$R_{right} = Vx' - k\sum[\left(y^2 + z^2\right) - k'\left(\theta_r^{\ 2} + \varphi_p^{\ 2} + \delta_y^{\ 2}\right) + k''] \tag{4}$$

$$R_{forward} = Vz - k\sum[\left(y^2 + x^2\right) - k'\left(\theta_r^{\ 2} + \varphi_p^{\ 2} + \delta_y^{\ 2}\right) + k''] \tag{5}$$

$$R_{backward} = Vz' - k\sum[(y^2 + x^2) - k'\left(\theta_r{}^2 + \varphi_p{}^2 + \delta_y{}^2\right) + k''] \tag{6}$$

Vy describes the velocity of UAV in the Y direction, Vx demonstrates the velocity of the UAV in the X direction and Vz is the velocity of the UAV in the Z direction. x is the initial position of UAV in X-axis when forward gesture initiated, z is the initial position of UAV in Z-Axis when the downward gesture initiated and y is the initial position of UAV in Y-Axis when right hand gesture is initiated. y', x', z' for the opposite direction. θ_r are angular values for Roll, φ_p for Pitch and δ_y for Yaw. k: Constant value to minimize the motion of drone in x and z axis. k': constant value to minimize the Euler angles k": constant value to push the drone in y-direction.

The velocity (Vy, Vx, Vz) of brushless DC electric motors are adjusted near to minimize value for hovering purposes. Initially, when the UAV started, it directly hovers to 6 feet from the ground position. Fig. 2 demonstrate different positions of the UAV in the coordinate system for Pitch, Roll and Yaw.

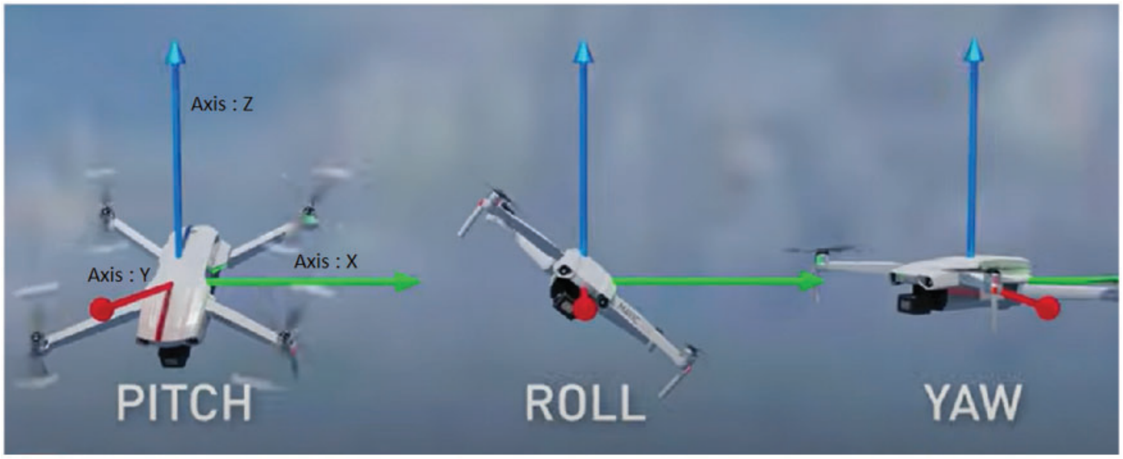

Figure 2: UAV attributes (Roll, Pitch, Yaw) with coordinate description

3.2 Algorithm for 3D Hand Gestures Recognition for UAV

The algorithm is designed for the embedded system platform to control the UAV and can be scalable for any non-embedded system.

```
Initialize Hand = hand_Detection ()
Define class Hand = classify_Hand ()
          while True
if Hand
     Gesture = class_Hand.gesture (Hand)
if Gesture == 'forward'
     callfunction $Vz - k\sum[(y^2 + x^2) - k'\left(\theta_r{}^2 + \varphi_p{}^2 + \delta_y{}^2\right) + k'']$
if Gesture == 'backward'
     callfunction $Vz' - k\sum[(y^2 + x^2) - k'\left(\theta_r{}^2 + \varphi_p{}^2 + \delta_y{}^2\right) + k'']$
```

```
if Gesture == ‘left’
    callfunction $Vx - k\sum[\left(y^2 + z^2\right) - k'\left(\theta_r^{\,2} + \varphi_p^{\,2} + \delta_y^{\,2}\right) + k'']$
if Gesture == ‘right’
    callfunction $Vx' - k\sum[\left(y^2 + z^2\right) - k'\left(\theta_r^{\,2} + \varphi_p^{\,2} + \delta_y^{\,2}\right) + k'']$
if Gesture == ‘upward’
    callfunction $Vy - k\sum[(x^2 + z^2) - k'\left(\theta_r^{\,2} + \varphi_p^{\,2} + \delta_y^{\,2}\right) + k'']$
if Gesture == ‘downward’
    callfunction $Vy' - k\sum[(x^2 + z^2) - k'\left(\theta_r^{\,2} + \varphi_p^{\,2} + \delta_y^{\,2}\right) + k'']$
else
            print (“No hand is detected”)
```

