engineering proceedings

MDPI

*Proceeding Paper*

# Prediction Error-Based Action Policy Learning for Quadcopter Flight Control †

**Jamal Shams Khanzada, Wasif Muhammad * and Muhammad Jehanzeb Irshad**

Intelligent Systems Laboratory, Department of Electrical Engineering & Technology, University of Gujrat, Gujrat 50700, Pakistan; jamalshamskhanzada@gmail.com (J.S.K.); jehanzeb.irshad@uog.edu.pk (M.J.I.)

* Correspondence: syed.wasif@uog.edu.pk

† Presented at the 1st International Conference on Energy, Power and Environment, Gujrat, Pakistan, 11–12 November 2021.

**Abstract:** Quadcopters are finding their place in everything from transportation, delivery, hospitals, and to homes in almost every part of daily life. In places where human intervention for quadcopter flight control is impossible, it becomes necessary to equip drones with intelligent autopilot systems so that they can make decisions on their own. All previous reinforcement learning (RL)-based efforts for quadcopter flight control in complex, dynamic, and unstructured environments remained unsuccessful during the training phase in avoiding the trend of catastrophic failures by naturally unstable quadcopters. In this work, we propose a complementary approach for quadcopter flight control using prediction error as an effective control policy reward in the sensory space instead of rewards from unstable action spaces alike in conventional RL approaches. The proposed predictive coding biased competition using divisive input modulation (PC/BC-DIM) neural network learns prediction error-based flight control policy without physically actuating quadcopter propellers, which ensures its safety during training. The proposed network learned flight control policy without any physical flights, which reduced the training time to almost zero. The simulation results showed that the trained agent reached the destination accurately. For 20 quadcopter flight trails, the average path deviation from the ground truth was 1.495 and the root mean square (RMS) of the goal reached 1.708.

**Keywords:** quadcopter; reinforcement learning; flight control; autonomous

check for updates

**Citation:** Khanzada, J.S.; Muhammad, W.; Irshad, M.J. Prediction Error-Based Action Policy Learning for Quadcopter Flight Control. *Eng. Proc.* **2021**, *12*, 47. https://doi.org/10.3390/engproc2021012047

Academic Editor: Qasim Awais

Published: 29 December 2021

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

**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

The increasing usage of modern flying machines is revolutionizing human life. One prime example of these machines is the quadcopter. The autonomous navigation and control of a quadcopter is an existing challenge that still requires significant efforts. Now, artificial intelligence (AI) is spreading wildly in every field, so it is not meaningless to mention that AI machines bring dramatic changes in the autonomous control of a quadcopter. All the state-of-the-art reinforcement learning (RL) quadcopter navigation and control techniques have failed to avoid catastrophic failure by naturally unstable quadcopters during the training phase in complex, dynamic, and unstructured environments. Another limitation associated with these RL-based training approaches is their reliance on action space reward, which is practically impossible without circumventing physical damage to the quadcopter. Using a proportional integral differential (PID) controller for the control of quadcopter does not allow to deal with external harsh environments [1]. Greatwood [2] combined two different techniques in two stages for two different jobs. RL works for node detection so that it can move to find a destination and model predictive control (MPC) operates to avoid obstacles. The experiment was successful, but the results be more appreciable if the nodes were defined using the RL technique. Zhang [3] and his team also used MPC along with RL in the framework of policy search. MPC was used to generate data which were then used to train a deep neural network that directly controls the rotor speed

using raw data. Mareef [4] used an online self-tunable fuzzy inference system to control the flight of a quadcopter. A fuzzy inference system is used to minimize cost function and follow a path trajectory. However, Yang et al. are not the only one who used two loops; Richard [5] suggested to use RL for the sake of altitude control of a unoccupied flying vehicle (UFV) during flight. They controlled the altitude via the inner loop using a PID controller and the outer loop was responsible for the intelligent flight control system trained using state-of-the-art of RL algorithms, trust region policy optimization, proximal policy optimization, and deep deterministic gradient policy; though it was only in a simulation test. In this paper, we propose a novel sensory space reward-based quadcopter navigation and control learning approach. The proposed approach uses predictive coding/biased competition using divisive input modulation (PC/BC-DIM) [6] neural network as a core component. The PC/BC-DIM neural network uses the prediction error computed from the input sensory information as a reward for quadcopter training, which circumvents the limitations of conventional RL approaches.

## 2. Methodology

The architecture of the PC/BC-DIM neural network is shown in Figure 1. It can be considered as a cycle that starts by generating reconstruction neurons from prediction neurons on the basis of error neurons taken from inputs, as shown in Figure 1.

