

# Modeling and prediction of powered parafoil unmanned aerial vehicle throttle and servo controls through artificial neural networks

Prashant Kumar<sup>DOI<sup>a</sup></sup>, Bisheswar Choudhury<sup>b</sup>, Amandeep Singh<sup>b</sup>, Janakarajan Ramkumar<sup>b</sup>, Deepu Philip<sup>c</sup>, and Ajoy Kanti Ghosh<sup>a</sup>

<sup>a</sup>Department of Aerospace Engineering, IIT Kanpur, Uttar Pradesh 208016, India; <sup>b</sup>Department of Mechanical Engineering, IIT Kanpur, Uttar Pradesh 208016, India; <sup>c</sup>Department of Industrial and Management Engineering, IIT Kanpur, Uttar Pradesh 208016, India

Corresponding author: Prashant Kumar (emails: [prash@iitk.ac.in](mailto:prash@iitk.ac.in); [iitk.prash@gmail.com](mailto:iitk.prash@gmail.com))

## Abstract

This study proposes a framework for developing a realistic model for throttle and servo control algorithms for a powered parafoil unmanned aerial vehicle (PPUAV) using artificial neural networks (ANNs). Two servo motors on an L-shaped platform, control and steer the PPUAV. Six degrees of freedom mathematical model of a dynamic parafoil system is built to test the technique's efficacy using a simulation in which disturbances mimic actual flights. A guiding law is then established, including the cross-track error and the line-of-sight approach. Furthermore, a path-following controller is constructed using the proportional-integral derivative, and a simulation platform was created to evaluate numerical data illustrating the route's validity following the technique. PPUAV was developed, built, and instrumented to collect real-time flight data to test the controller. These dynamic characteristics were sent into the ANN for training. A diverging-converging design was identified to obtain the best consistency between predicted and observed throttle and servo control values. For a comparable flight route, the control signal of the simulated model is compared with those of the actual and ANN-predicted models. The comparative findings show that the ANN-predicted and actual control inputs were almost identical, with an 80%–99% match. However, the simulated response showed deviation from the actual control input, with an accuracy of 50%–80%.

**Key words:** neural networks, artificial intelligence, parafoil, control systems, unmanned aerial vehicle

## 1. Introduction

Parafoils are flexible aerofoils with low-speed handling characteristics, usually used for precision delivery payloads or aerial recovery of platforms. These payloads may be military (ammunitions, vehicles, boats, etc.) or civil (disaster relief, rescue materials, etc.) loads. Sometimes, smaller hand-launched unmanned aerial vehicles (UAVs) are recovered on the ground using parafoils to avoid the runway requirement. A regular parachute is a hemispherical fabric, having rigging lines below to decrease the rate of descent by drag. However, a parafoil is a double-layered rectangular or ellipsoid-shaped fabric that inflates to create the aerofoil (i.e., the shape of an aircraft wing). Precision delivery or controlled air delivery systems refer to these systems where a payload is dropped from an airborne platform with servo actuators attached to the suspension lines to control the drop trajectory to land in a designated drop zone. Such systems lack a propulsion system and hence cannot takeoff from the ground or climb after drop to clear obstructing terrain features. Figure 1 depicts a powered parafoil unmanned aerial vehicle (PPUAV), a flying platform that derives lift from a ram-air inflated canopy act-

ing as a wing attached to the payload using suspension lines. The dynamic pressure of air flowing past the parafoil inflates it, creating a cross-sectional airfoil shape that, in turn, creates lift.

In contrast, conventional parachutes are simply used to create drag. Traditional parachute systems have round canopies that are either single or clustered. Such parachutes descend vertically and have minimum capability to control their horizontal motion. Thus, autonomously guided parafoils are light and efficient alternatives for payload delivery.

By matching a parafoil's ability to glide with the ability to control its turns and descent, it gives it maneuverability to be steered to a desired location. Modeling a parafoil using computational software is difficult and challenging due to its flexible fabric structure constantly changing its shape while flying. This makes it difficult for software tools like computational fluid dynamics to model their behavior.

Both traditional parachute and gliding parachute (or parafoil) systems are used in military, civil, and scientific areas for payloads ranging from a small to several

**Fig. 1.** The ram-air inflated canopy during the flight of the parafoil propelled vehicle (PPV) (Source: Flight Dynamics Laboratory, IIT Kanpur, India).



thousand pounds. Typical military uses include delivering personnel, supplies, and equipment to fighting fronts; resupplying in isolated and inaccessible places; and delivering ammunition, electronic countermeasures, sensors, and so on. There are several scientific applications for parachute systems, such as spacecraft recovery, sounding rockets, booster components for spacecraft, high-altitude balloon payloads, etc. Civil applications include airdrop of humanitarian aid, sport parachuting, stabilization, deceleration, etc. The proposed uses of powered parafoils are aerial spraying providing a slow-moving low-altitude platform for high-resolution photography and carrying ground-penetrating radar for detecting underground anomalies. Since PPUAVs have long endurance, they can also be used for reconnaissance applications like tracking forest fires, traffic monitoring, crowd monitoring, etc. Till now, powered parachutes have been utilized almost exclusively for recreational purposes. However, properties like cost effectiveness in manufacturing, even when compared with fixed-wing UAVs, the ability to perform slow speed (30–50 km/h) operation with pendulum stability, lightweight, easily packable into the small system, and performing accurate and smooth landing while carrying a sensitive payload make them an attractive alternative platform for UAV applications (Smith et al. 1999).

Aerial cargo delivery was realized in 1991 when Grant Morton jumped from an aircraft with a parachute. Jalbert discovered parafoil in the early 1960s (Iacomini and Cerimele 1999), and later refined and named “Parafoil” because it combines a parachute and airfoil. Parafoils have a high lift-to-drag ratio of 1:5, improving glide performance. Autonomously guided parafoils were first considered in 1960 to assess the feasibility of reaching a desired impact location designated by a homing beam by Nicolaides (1970). Similar attempts by the U.S. Army in the 1970s were of limited success due to a lack of navigation data and the nonavailability of onboard computers (Murray et al. 2013).

Global positioning system (GPS) was opened to the public in the 1990s, after which embedded computer systems with small form factors were readily available. ORION airborne delivery system demonstrated its capability of flying payload

varying from 200 to 28 000 lbs using parafoils with an accuracy of 100 m. National Aeronautics and Space Administration started evaluating parafoil feasibility for autonomous spacecraft recovery during re-entry, eventually resulting in the full-scale X-38 Crew Recovery Vehicle Program, as mentioned by Goodrick et al. (1973). U.S. Army started the Guided Parafoil Airborne Delivery System to demonstrate the applicability of a high glide recovery system for stabilization, deceleration, and precise landing of different military payloads (Wailes and Harrington 2013).

One of the primary goals of studying powered parafoil systems is to develop a mathematical model that accurately represents the system’s dynamics. Six- and eight-degrees-of-freedom (6/8-DoF) versions are now the most popular for powered parafoil systems. A unique framework (Li et al. 2019) was built for precise powered parafoil modeling. This model is built in three stages: (i) creating a linear dynamic model; (ii) reducing the model structure; and (iii) calculating the model mismatch due to model variance and external disturbances. For this purpose, Yakimenko (2005) and Slegers (2010) constructed an 8-DoF dynamic model of a parafoil system constrained by force and moment and studied the open and closed loop responses to a turn instruction. An explanation of how a powered parafoil’s dynamic model is constructed was provided in Tan et al. (2020). Lagrangian equations and dynamic constraints are used to create a model with 6-DoF for the canopy and 2-DoF for the payload. Constraints on velocity, angular velocity, and force are introduced, and the whole modeling procedure is shown. Researchers (Wise 2006; Zhu et al. 2015) considered the canopy’s motion in relation to the payload; two coordinates were constructed independently to assess the canopy’s and the payload’s motion features. An 8-DoF dynamic model of the propelled parafoil was constructed using the Kirchhoff motion equation (Yakimenko 2005; Tan et al. 2020). Gliding, adjusting, landing with a flare, and reactions to wind and power are the primary actions depicted in this simulation. Hua et al. (2013) constructed a basic static model that may be used for quick estimating or optimization design, and they also elaborated on the links between the various configuration settings and the resulting thrust.