3.3 Collision Avoidance Using Polar Mask

The GPS sensor used for fencing the area which covers 10 meters from the center of the origin as shown in Fig. 3. There can be various sizes and shapes of the objects (obstacles) in its path. The UAV consists of a camera having a field of view of 3 meters. The test scenario consists of four obstacles (A, B, C, D). We considered obstacle A as the “Tree”. The Image captured for the obstacle object (tree) during the flight of the UAV fed on the backbone network where different convolution layers with stride sizes are used to create the feature pyramid network (FPN), then the process for mask-segmentation is used to rebuild the captured image into a polar-coordinate plane. For depicting an obstacle center, each bounding box with annotated area for center, mass, and the upper bound of the segmented mask is examined for efficiency.

A center-sample, if it fell within a specific level from the obstacle mass-center. A Distance-Regression of Rays is drawn over the complete mask. A network was generated for confidence scores for the center and ray length. After the mask construction, Non-Maximum Suppression (NMS) is used to eliminate superfluous masks over the same image.

The minimal bounding boxes with masks are computed and then Non-Max Suppression (NMS) relying upon on IoU of the resulting bounding boxes. The shortest distance was calculated from the origin to the boundary of the mask, once the shortest distance computed, the reward function activated and decided to move the UAV and avoid a collision from the obstacle.

A Feature-Pyramid-Network was created for the mask from the highest-scoring predictions to build by combining the best forecasts of all levels using Non-Max Suppression (NMS). The mask assembling and NMS techniques can be defined by using the center locations (w_d, v_d) as well as the length of rays $(b_1,\ b_2,\ \ldots, b_n)$, the spot of each equivalent contour point $(w_j,\ v_j)$ can be calculated.

$$w_j = \cos\theta_j \times b_j + w_d \tag{7}$$

$$v_j = \sin\theta_j \times b_j + v_d \tag{8}$$

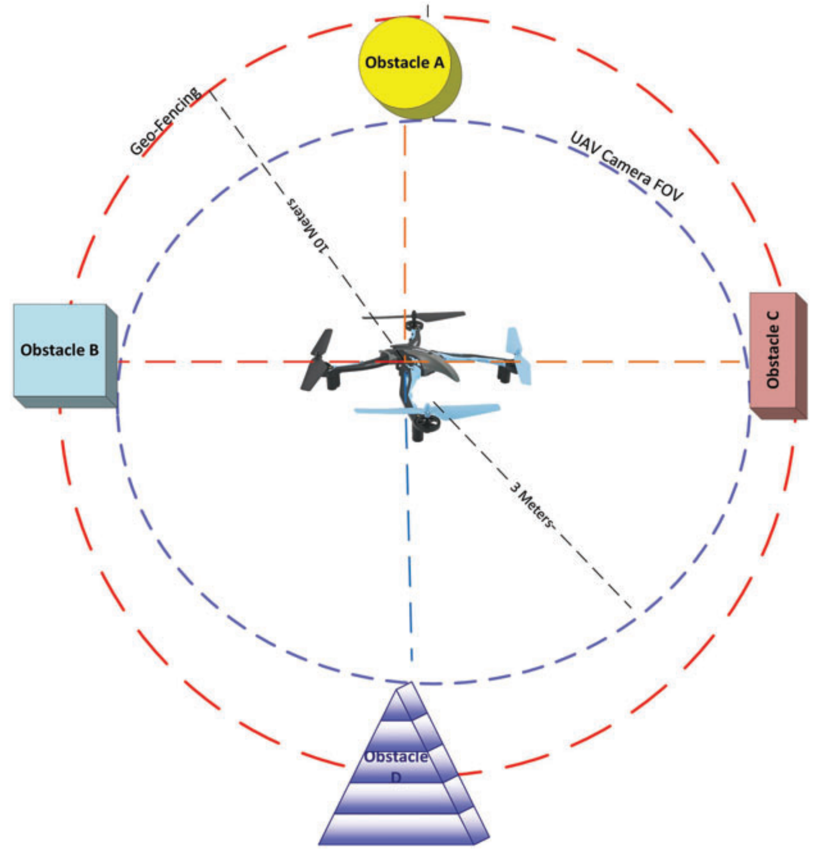

Figure 3: Obstacle avoidance scenario

For obstacle detection, centerness was developed to reduce poor bounding boxes. Nevertheless, merely implementing center-ness in a polar plane was insufficient as it was intended for conventional bounding boxes though not mask. Polar Centerness can be defined by supposing the length of rays $(b_1,\ b_2,\ \ldots, b_n)$:

$$Polar\ Centerness = \sqrt{\frac{\min\left(\{b_1,\ b_2,\ \ldots, b_n\}\right)}{max\left(\{b_1,\ b_2,\ \ldots, b_n\}\right)}} \tag{9}$$