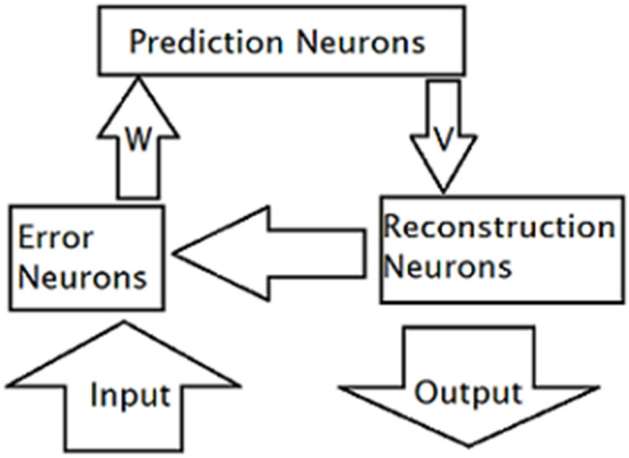


**Figure 1.** Predictive coding/biased competition using divisive input modulation (PC/BC-DIM) network.

The figure shows how the encoded input is compared to find an error on the basis of the error neurons that are calculated that generate synaptic weights ($W$). This weight, in combination with error neurons, make the prediction neurons. Finally, the prediction neuron is multiplied by the $V$ matrix to make reconstruction neurons. This is a loop for limited iterations. The equations represented by the blocks are as follows (Equations (1)–(3)).

$$r = Vy \tag{1}$$

$$e = \varepsilon/(x + r) \tag{2}$$

$$y = (\varepsilon + y)/We \tag{3}$$

Here, $r$ is reconstruction neuron, $V$ is a transpose of weight matrix $W$, $y$ is the prediction neuron, $x$ is the encoded input, $e$ stands for error and epsilon is a constant term introduced to reduce the chances of zero error. Reconstruction neurons are a major part of this PC/BC-DIM network to control the quadcopter. The whole research work is on the laminar flow of two different systems. The first one is the data acquisition part or say training that encodes a specific range of the aircraft principal axis, i.e., pitch and roll and then the navigation phase works as the main controller of the quadcopter, as shown in Figure 2.

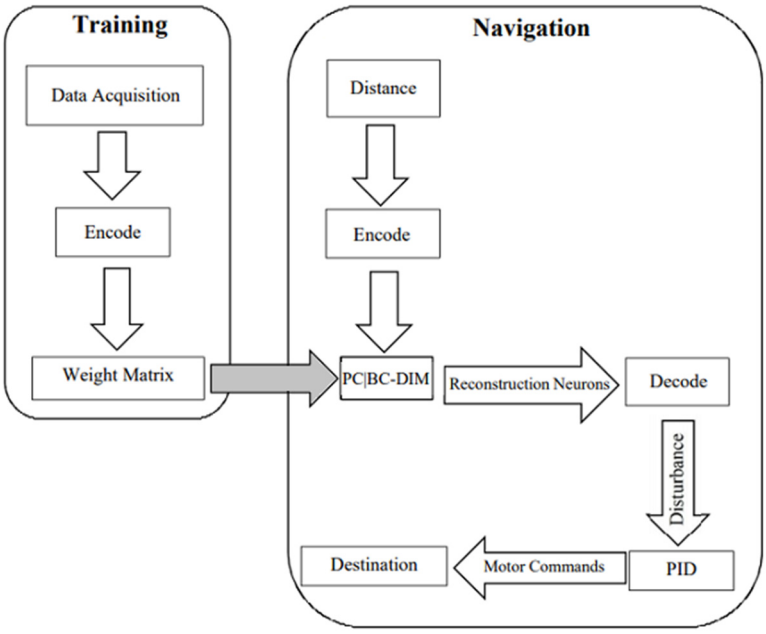


**Figure 2.** Workflow of the control system.

### *2.1. Training*

The training phase acquires predefined data as a set of ranges of roll and pitch disturbances. These disturbances are the prime movers for any aircraft. Each value is encoded and the set is saved as a fixed-weight matrix. This matrix is then fed into the network. The working of the training phase can be understood by following a simple algorithm (Algorithm 1).

**Algorithm 1.** Training

1. distribute aircraft principal axis and maximum distance
2. for iterations in range of size (distribution)
3. encode attribute (pitch or roll)
4. concatenate attributes in a vector
5. arrange in weight matrix

Figure 2 shows that raw data are encoded in a Gaussian format. This encoded value is arranged in a matrix. This matrix serves as a synaptic weight for the PC/BC-DIM network.

### *2.2. Navigation*

In the navigation phase, the distance between the quadcopter and the goal is measured via point distance formulae (Equations (4) and (5)):

$$x_{distance} = x_{goal} - x_{aircraft} \tag{4}$$

$$y_{distance} = y_{goal} - y_{aircraft} \tag{5}$$

First of all, the exact distance between the quadcopter and the goal location is measured. The distance is encoded and works as an input for the PC/BC-DIM network. The network iterates 150 times and the resulting reconstruction neurons are decoded to extract the exact roll/pitch disturbance that moves the quadcopter. This decoded value is fed to the PID controller. This controller generates direct actuator commands to adjust the rpm of connected motors. This change in the angular velocity of motors gives the requested thrust and roll necessary to move the aircraft. Once the motor commands are generated, they move the quadcopter and eventually the distance is measured again. If the distance is in the specific range of the goal, brakes are applied and the quadcopter stops. Algorithm 2 can help to understand the phase better.