The guidance-based trajectory tracking technique (Kumar et al. 2020; Zhu and Gao 2020), which follows the reference point on the trajectory that moves according to a scalar variable, is the most popular way to track parafoil trajectories. Using Lyapunov's approach, Breivik (2005) presented this tactic and demonstrated that the trajectory error converges under specific conditions. Parafoil tracking was implemented using a guidance-based trajectory tracking strategy. Using this strategy and combining active disturbance rejection control with wind feedforward compensation, Luo et al. (2019) proposed a control strategy for accurately tracking the flight path of a powered parafoil. They used a generalized predictive control-based method for parafoil systems to monitor the specified trajectory to enhance the control effect. They also developed a hybrid approach combining lateral track error and the line of sight to build guiding law (Tao et al. 2019).

Recent years have shown many significant developments in neural computing techniques in aviation. Over a few years, the business has actively flight-tested the usage of a neural network for visual landing guidance in air taxis, drones, and a Cessna 180. Avidyne, an avionics company located in Melbourne, Florida, has begun testing Daedalean's neural networks in a camera and autopilot setup. Technology cooperation was also inked with Honeywell Aerospace to create autonomous takeoff and landing technologies, GPS-independent navigation, and collision avoidance. The model and corresponding experimental data are used to form an empirical neural network to model unmanned aerial systems or aircraft (Ben Mosbah et al. 2016; Egorchev and Tiumentsev 2018). Various literature is available for modeling aerodynamic characteristics, aerodynamics database improvement, and prediction of aerodynamics coefficients based on neural networks and hybrid neuro-fuzzy systems. Ignatyev and Khrabrov (2015) compare different neural network architectures, namely feedforward and recurrent, for modeling of aerodynamic characteristics of canard aircraft at high angle of attack. Various types of literature are available on the parafoil-payload systems, which describe 9-DoF (Mooij et al. 2003; Prakash and Ananthkrishnan 2006; Gang 2015), 8-DoF (Yakimenko 2005; Tan et al. 2020) mathematical models, and different control strategies such as proportional-integral derivative (PID), nonlinear dynamic inversion, and model predictive control (Qian and Chen 2010).

However, this technique's application for controlling flexible parafoil is quite scarce. Most of the aviation control field studies are toward higher order chaos in neural systems or fuzzy logic (Harth 1983; Wu et al. 2001). Similarly, genetic algorithms combined with fuzzy logic are successfully used in helicopter flight control (Phillips et al. 1996; Ortega et al. 1999). Talwar et al. (2017) and Jaiswal et al. (2020) investigated methodology to estimate the aerodynamic derivatives taking into account the flying vehicle system identification, using statistical machine learning technique. Several methods have been proposed to deal with the unmodeled factors in real-world scenarios. For example, some researchers have adopted data-driven methods to construct a mathematical description of a dynamical system. Neural networks, fuzzy logic, and expert system methods have proven helpful in some cases, es-

pecially with process control problems (Tsai et al. 2010; Wang et al. 2016; Zou et al. 2018).

The primary goal of this study was to create a PPUAV equipped with a suitable control mechanism and data acquisition (DAQ) system to record flight data in real time. The secondary goal was to fly a PPUAV, test, and record the data needed to create an effective deep neural network (DNN) that can be used to train, evaluate, and verify other models. We compared the DNN's projected data with the recorded flight data to determine the control's efficacy. This research also intended to provide a dependable and universal framework of dynamic design and control approaches for unmanned powered parachute aircraft. This study presents a set of simulation methods that may be used to build a reliable unmanned system that uses powered parachutes instead of multirotor or fixed-wing aircraft. Delphi programming language was chosen for its portability across several platforms and portable devices.

Additionally, the inbuilt graphic user interface allows for the development of DNN with five hidden layers with linear or sigmoidal output function. Also, it can handle up to 40 output parameters and any number of input parameters. Separate modules are created for training, testing, and validation of the DNN. The estimated neural network response is then compared with the control output of the mathematical model. The rest of the section is organized as follows. Section 2 discusses the dynamic model of a powered parafoil system with control architecture. The performance of the control architecture is tested in MATLAB/Simulink platform. Section 3 describes the experimental setup, the mission performed, and the neural network training parameters. Section 4 compares the predicted neural model and simulation results, followed by the conclusion in Section 5.

## 2. Dynamic model of powered parafoil system

Aerodynamics principles are based on rigid body dynamics assuming that the aircraft surface and structure are rigid and the airflow around it produces aerodynamic forces on the aircraft. Even though the structural deformity is accounted for, the surface deformation is difficult to model and quantify. The parachute's flexible and inflatable body makes it further difficult to estimate its control dynamics. Attempts to simulate parachute dynamics from first principles have resulted in minimum successes.

The low and stable operating speed of PPUAV comes from its pendulum stability, where the gondola moves ahead of the parafoil as the propeller speed is increased from the cruise condition. This increases the angle of attack of the parafoil, creating more lift, thereby PPUAV gains altitude while maintaining a stable speed between 30 and 50 km/h. Similarly, aileron reversal is another peculiar feature of the system in which by pulling the port servo actuators by 50%, the rear suspension control lines are pulled down, and the parafoil angle of attack increases, generating more lift. This results in the PPUAV turning in the opposite direction, the starboard direction. However, when the same servo is pulled by 100%, the

parafoil produces more drag on the port side, so the PPUAV turns in the same direction.

To simulate the aircraft in this manuscript, we used Euler angles. The Euler angle has just three parameters, resulting in a singularity on the pitch of 90 degrees. As a result, the Euler angle cannot be applied to a system with complex motion. On the other hand, the Euler angle provides a beneficial separation between tilt angle and heading angle inaccuracy. Because this system was not designed to execute sophisticated motion, we chose the Euler angle because there is a plethora of information available for the mathematical modeling of a parafoil payload system utilizing the Euler angle.

## 2.1. Simplified linear 6-DoF model of a parafoil

A 6-DoF dynamic model of the paramotor motion has been constructed to simulate the performance. Using Newton's second law, we obtain the dynamic equations for the translational and rotational DoF from the forces and moments (Umenberger and Göktoğan 2012; Kumar et al. 2020). Using the parafoil-payload system dynamics, a linearized equation of motion is created to lay out the control system and conduct stability analysis. The physical and inertial properties of the parafoil payload system are listed in Table 1 (Kumar et al. 2020). The total mass of the parafoil payload system is 1.880 kg. Detailed modeling of forces and moments is given in Kumar et al. (2020), and the derivatives were estimated based on the physical parameters and data acquired from the test flights. Lastly, for simulation, a few assumptions were taken into consideration as listed here: (i) at equilibrium, the effects of apparent mass and inertia are negligible; (ii) parafoil drag is dominant, and the parafoil center moves with the same velocity as the center of gravity (CG); (iii) moment at CG is only due to aerodynamic forces; and (iv) effect of wind is neglected. For steady-state flight, pitch ( $\theta_0$ ) is constant and hence  $q$  is zero, so the simplified matrix for controller design is given by eq. 1 (Kumar et al. 2020):

$$(1) \quad \begin{Bmatrix} \dot{\phi} \\ \dot{\psi} \\ \dot{p} \\ \dot{r} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/\cos\theta_0 \\ \frac{k_1 C_{l_p} b^2 J_{zz}}{2V_p} & 0 & k_1 C_{l_\phi} b J_{zz} & \frac{-k_1 C_{n_r} b^2 J_{xz}}{2V_p} \\ \frac{k_2 C_{l_p} b^2 J_{xz}}{2V_p} & 0 & k_2 C_{l_\phi} b J_{xz} & \frac{-k_2 C_{n_r} b^2 J_{xx}}{2V_p} \end{bmatrix} \begin{Bmatrix} \phi \\ \psi \\ p \\ r \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{k_1 b (C_{l_{\delta_a}} J_{zz} - C_{n_{\delta_a}} J_{xz})}{d} \\ \frac{k_2 b (C_{l_{\delta_a}} J_{xz} - C_{n_{\delta_a}} J_{xx})}{d} \end{bmatrix} \delta_a$$

$$k_1 = \frac{\rho S_p |V_p|^2}{2 (J_{xx} J_{zz} - J_{xz}^2)} \quad k_2 = \frac{\rho S_p |V_p|^2}{2 (J_{xz}^2 - J_{xx} J_{zz})}$$

where  $V_p$  is the parafoil velocity and rest of the parameters are listed in Kumar (2014) and Kumar et al. (2020).