Polar-Ray-Regression developed a convenient and straightforward approach for computing the mask-IoU in a polar plane and the Polar-IoU loss function, in order to enhance the modeling and attain competitive results. So, Polar IoU is calculated as:

$$Polar\ IoU = \frac{\sum_{j=1}^{n} b_j^{min}}{\sum_{j=1}^{n} b_j^{max}} \tag{10}$$

To maximize the size of each ray, the Polar IoU loss function is described by the Polar-IoU's Binary Cross Entropy (BCE) loss. The minus log of the Polar-IoU is used to illustrate the polar-IoU loss function. The Polar Mask architecture consist of backbone + FPN combined with the Head network is shown in Fig. 4. The polar loss function is computed as:

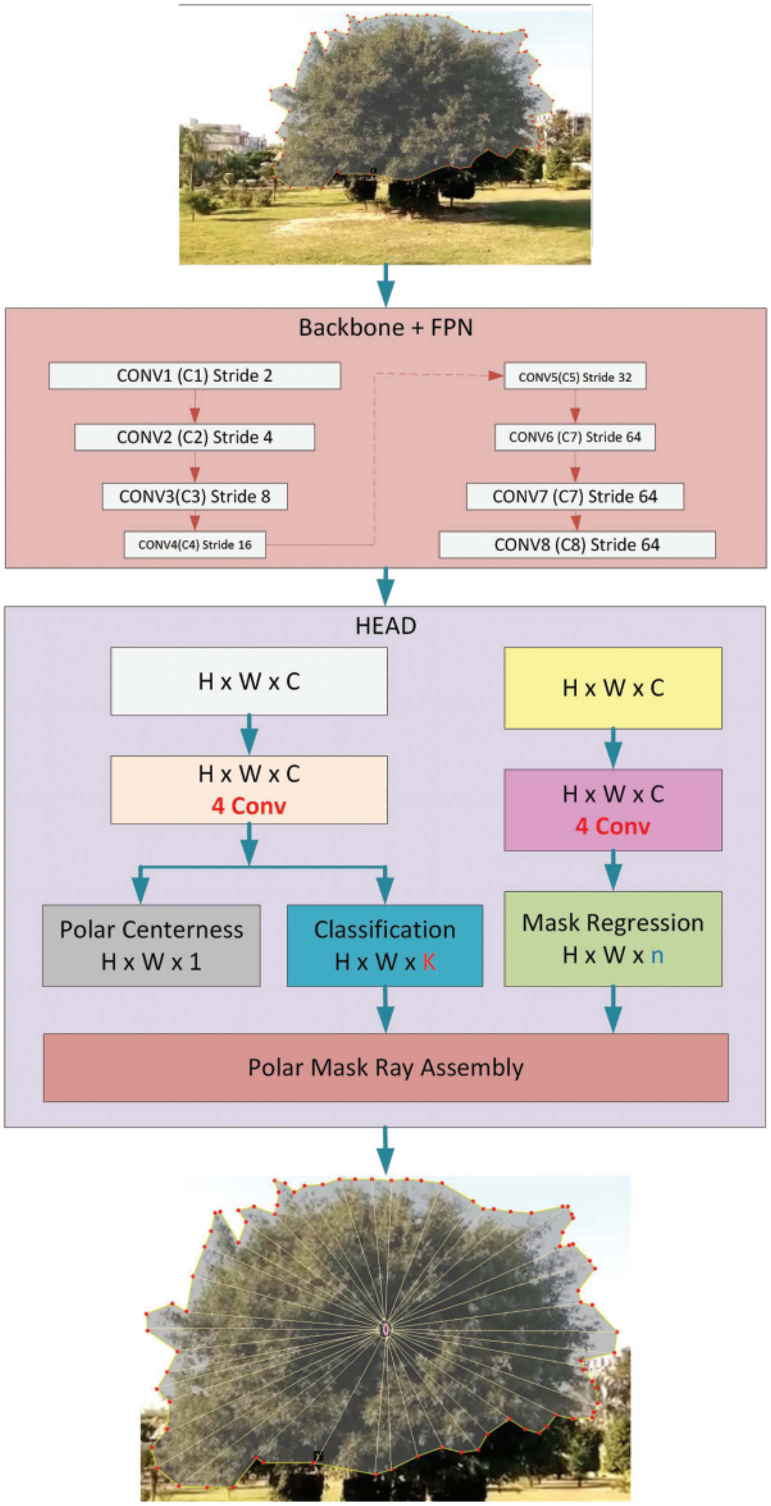

Figure 4: Architecture for collision avoidance using polar mask

$$Polar\ IoU\ Loss = \log \frac{\sum_{j=1}^{n} b_j^{max}}{\sum_{j=1}^{n} b_j^{min}} \tag{11}$$

Integrating the differential Intersection Over Union (IoU) distribution in terms of differentially angles yields for the mask-IoU in polar coordinates. Polar-IoU loss improves the mask regression overall rather than improving every ray individually and resulting in higher efficiency. Mask IoU is found by using polar integration.

$$IoU = \frac{\int_0^{2\pi} \frac{1}{2} min(b, b^*)^2 d\theta}{\int_0^{2\pi} \frac{1}{2} \max(b, b^*)^2 d\theta} \tag{12}$$

The architecture for the design of the collision avoidance system is shown in Fig. 3 consists of backbone network and the feature pyramid network (FPN) having eight convolution layers with different stride sizes. The designated polar rays are marked from the center using Eq. (9) and the least distance calculated for each ray from all directions.