**Algorithm 2.** Navigation

1. while distance $\neq$ 0:
2. read GPS and gyro sensors
3. distance = goal GPS-current GPS reading
4. encode distance
5. PC/BC-DIM $\leftarrow$ distance
6. reconstruction neurons $\leftarrow$ PC/BC-DIM
7. disturbances $\leftarrow$ decoded reconstruction neurons
8. PID $\leftarrow$ disturbances

## 3. Results

Twenty successful simulation experiments for random test goals were performed with the help of open-source software, "Webots 2021" (Cyberbotics Ltd., Lausanne, Switzerland). DJI Mavic 2 Pro (DJI, Shenzhen, China) (the used quadcopter) can operate smoothly with a pitch of two units and roll of one unit. Crossing this threshold will make quadcopter unstable. This pitch and roll was distributed into small steps, equal to the steps of the maximum distance. Each value of the dataset is encoded and arranged into a vector form. Thus, we get the vector of one specific value. The next value is encoded and concatenated to the previous one, generating a huge synaptic weight matrix, as explained in the training phase. Next the navigation phase that utilizes the synaptic weights in the PC/BC-DIM network occurs and later generates the specific command according to the need. As the navigation phase starts, it measures the distance from the goal and then encodes it. This encoded distance later acts as an input for the network. This input executes the network that generates the reconstruction neurons. Firstly, the pitch and roll parts from the whole reconstruction neuron are separated and then decoded to get the disturbance value to feed the PID controller. The PID controller generates motor commands to move the quadcopter in the desired direction. The proposed reward-lessness method from action space was successfully tested and satisfactory results were obtained, proving the correctness of the method. The quadcopter successfully took off from ground level and moved to the defined 3D goal location by maintaining its stability and robustness. It precisely reached the goal location and started hovering. For the sake of showing results, a location of (5, 1, 2) is assumed (Figure 3). Altitude is assumed as a *z*-axis and is 1 unit. Twenty successful experiments proved the average of difference between the x coordinate of the quadcopter and the ground truth was l.02 and that of the y coordinate of the quadcopter to the ground truth was 0.347. The root mean square error value from the ground truth of aircraft was 1.09.

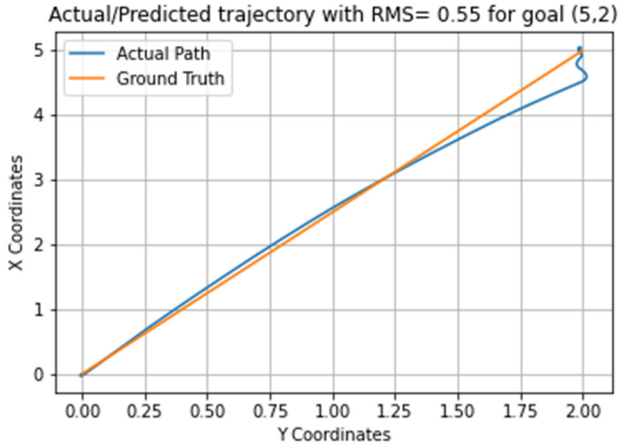


**Figure 3.** Simulation results.

It can be clearly seen that the quadcopter is at the desired location in a minimal amount of time. The best part of this method is that the quadcopter can go beyond the training limits.

## 4. Conclusions

This paper presents a novel prediction error-based quadcopter navigation control approach using the PC/BC-DIM neural network. The proposed approach overcame the limits of old RL techniques and had an accuracy of RMS error of 1.09.

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

## References

1. Maleki, K.N.; Ashenayi, K.; Hook, L.R.; Fuller, J.G.; Hutchins, N. A reliable system design for nondeterministic adaptive controllers in small UAV autopilots. In Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 25–29 September 2016; pp. 1–5.
2. Greatwood, C.; Richards, A.G. Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control. *Auton. Robot.* **2019**, *43*, 1–13. [CrossRef]
3. Zhang, T.; Kahn, G.; Levine, S.; Abbeel, P. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 528–535.
4. Zemalache, K.M.; Maaref, H. Controlling a drone: Comparison between a based model method and a fuzzy inference system. *Appl. Soft Comput.* **2009**, *9*, 553–562. [CrossRef]
5. Koch, W.; Mancuso, R.; West, R.; Bestavros, A. Reinforcement learning for UAV attitude control. *ACM Trans. Cyber-Phys. Syst.* **2019**, *3*, 22. [CrossRef]
6. Spratling, M.W. A neural implementation of Bayesian inference based on predictive coding. *Connect. Sci.* **2016**, *28*, 346–383. [CrossRef]