Here,  $S_p$  is the parafoil area, and  $b$ ,  $c$ , and  $d$  are the span, chord, and depth of the parafoil, respectively. Iner-

tial parameters are represented by  $J_{(*)}$ , and  $\rho$  and  $g$  are density and acceleration due to gravity. Nondimensional derivatives corresponding to the longitudinal dynamics are given by  $[C_{D_0}^{Gd}, C_{D_\alpha}^{Gd}, C_{L_0}, C_{L_\alpha}, C_{D_\alpha}, C_{m_0}, C_{m_\alpha}, C_{m_q}]$ , where subscript "D" denotes the drag derivatives, "L" the lift derivatives, and "m", corresponding to pitching moment derivatives. Superscript "Gd" is the derivative specific to the gondola or payload.

Similarly, for lateral dynamics, the derivatives are given by  $[C_{l_p}, C_{l_\phi}, C_{l_{\delta_a}}, C_{n_r}, C_{n_{\delta_a}}]$ , where subscripts  $l$  and  $n$  denote coefficients with respect to rolling and yawing moments. Derivatives  $[C_{L_{\delta_a}}, C_{L_{\delta_s}}, C_{D_{\delta_a}}, C_{D_{\delta_s}}]$  represent the lift and drag coefficients due to symmetrical  $\delta_s = \min(\delta_L, \delta_R)$  and asymmetric deflection  $\delta_a = (\delta_L - \delta_R)$ .

## 2.2. Control architecture

This section covers an autopilot architecture that is appropriate for the sensors and computational resources available onboard the paramotor. A sequential loop closure was used to create lateral and longitudinal autopilots while considering saturation limitations. The inner loop has the most bandwidth, with each subsequent loop bandwidth being 5–10 times smaller in frequency. Each loop has a discrete-time version of the PID feedback control law.

### 2.2.1. Longitudinal and lateral control loops

Figure 2 describes the complete autopilot architecture for the simplified 6-DoF equations for the powered parafoil. The longitudinal autopilot design uses a combination of throttle and brake deflection to maintain a desired altitude and cruise speed. In longitudinal dynamics, the two primary control loops are the (i) pitch hold control loop and (ii) airspeed hold utilizing a throttle loop.  $T_{f_V}$ ,  $T_{d_o}$ , and  $T_V$  are all transfer functions defined in Beard and McLain (2010, 2012). Similarly, lateral autopilot is made up of two separate but interconnected loops: roll angle hold and heading angle hold. A gyroscope is used to measure the angle by which the system tilts with respect to the Y-Z plane serving as a reference orientation. It is possible to fine-tune the system's rolling angle via feedback based on the information supplied by the deviation signal for flap deflection. Heading control is used to keep the aircraft pointed on a predetermined course. An extra feedback loop is added to the circuit that controls the roll angle. Regulator control servos are built into the actuator block as a first-order function (Umenberger and Göktoğan 2012).

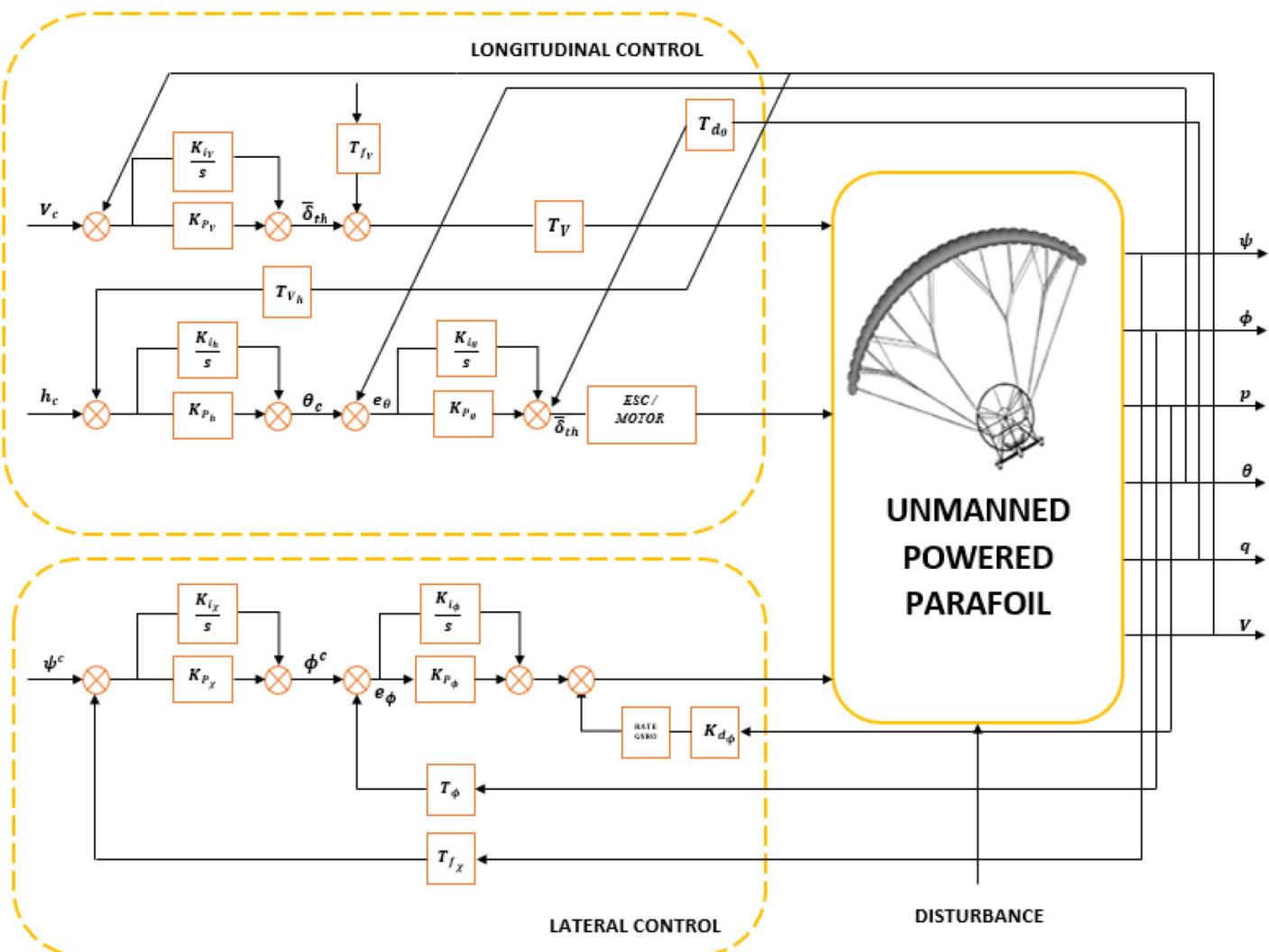
### 2.2.2. Guidance algorithm

In this part of the plan, the UAV will be programmed to follow a predetermined straight line by constructing a desired course angle at each spatial location. Rather than using a trajectory tracking approach, the focus is on the path following, where the objective is to be on the path rather than at a certain point at a particular time. With the path following, the time dependence of the problem is removed. Our objective was to develop a method for accurate path following in the presence of wind. A straight-line path is described by two vectors in  $R^3$ , namely  $P_{line}(r, q)$  where