The major effect of using Polar mask segmentation is to provide a length of predicted rays which must be similar to the target rays, once the rays are equal to IoU which calculates the minimized mask in the polar space. Feature Pyramid Network (FPN) may also be refined used in the backbone network by re-scaling into a different level of feature maps that have been achieved by contextual information.

Eqs. (9), (10) and (12) is used to provide the mechanism to calculate the least distance from the center of the obstacle object to the end of the boundaries marked during the rays are regressed. Once the least distance is calculated either from left to right or down to up or in inverse directions, the reward function mentioned above activated for the avoidance of collision with the object (tree).

The velocities of the brushless DC electric motors will be minimized to take the hover position, once the obstacle is detected inside the GPS coordinates circle is shown in Fig. 3.

4 Results

The results of the proposed framework are divided into three-part (i) The reward estimation for six different hand gestures using Deep Deterministic Policy using Actor critic Network and (ii) PID based controller results for the analysis between the RL based controller. (iii) Accuracy and loss results for the Polar Mask segmentation.

4.1 Experimental Setup

The Nvidia Jetson nano with intel D435i depth camera is used for the experimentation. The UAV consists of 4 x brushless DC electric motors, F450 UAV chassis, Electronic Speed Controller (ESC) four quantity, 10-inch four quantity fiber propellers, Power distribution box (PDB) for connecting different wires from motors, batteries, landing gears, and Inertial Measurement Unit (IMU). The 40 General Purpose Input/Output (GPIO) pins of Jetson Nano embedded board contains 4 × I2C pins, 4 × Universal Asynchronous Receiver-Transmitter (UART) Pins, 1 × 5 V pin, 2 × 3V3, and 3 Ground Pins other 26 GPIO Pins. The pin # 3 (SDA) on jetson nano connected with pin # 27 Serial Data Pin (SDA) on IMU and Pin # 5 Serial Clock Pin (SCL) on jetson nano with Pin # 28 (SCL) on IMU. We send the Pulse Width Modulation (PWM) signal from pin # 33 to ESC which operates at 3.3v and sends the 3-phase supply to brushless DC electric motors.

In the environment created on Ubuntu 18.04, different libraries of deep learning installed consisting of NumPy, Pandas, Tensor Flow, and Keras. For the DDPG agent, we used Actor-Critic network followed by a reply buffer for the storage of reward functions during the training. The reset function self. reset () has been created. This function is activated when it follows the wrong path during the training. Multiple epochs are considered during the training for which maximum reward achieved on 2500 epochs by hit and trial mechanism which resulted to stabilized for six different reward functions is shown in Fig. 5.

Fig. 5 describes the reward estimation for the implementation of six different hand gestures. The best reward estimation was observed during the training of 2500 episodes.