**Table 1.** Mass, geometric, and inertial characteristics of parafoil and payload (Kumar et al. 2020).

Symbol	Mass (kg)	$\rho \text{ (kgm}^{-3}\text{)}$	g (ms $^{-2}$ )	$S_p \text{ (m}^2\text{)}$	b(m)	d(m)
Value	1.880	1.225	9.81	1.16	2.15	0.40
Symbol	$J_{xx} \text{ (kgm}^2\text{)}$	$J_{yy} \text{ (kgm}^2\text{)}$	$J_{zz} \text{ (kgm}^2\text{)}$	$J_{xz} \text{ (kgm}^2\text{)}$	$V_p \text{ (ms}^{-1}\text{)}$	c (m)
Value	0.0160	0.0186	0.0179	-0.0065	6	0.54
Symbol	$C_{D_0}^{Gd}$	$C_{D_\alpha}^{Gd}$	$C_{L_0}$	$C_{L_\alpha}$	$C_{D_\alpha}$	$C_{m_0}$
Value	0.15	1.00	0.034	2.52	0.277	0.016
Symbol	$C_{m_\alpha}$	$C_{m_q}$	$C_{l_p}$	$C_{l_\phi}$	$C_{l_{\delta_a}}$	$C_{n_r}$
Value	-0.065	-0.269	-0.108	-0.051	0.0031	-0.087
Symbol	$C_{n_{\delta_a}}$	$C_{L_{\delta_a}}$	$C_{D_{\delta_a}}$	$C_{L_{\delta_S}}$	$C_{D_{\delta_S}}$	
Value	0.0043	0.0001	0.0001	0.21	0.30	

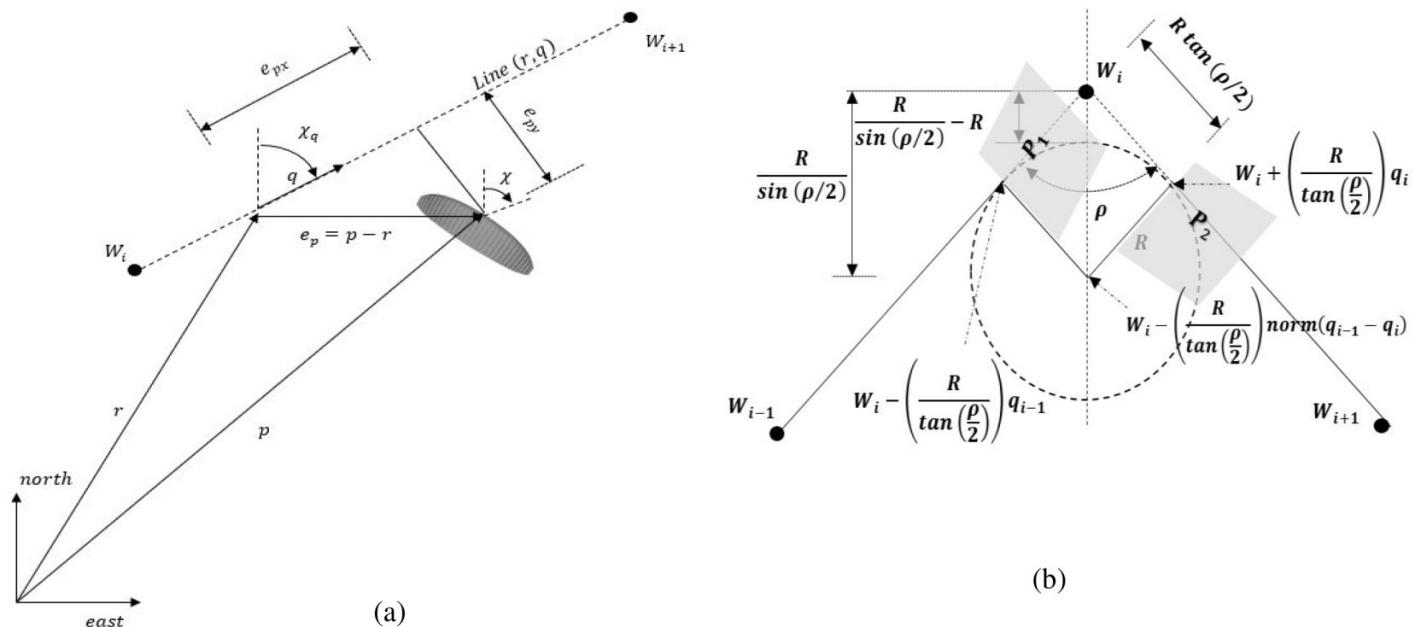
**Fig. 2.** Autopilot control logic for parafoil payload system (Beard and McLain 2010, 2012).



$r \in \mathbb{R}^3$  is the origin of the path, and  $q \in \mathbb{R}^3$ , a unit vector whose direction indicates the desired direction of travel. The course angle of *Pline* ( $r, q$ ), as measured from the north is given by  $\chi_q = \tan^{-1}(q_e/q_n)$ , where  $q = (q_n \ q_e \ q_d)^T$  expresses the unit direction vector's north, east, and down components.

An orbit path is described by a center  $c \in \mathbb{R}^3$ , a radius  $\rho \in \mathbb{R}$ , and a direction  $\lambda \in \{-1, 1\}$ , as  $P_{\text{orbit}}(c, \rho, \lambda) = r \in \mathbb{R}^3$ , where  $\lambda = 1$  signifies a clockwise orbit and  $\lambda = -1$  signifies a counterclockwise orbit. We assume that the center of the orbit is expressed in inertial coordinates so that

**Fig. 3.** (a) UAV position and orientation represented by  $(p, \chi)$ , and the straight-line path indicated by line  $(r, q)$ . (b) The transition from straight-line path  $W_{i-1}W_i$  to  $W_iW_{i+1}$  using preturn (Beard and McLain 2012; Kumar et al. 2020).



$c = (c_n, c_e, c_d)$ , where  $-c_d$  represents the desired altitude of the orbit, and to maintain altitude, we let  $h_c = -c_d$ . The path-following algorithm for our system was made using the algorithm in Kumar (2014) and Beard and McLain (2010). Using command of the course angle can reduce the cross-track error  $e_{py}$  to zero in the path following. When,  $e_{py} \rightarrow 0$  and,  $\chi_q$  is known, as in Fig. 3, the straight-line path-following issue becomes selecting the value  $\chi_c$ .

### 3. Experimental setup and neural network architecture

A commercially available parafoil of length 2.15 m was procured for the manufacturing of PPUAV. The rigid part of the PPUAV, the gondola, was fabricated in an L shape so that the propeller and payload could be housed. Three models of gondola using three different materials, viz., (i) aluminum, (ii) balsa wood reinforced with carbon fiber rods and Styrofoam dampers, and (iii) corrugated plastic core (Coro) having solid plastic surface were fabricated. Weight reduction, increase in strength, and ease of reparability were achieved with each successive material. Figures 4a–4c depict the three separate gondolas using three different materials.

The PPUAV was propelled using a brushless DC motor mated with a 16 inch  $\times$  12 inch (length  $\times$  pitch) propeller, drawing power from a 2200 mAh lithium polymer four-cell battery. This combination was chosen to achieve an optimum thrust-to-weight ratio while ensuring 30 min of endurance. An electronic speed controller of 50 amperes was used to control the propeller speed of rotation so that power requirements during takeoff could be met. The total weight of the PPUAV with its controls was 2.25 kg with an aluminum gondola (Fig. 4a), and 1.882 kg with corrugated plastic (Fig. 4c). All

the flight data acquired were using corrugated plastic (Fig. 4c) model. The PPUAV is devoid of the horizontal tail plane as in the case of fixed-wing systems, thereby making it have a low stability margin. Such low margins mandate accurately determining the CG position. The CG position and the angle of attack of the parafoil were ascertained using the trial-and-error method, which required several trials. The PPUAV can be launched by a single person, as shown in Fig. 5a. The launch is carried out during calm weather conditions, i.e., absence of wind. The launch alignment of PPUAV is by having a nose-down approach in the longitudinal axis, while the parafoil generates lift, thereby producing a balancing nose-up moment as shown in Fig. 5a. Once the propeller's correct alignment and aerofoil's angle of attack is achieved, takeoff becomes easier.

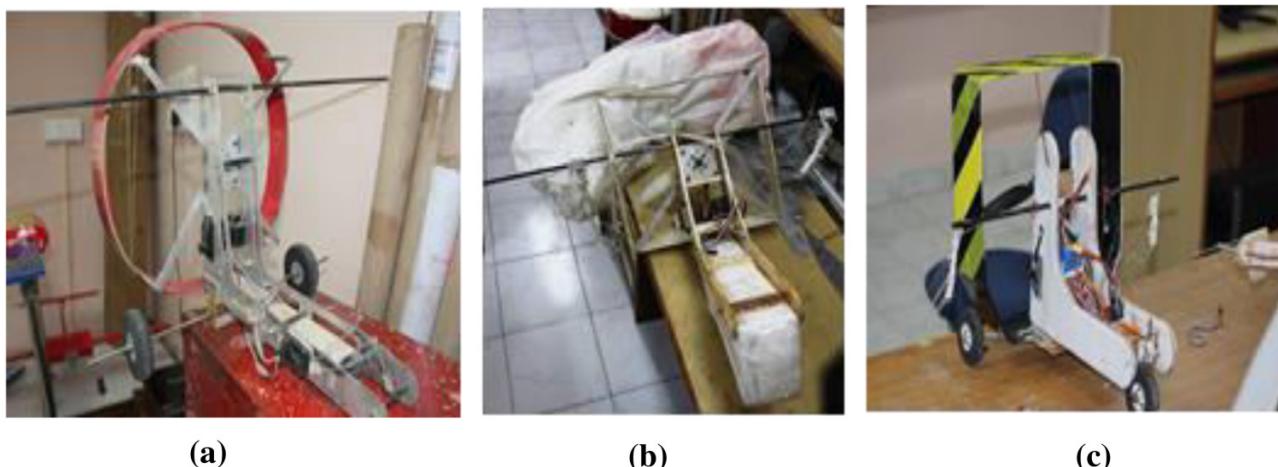
#### 3.1. Mission profile and data collection

A total of 30 sorties were flown performing different maneuvers, and the dataset acquired without any break in transmission or loss of GPS fixation was chosen for training. The chosen dataset had 1425 fields with 23 records in each field. Figure 5b depicts the mission profile showing the longitude and latitude variation.

The flight data of the PPUAV was necessary to train and test the artificial neural network (ANN) control system. Three aspects of flight data, viz., (i) authenticity, (ii) quantum, and (iii) frequency are critical for the successful test of the control system. An inertial measurement unit (IMU) with 9-DoF was integrated into an Arduino controller, with an independent GPS for realizing data acquisitions as shown in Fig. 5c.

Razor make IMU was used for the DAQ system, which has an inertial navigation unit with both angular and linear accelerometers for measuring positional changes and a

**Fig. 4.** Different gondolas using different materials: (a) aluminum, (b) balsawood, and (c) corrugated plastic.



gyroscope for maintaining absolute angular reference. The instantaneous rate of acceleration was measured using accelerometers, whereas changes in rotational attributes like pitch, roll, and yaw were measured using gyroscopes. The orientation drift of PPUAV was accounted for and compensated using a magnetometer. The Arduino processor continually calculated the current position of the PPUAV. To determine velocity, the controller used the measured acceleration with time for each of the 6-DoF, namely  $x$ ,  $y$ ,  $z$ ,  $\theta x$ ,  $\theta y$ , and  $\theta z$ , as well as an estimate of gravity. The computed instantaneous velocity was then time-integrated to determine the current position.

An independent battery powered the DAQ system. The throttle control pulse was recorded as a percentage of the maximum throttle (0%–100%). The port and starboard servo controls were recorded independently from 0 to 200, with 100 as the trim position. These three control parameters were recorded in real time on the PPUAV and were not transmitted to the ground to avoid any phase lag in the transmission system. **Table 2** summarizes the parameters recorded by the two independent recording systems.

In addition to the parameters mentioned in **Table 2**, three control parameters, namely throttle control, port servo control, and starboard servo control, were also recorded along with IMU inputs. Data for several sorties flown were recorded, and the GPS data were verified against the corresponding video to check for any break in transmission or loss of GPS fixations.

### 3.2. Flight data preprocessing

A total of 30 sorties were flown, and the dataset without any break in transmission or loss of GPS fixation, along with maximum maneuvers, was chosen for training. The chosen dataset had 1425 fields with 23 records in each field. The variation of the altitude with respect to time is shown in **Fig. 6a**, and the input variation in throttle control along with acceleration in the X, Y, and Z axes has been shown in **Figs. 6b–6e**. The port and starboard servo control are shown in **Fig. 6f**. Among these parameters, those that exhibited limited or no varia-

tion during each sortie were discarded. For example, Gyro Y and Gyro Z fields had identical values, thereby allowing for discarding the Gyro Y field. Additionally, seven records were identified that did not directly impact the control parameters, thereby allowing for their safe discarding. Such discarding also helps to eliminate the possibility of misguiding the network during training.

The remaining 15 fields, namely roll angle ( $\phi$ ), pitch angle ( $\theta$ ), yaw angle ( $\psi$ ), roll rate ( $p$ ), yaw rate ( $r$ ), acceleration in X, Y, and Z directions ( $a_x(k)$ ,  $a_y(k)$ , and  $a_z(k)$ ), position vector ( $x$ ,  $y$ , and  $z$ ), and velocity vector ( $V$ ) were used for training the ANN. Any incomplete data fields were also discarded and there were very few of them. Spurious entries identified by verifying against each field's maximum and minimum values were also discarded. For example, the speed of the PPUAV should vary between 2 and 55 m/s. However, during the 12 min sortie, one record had the value of 94 and was discarded; else, it would have attenuated the impact of the entire field. Finally, 1300 records having 15 fields each were used for the ANN. Each dataset was normalized in the range of  $[-0.9, 0.9]$ , using **eq. 2**:

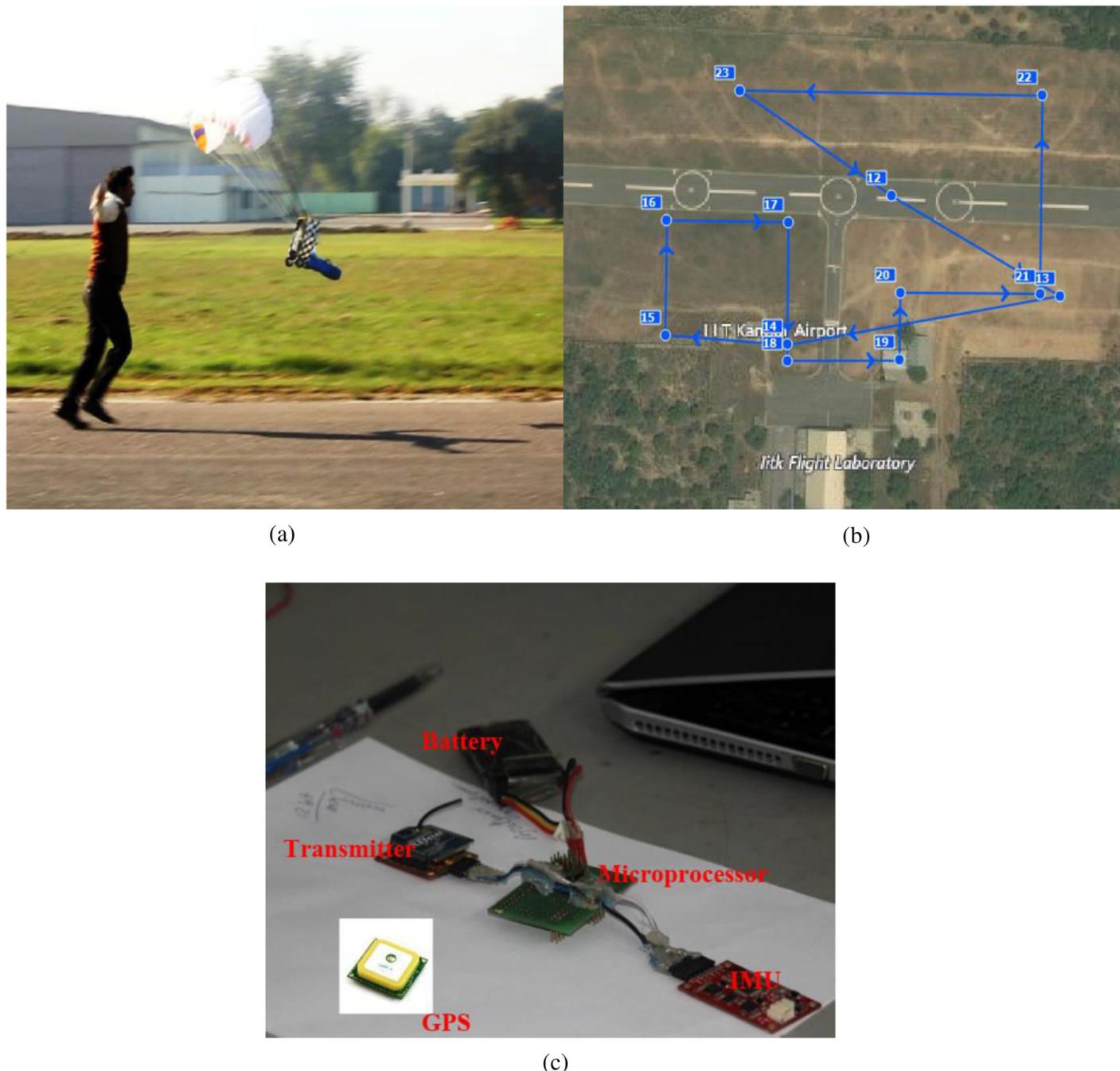
$$(2) \quad Y_n = -0.9 + \{(Y - Y_{\min}) / (Y_{\max} - Y_{\min})\} - 1.8$$

where  $Y_n$  is the normalized value for the variable, and  $Y_{\min}$  and  $Y_{\max}$  are the minimum and maximum of each variable "Y".

### 3.3. Artificial neural network model

This section presents a method in which feedforward artificial neural networks (FFANNs) are used to estimate the control input parameters required to maneuver the paramotor following the dedicated trajectory. It is important to note that the success of FFANN relies heavily on the numerical values assigned to the input variables, and these values are optimized using a batch training method. An initial training set, consisting of input vector  $U(k)$  and output vector  $Z(k)$  as shown in **eqs. 3** and **4**, is used to construct a multinode, mul-

**Fig. 5.** (a) PPUAV being launched in a calm wind condition (Source: Flight Dynamics Lab, IIT Kanpur). (b) Mission profile for the paramotor flight. (c) Data acquisition system with IMU, GPS, controller, and battery.



tilayer neural network capable of managing a large number of parameters and nonlinearity involved:

$$(3) \quad U(k) = [\phi(k) \theta(k) \psi(k) p(k) r(k) a_x(k) a_y(k) \\ \times a_z(k) x(k) y(k) z(k) V(k)]^T$$

$$(4) \quad Z(k) = [\text{PWM}_{\text{port servo}}(k) \text{ PWM}_{\text{throttle}}(k) \text{ PWM}_{\text{starboard}}(k)]$$

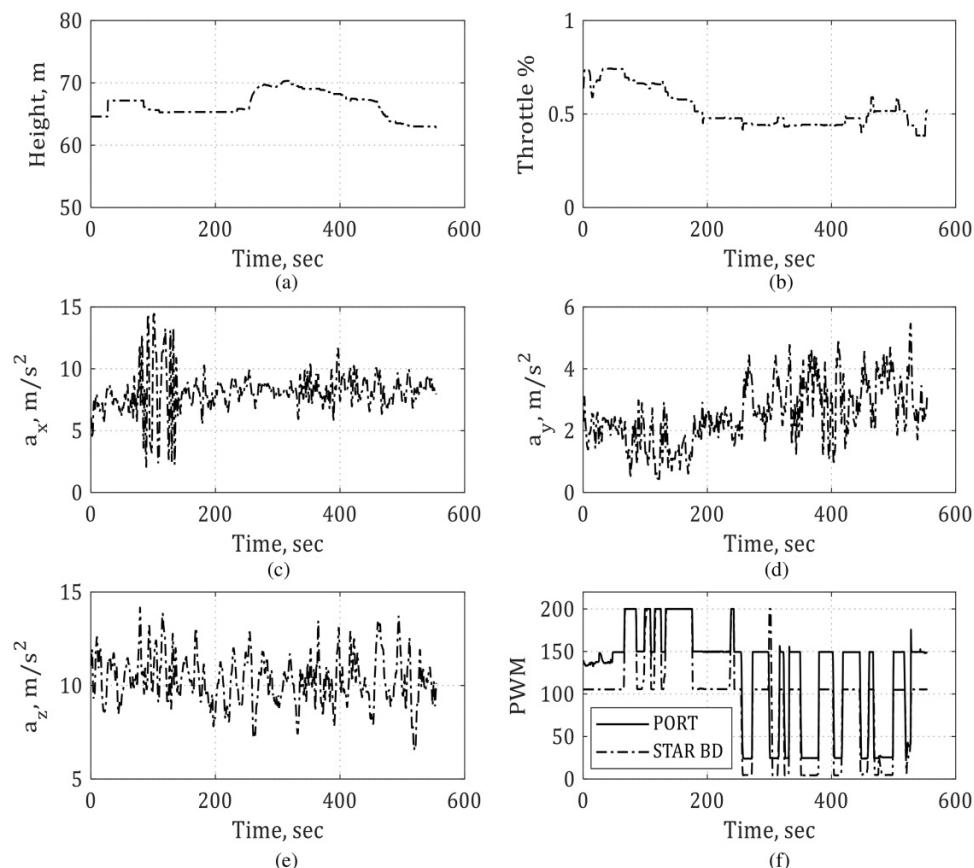
where “ $k$ ” in parenthesis denotes the variable values measured at  $t$ th instant.

Input and output vectors may be built using flight data once it has passed the data compatibility check test and CG translation as depicted in Fig. 7. To guarantee proper training of neural networks, we use the mean squared error (MSE) metric to fine-tune the weights of each neural network module until a good fit is achieved. Once the MSE is below a certain threshold, training on the ANN model of the system can be terminated and it can be used in some limited capacity. This process is referred to as back propagation.

Neural networks are suitable for the approximation of nonlinear time-variant behaviors. Such models are developed

**Table 2.** The flight parameters recorded by the IMU and GPS.

Serial no.	Parameters recorded by IMU	Parameters recorded by GPS
1	System time in 0.5 s interval	System time in 0.5 s interval
2	Roll in degrees, from 0 to 360. It is the instantaneous position of the gondola about the longitudinal axis (set up as the X-axis)	Number of satellites that are visible at the given instant (recorded as "Sat")
3	Pitch in degrees, from 0 to 360, about the lateral axis	"Fix" shows the number of satellites locked upon, akin to the tower strength of a mobile handset. Minimum four satellites are required for extracting the location
4	Yaw in degrees from 0 to 360, about the vertical axis	Latitude (Lat) in degrees
5	Roll rate or the angular velocity along longitudinal axis (Gyro X), measured degrees/s	Longitude (Lon) in degrees
6	Pitch rate or the angular velocity along lateral axis (Gyro Y), measured degrees/s	Height above MSL (Alt) in meter
7	Yaw rate or the angular velocity along vertical axis (Gyro Z), measured degrees/s	Scalar speed of the vehicle (Speed 2D)
8	Acceleration along the longitudinal axis (Acc X), measured m/s <sup>2</sup>	Velocity in the North direction (Vel N) in m/s
9	Acceleration along the lateral axis (Acc Y), measured m/s <sup>2</sup>	Velocity in the East direction (Vel E) in m/s
10	Acceleration along the vertical axis (Acc Z), measured m/s <sup>2</sup>	Velocity in the downward direction (Vel D) in m/s