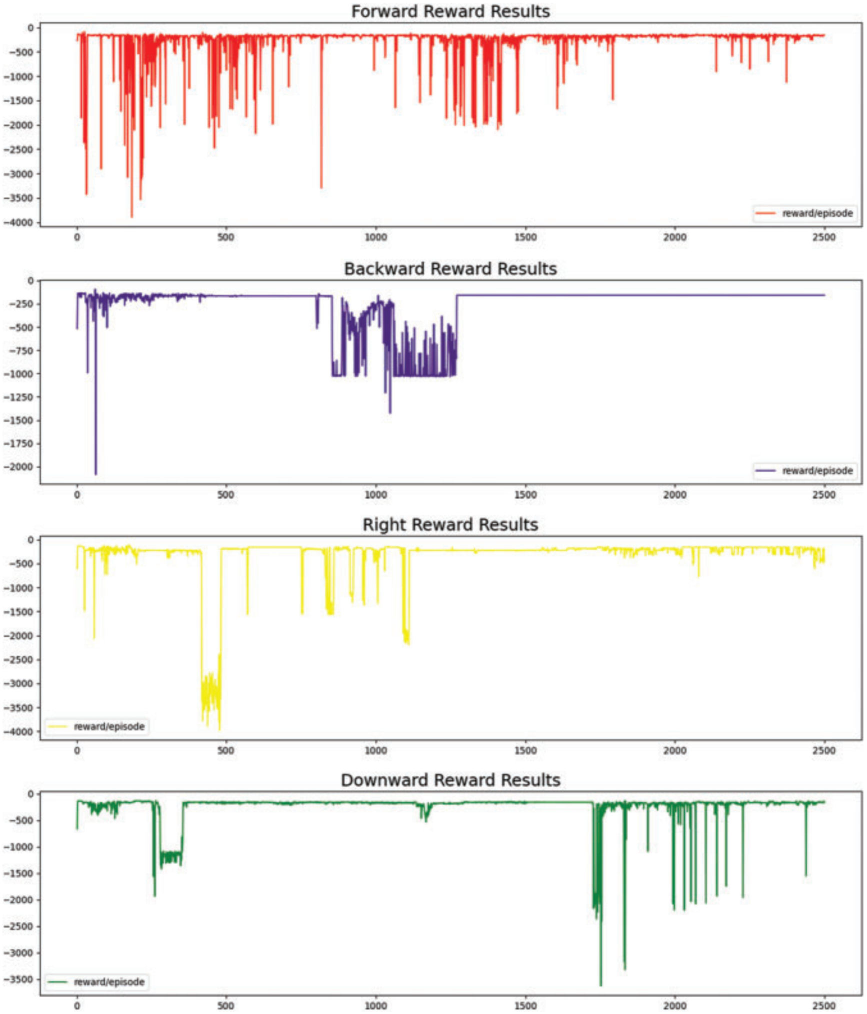

Figure 5: (Continued)

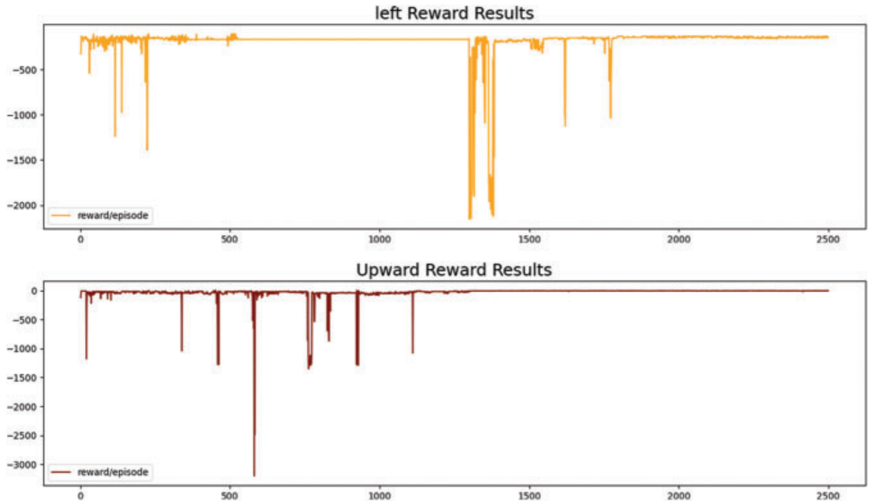

Figure 5: Reward estimation with six different 3D hand gestures

The training cycle of 30 epochs with 1560 iterations configured 52 iterations per epochs with the learning rate of 3.2e-08 for the calculation of Polar Mask for collision avoidance, below Fig. 6. demonstrates the accuracy and loss where black dots show the validation cycle. The received accuracy is 86.36%.