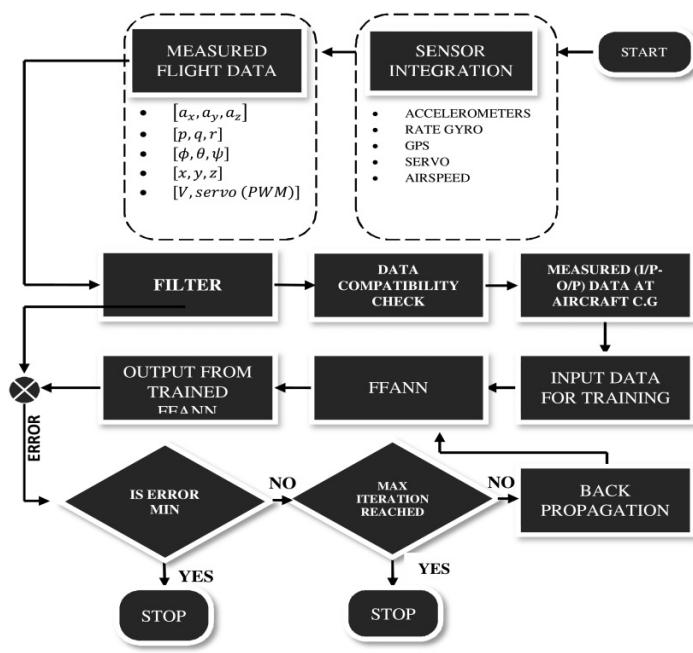
**Fig. 6.** Time histories of parameter variation: (a) altitude, (b) throttle %, (c) acceleration along x-axis, (d) acceleration along y-axis, (e) acceleration along z-axis, and (f) pulse width modulation (PWM) for port and starboard servos.

from original data to establish relationships between various system parameters while overcoming kinetic assumptions. Due to the inherent learning capabilities associated with neural network models, they are proven to help control complex systems. Also, such models have the disadvantage of not being able to provide an insight into the physical process in-

volved. Moreover, the predictions made by such models are system specific and valid only for the range to accommodate changes in process or variations in operating range.

The instantaneous error value is monitored using the graph of each output parameter's measured and calculated values. Specific steps in training the ANN are as follows:

**Fig. 7.** Flowchart data collection and neural network training.



1. Select a training record from the training dataset and apply it to network input nodes.
2. Calculate the output error of the network.
3. Quantify the error between the output of the network to the desired output.
4. Adjust network weights so that error is minimized.
5. Repeat above steps for the training dataset until the error for the entire set is less than the decided error tolerance limit.

The sigmoidal function utilized as the activation function is given in eq. 5. An activation function feeds the output of the previous layer's weighted sum of inputs into the next layer. When a neuron's activation function is sigmoid, the unit's output is guaranteed to always fluctuate between 0 and 1. Since the sigmoid is a nonlinear function, the output of this system would also be a nonlinear function of the inputs' weights. In eq. 5,  $Y_i$  denotes the output of the neuron,  $\lambda$  denotes the logistic gain,  $X_i$  denotes the internal activity of neuron, and index  $i$  denotes the address of the neuron. This specific function was chosen for its popularity among nonlinear responses, i.e., sigmoidal function (Tepedelenlioglu and Scalero 1992; Gauche et al. 1999).

$$(5) \quad Y_i = \frac{1}{1 + e^{-X_i/\lambda}}$$

The method of steepest gradient descent was used to calculate local gradient of the network ( $\Delta$ ) as given in eq. 6. This is popularly known as the delta rule by proceeding backward, layer by layer (Tepedelenlioglu and Scalero 1992; Hess 2013):

$$(6) \quad \Delta_j^{(k)}(n) = e_j^{(k)}(n) O_j(n) [1 - O_j(n)]$$

where,  $e_j^{(k)}(n)$  is the network error and  $O_j$  is the output of the network in nth iteration. The main objective of developing

the PPUAV was to create an ANN-based control system that could autonomously fly the PPUAV. To acquire data, an experienced pilot flew the PPUAV manually via visual flight. These data were utilized to train the ANN to predict the control input given a specified outcome. In a nutshell, stating GPS coordinates of the waypoints, the PPUAV would be allowed to fly on its own.

### 3.3.1. Artificial neural network architecture

As mentioned earlier, out of the 1300 preprocessed data, 1000 were utilized for training, 200 for testing, and the final 100 for ANN validation. From the data analysis, 12 parameters were taken as input parameters given by the input vector  $U(k)$ . As can be seen in Fig. 8, the vector  $Z(k)$  represents the three output parameters.

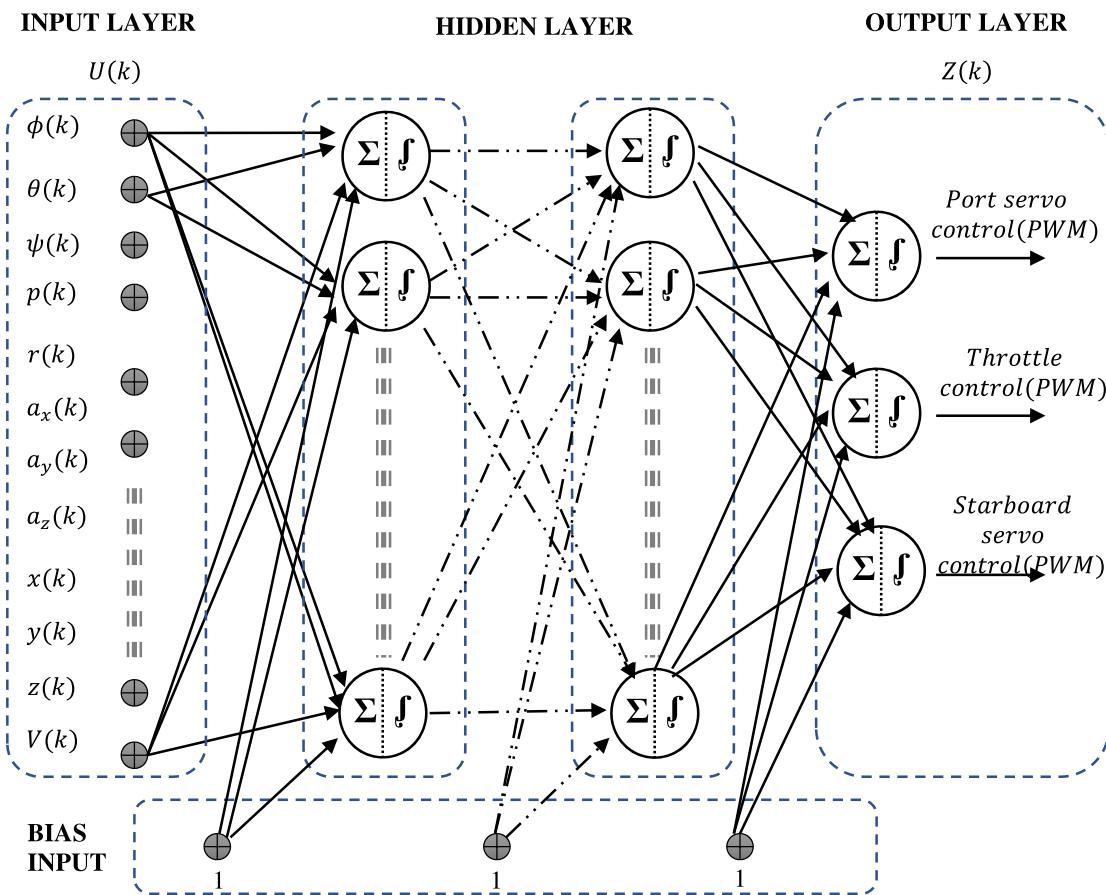
### 3.3.2. ANN training algorithm

Typically, the performance function used to train feedforward neural networks is chosen to be the sum of squares of the network errors on the training set:

$$F_{(w)} = \text{MSE} = \frac{1}{n} \sum_{i=1}^n (e_i)^2 = \frac{1}{n} \sum_{i=1}^n (t_i - o_i)^2$$

where  $o_i$ ,  $t_i$ ,  $e_i$ ,  $n$ , and  $w$  are the network output, the intended target, the network error, number of training samples, and the vector of the network weights, respectively (Kumar et al. 2022a, 2022b). The training process of a neural network involves optimizing the performance function via updating the network weights and biases at every iteration. This can be accomplished using a variety of training algorithms. The primary goal is to stop gradients from bursting or disappearing at the layer activation outputs during forward propagation. For the network to converge, the loss gradients must be just right, and if each of these issues arises, the network will take longer to converge, if it converges at all. If the weights are properly initialized, we may use gradient descent to converge to a minimum and accomplish our goal of optimizing the loss function in a reasonable amount of time. It is typically employed to create asymmetry. Weights should ideally be assigned random values other than zero.