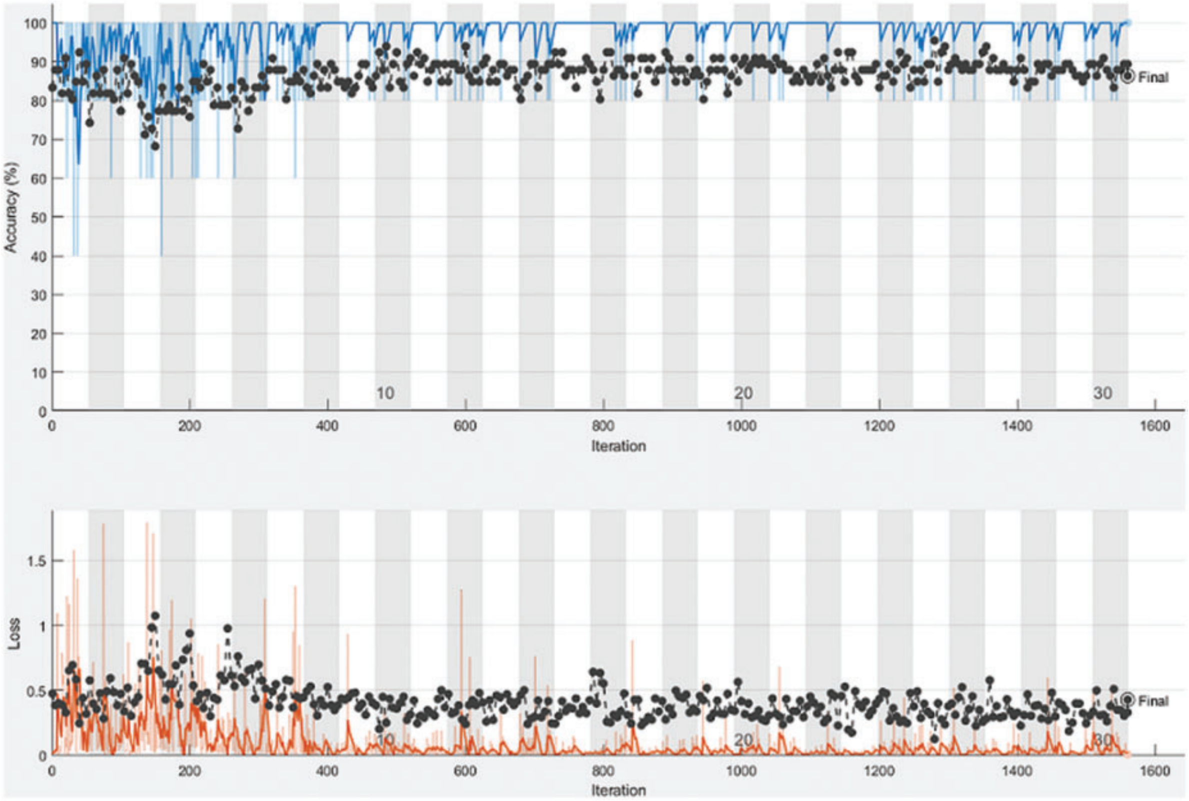

Figure 6: Estimations of accuracy and loss for polar mask segmentation

5 Analysis and Discussion

A PID controller may also be used to adjust and train the Reinforcement Learning (RL) algorithms. The controller updates the reward values and the next action based on the inputs and observations of the UAV's current state. The PID controller receives data from onboard sensors as well as the value of the three gains used to assess the system's durability. The analysis has been made both for PID and RL-based controller, it is quite obvious that after the training for 2500 episodes, the reward functions for six different hand gestures provide the best accuracy and control for UAVs using the proposed framework. Figs. 7 and 8 below shows the values of pitch and roll from the PID-based controller. PID controller utilized to train on fewer iterations where RL based controller trained on a minimum of 2500 epochs.

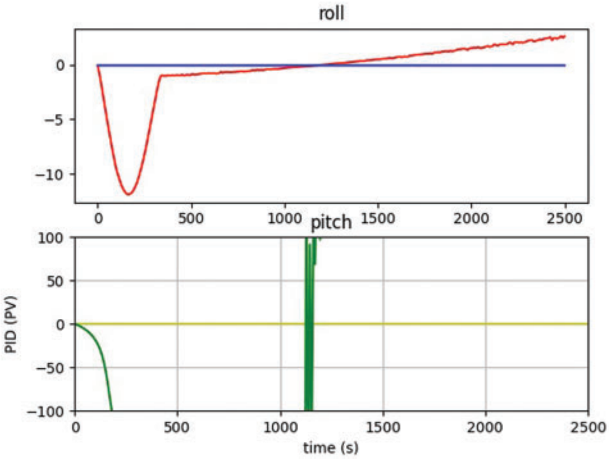

Figure 7: PID controller initialized state without 3D hand gestures

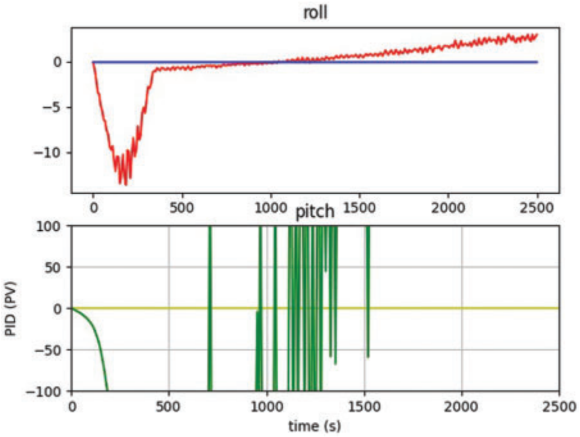

Figure 8: PID controller initialized state without 3D hand gestures

It was also observed that the polar mask technique used for collision avoidance provided better results without using any sensors to stop the UAVs, the system has calculated the center location and marked the edges and construct the rays with different angles. The segmented image once marked with IoU then calculate the least distance with the center locations, the distance then utilized for the activation of reward functions to move the UAV for collision avoidance. The initial threshold for the distance between the UAV and the obstacle (tree) was set for 5 feet and marked before the experimentation. Figs. 9 and 10 shows the attitude control obtained during collision avoidance while the movement of UAV calculated through roll and pitch in back-and-forth direction, the IMU is calibrated and the offset in the accelerometer and gyroscope removed from the initial values, so the graph of the pitch only changes and the graph of roll smoothen near zero.

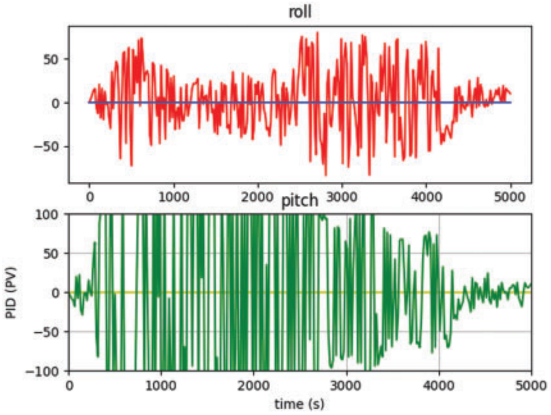

Figure 9: : Attitude control after calibration of IMU

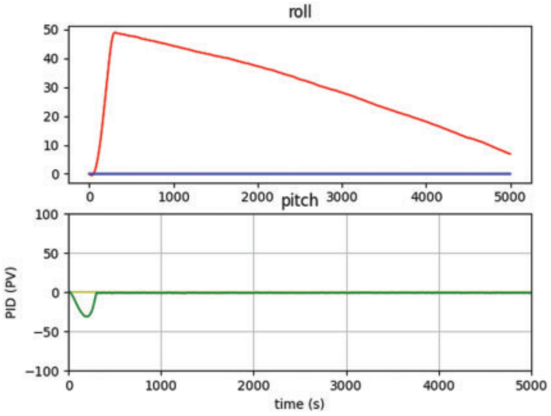

Figure 10: : Attitude control before calibration of IMU

6 Conclusion

Deep reinforcement learning has revolutionized the area of UAV route planning, navigation, and control. Luckily, advances in DRL controller design and UAV mechanical architecture are constantly being created and evaluated. As a result, new difficult tasks and uses for various types of UAVs have emerged.

The state of art reinforcement learning UAV control with 3D hand gestures provided evident contribution in the field of robotics. There are some environmental factors including wind speed, rainfall, and dirt which must be addressed while improving whole systems because they create ambiguity in outcomes. As a result, it should be classified as a system disruption and dealt with properly. The limitation of detecting 3D hand gestures due to the FOV of the camera ranging to 3 meters can be removed by replacing a better range of FOV of the camera.

The reward function, which is defined by the UAV's behaviors, is important to using RL in UAV navigation. The designed reward functions imply the best stability during training with 2500, 5000, 7500, and 10000 episodes where it has observed the maximum reward received on 2500 episodes. The computational time on NVidia jetson nano observed for each episode is 15 micro second during training. The system works by continuous modification of the UAV state depending on data produced by onboard sensors, and calculating the best course of action and associated reward values.

For future work, the collision avoidance system may be improved by replacing the GPS sensor with the Camera Field of View (FOV) to avoid limitations with GPS and its accessories.

Acknowledgement: This research was funded by Yayasan Universiti Teknologi PETRONAS (YUTP), grant number 015LC0-316, and the APC was funded by Research Management Centre, Universiti Teknologi PETRONAS, Malaysia under the same grant.

Funding Statement: The hardware used in this research consisting of Jetson nano, brushless DC electric motor, sensors, camera, UAV body, batteries are made available by CONVSYS (Pvt. Ltd., Islamabad) Pakistan.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. S. Khan, M. N. H. Mohd, D. M. Soomro, S. Bagchi and M. D. Khan, "3D hand gestures segmentation and optimized classification using deep learning," *IEEE Access*, vol. 9, pp. 131614–131624, 2021, https://doi.org/10.1109/ACCESS.2021.3114871.

[2] F. S. Khan, M. N. H. Mohd, R. M. Larik, M. D. Khan, M. I. Abbasi *et al.,* "A smart flight controller based on reinforcement learning for unmanned aerial vehicle (UAV)," in *2021 IEEE Int. Conf. on Signal & Image Processing Applications*, Malaysia, no. c, pp. 203–208, 2021, https://doi.org/10.1109/icsipa52582.2021.9576806.

[3] A. T. Azar, A. Taher, A. Koubaa, N. A. Mohamed, H. A. Ibrahim *et al.,* "Drone deep reinforcement learning: A review," *Electronics*, vol. 10, no. 9, pp. 1–30, 2021.

[4] S. Dhargupta, M. Ghosh, S. Mirjalili and R. Sarkar, "Selective opposition based grey wolf optimization," *Expert System Applications*, vol. 151, pp. 113389, 2020, https://doi.org/10.1016/j.eswa.2020.113389.

[5] W. Koch, R. Mancuso, R. West and A. Bestavros, "Reinforcement learning for UAV attitude control," *arXiv*, vol. 3, no. 2, pp. 1–21, 2018.

[6] M. U. Rehman, F. Ahmed, M. A. Khan, U. Tariq, F. A. Alfouzan *et al.,* "Dynamic hand gesture recognition using 3D-CNN and LSTM networks," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 4675–4690, 2022.

[7] A. Khan, M. A. Khan, M. Y. Javed, M. Alhaisoni, U. Tariq *et al.,* "Human gait recognition using deep learning and improved ant colony optimization," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2113–2130, 2022.

[8] X. Enze, W. Wang, M. Ding, R. Zhang *et al.* "PolarMask++: Enhanced polar representation for single-shot instance segmentation and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[9] N. Passalis and A. Tefas, "Continuous drone control using deep reinforcement learning for frontal view person shooting," *Neural Computing Applications*, vol. 32, no. 9, pp. 4227–4238, 2020.

[10] M. Liaq and Y. Byun, "Autonomous UAV navigation using reinforcement learning," *International Journal of Machine Learning and Computing*, vol. 9, no. 6, pp. 756–761, 2019.

[11] S. Amr and S. Qin, "Robust adaptive flight controller for UAV systems," in *Proceeding - 2017 4th Int. Conf. on Information. Science and Control Engineering ICISCE 2017*, Changsha, China, no. 1, pp. 1214–1219, 2017, https://doi.org/10.1109/ICISCE.2017.252.

[12] E. Ebeid, M. Skriver and J. Jin, "A survey on open-source flight control platforms of unmanned aerial vehicle," in *Proceeding - 20th Euromicro Conference on Digital System Design (DSD) 2017*, pp. 396–402, 2017, https://doi.org/10.1109/DSD.2017.30.

[13] Z. Cheng, R. West and C. Einstein, "End-to-end analysis and design of a drone flight controller," *arXiv*, vol. 37, no. 11, pp. 2404–2415, 2018.

[14] K. Tanaka, M. Tanaka, Y. Takahashi, A. Iwase and H. O. Wang, "3-D flight path tracking control for unmanned aerial vehicles under wind environments," *IEEE Transactions. Vehicle Technologies*, vol. 68, no. 12, pp. 11621–11634, 2019.

[15] M. Chen, S. Xiong and Q. Wu, "Tracking flight control of quadrotor based on disturbance observer," *IEEE Transactions on System, Man and Cybernatics System*, vol. 51, no. 3, pp. 1414–1423, 2021.

[16] A. R. Al Tahtawi and M. Yusuf, "Low-cost quadrotor hardware design with pid control system as flight controller," *Telkomnika (Telecommunication Comput. Electron. Control*, vol. 17, no. 4, pp. 1923–1930, 2019.

[17] B. Duo, "Energy efficiency maximization for full-duplex UAV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4590–4595, 2020.

[18] B. Mu, K. Zhang and Y. Shi, "Integral sliding mode flight controller design for a quadrotor and the application in a heterogeneous multi-agent system," *IEEE Tranactions on. Industrial Electronic*, vol. 64, no. 12, pp. 9389–9398, 2017.

[19] J. Zhang, J. Yan, P. Zhang and X. Kong, "Collision avoidance in fixed-wing UAV formation flight based on a consensus control algorithm," *IEEE Access*, vol. 6, pp. 43672–43682, 2018, https://doi.org/10.1109/ACCESS.2018.2864169.

[20] J. Zhao, F. Gao, G. Ding, T. Zhang, W. Jia *et al.,* "Integrating communications and control for UAV systems: Opportunities and challenges," *IEEE Access*, vol. 6, pp. 67519–67527, 2018, https://doi.org/10.1109/ACCESS.2018.2879637.

[21] G. E. M. Abro, S. A. Zulkifli, V. S. Asirvadam and Z. A. Ali, "Model-Free-Based Single-Dimension Fuzzy SMC Design for Underactuated Quadrotor UAV," in *Actuators*, vol. 10, no. 8, pp. 191. Multidisciplinary Digital Publishing Institute, 2021.

[22] H. Gu, X. Lyu, Z. Li, S. Shen and F. Zhang, "Development and experimental verification of a hybrid vertical take-off and landing (VTOL) unmanned aerial vehicle(UAV)," in *2017 Int. Conf. on Unmanned Aircraft Systems ICUAS 2017*, Miami, FL, USA, pp. 160–169, 2017, https://doi.org/10.1109/ICUAS.2017.7991420.

[23] M. K. Z. B. A. Mutalib, M. N. H. Mohd, M. R. B. M. Tomari, S. B. Sari and R. Bin Ambar, "Flying drone controller by hand gesture using leap motion," *International Journal of Advance Trends in Computer Science and Engineering*, vol. 9, no. 1.4 Special Issue, pp. 111–116, 2020.

[24] A. Sarkar, K. A. Patel, R. K. G. Ram and G. K. Capoor, "Gesture control of drone using a motion controller," in *2016 Int. Conf. on Industrail Informatics and Computer System CIICS 2016*, Sharjah, United Arab Emirates, 2016, https://doi.org/10.1109/ICCSII.2016.7462401.

[25] C. Wang, Y. Niu, M. Liu, T. Shi, J. Li *et al.,* "Geomagnetic navigation for AUV based on deep reinforcement learning algorithm," *IEEE Int. Conference on Robotic Biomimetics, ROBIO*, vol. 2, no. December, pp. 2571–2575, 2019, https://doi.org/10.1109/ROBIO49542.2019.8961491.

[26] C. C. Tsai, C. C. Kuo and Y. L. Chen, "3D hand gesture recognition for drone control in unity," in *IEEE Int. Conf. on Automation Science and. Engineering*, Hong Kong, China, vol. 2020-Augus, pp. 985–988, 2020, https://doi.org/10.1109/CASE48305.2020.9216807.

[27] J. Hwangbo, I. Sa, R. Siegwart and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Autommation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[28] H. Bou-Ammar, H. Voos and W. Ertel, "Controller design for quadrotor UAVs using reinforcement learning," *Proc. IEEE Int. Conference on Control Applications*, Yokohama, Japan, pp. 2130–2135, 2010, https://doi.org/10.1109/CCA.2010.5611206.