The most critical aspect of ANN modeling is the design of the architecture. An appropriate architecture with infinite training can never generate fruitful results. Identifying the optimum rate of learning is a time-consuming process. The rate at which the model learns is adjusted with a tuning parameter known as the learning rate. However, the model may overshoot the minimum value if the learning rate is too high since the weights are updated too quickly. The model may take its time and improve its performance with a modest learning rate. Minor adjustments to the weight are made with each revision, so it converges too slowly. A slow learning rate might cause a system to converge on an undesired local minimum. Thus, we should avoid using extremes of the learning rate (both high and low). The learning rate's value depends on the structure of your neural network and the size of your training data collection. A big number will lead to rapid learn-

**Fig. 8.** ANN architecture with (N) hidden layer.

ing, but there is a risk of overshooting the local minimum of the output function. Hence, values between 0.05 and 0.95 are typical values used for training. When we are unsure about the optimal default learning rate, it is a decent approximation to start with the typical default value for the learning rate, i.e., 0.1.

All major combinations of architectures comprising one, two, three, and four hidden layers were created and trained with dataset of 1000 fields with a 0.1 learning rate parameter and zero momentum factor. The momentum factor is increased from zero only when the network fails to learn because of unusual conditions in the training set that has to be smoothed. Sigmoidal function is selected because of its differentiability resulting in good learning property. The error tolerance in the program was kept high at 10%, to avoid the training getting stuck at a local minimum; however, the actual error value, which was always less than 2%, was constantly monitored and recorded. Each architecture was subjected to 100 iterations of training after which the worst actual error out of the three output functions (port servo, starboard servo, and throttle control) was recorded for comparison. The complete list of assumed ANN designing parameters that are used for training is listed in **Table 3**.

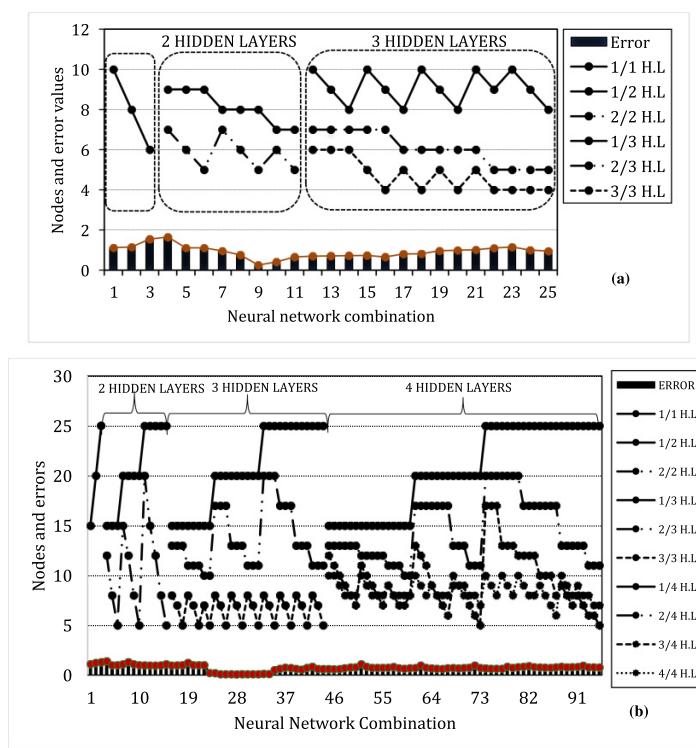
**Table 3.** The ANN designing parameter values assumed for the training and testing of the architecture.

Serial no.	Parameter	Value
1	Learning rate	0.1
2	Momentum factor	0
3	Transfer function	Tan-Sigmoidal
4	Error tolerance	10%
5	No. of iteration	100
6	No. of inputs	12
7	No. of outputs	3
8	No. of hidden layers	1–4

### 3.3.3. ANN number of layers and nodes selection

All possible combinations of converging ANN architecture were trained for 100 iterations and the minimum value of the error was recorded, for various hidden layers. For the fixed 12 parameters in the input layer and 3 parameters in the output layer, the number of nodes in the first hidden layer was varied from 10 to 8 to 6 so as to keep the basic convergence nature of the architecture intact, plotted at serial numbers 1–3 in X-axis

**Fig. 9.** Nodes in hidden layers (H.L.) versus estimated error values after training. (a) Three hidden layers. (b) Four hidden layers.



of Fig. 9a. With two hidden layers, the number of nodes in the first hidden layer was varied from seven to nine, and in the second hidden layer it was varied from five to seven making a total of eight combinations, plotted at serial numbers 4–11 in X-axis of Fig. 9a. Thereafter, when using three hidden layers, the number of nodes in the first hidden layer was varied from 8 to 10, in the second hidden layer it was varied from 6 to 7, and in the third hidden layer from 4 to 6, making a total of 14 combinations plotted at serial numbers 12–25 in X-axis of Fig. 9a. The error value obtained after training is plotted as an error bar. As shown in Fig. 9, all the possible combinations were tried to select the number of nodes and layers.

The error was minimal for 12-8-5-3 architecture, as shown in serial number 9 in Fig. 9a. The correlation between the measured and the calculated value from ANN is shown in Fig. 10a. The trained network was tested with the 200 records kept aside for the same, and the result is shown in Fig. 10b. However, the result was not found to be satisfactory, as at some points the calculated output varied from the measured output as evident from the graphs below.

Despite good training results, the network failed when tested with new data, which indicates the number of nodes in the hidden layers was insufficient to generate the correct relationship between the input and the output parameters. The dataset of 1000 fields appeared to be sufficient for training. A conventional converging architecture was found to be inadequate to be trained to predict good results.

The authors decided to explore the diverging network with one, two, three, and four hidden layers with the same dataset

of 1000 fields for training with 100 iterations, having fixed 12 input and 3 output parameters. In a nutshell, the input nodes were diverged from 12 to 16, 20, and 24 nodes. It was then converged in all the possible ways to reach the output parameters of three nodes in the number of combinations as given in Table 4.

As the number of hidden layers went up, it was clear that the number of possible combinations went up exponentially. Table 4 shows that the input layer of 12 parameters was split into 16, 20, and 24 nodes in one hidden layer, giving three architecture combinations that were plotted against the serial numbers 1–3. Similarly, serial numbers 4–15 are for two hidden layers, serial numbers 16–44 are for three hidden layers, and the rest are for four hidden layers. Figure 9b shows the error values obtained after 100 iterations.

Figure 9b shows that the least error value was found for serial number 28, corresponding to the three hidden layers network design matching the 12-20-14-7-3 network. The number of hidden layers was not increased beyond four because the error value showed no reducing trend, indicating that the network was getting overtrained.

The 12-20-14-7-3 network architecture had the minimum error, as indicated in the plots obtained in Fig. 9b. Figure 10c shows the relationship between the measured and the calculated values for the throttle control parameter. The ANN architecture was now trained extensively for several iterations, gradually reducing the error tolerance value until a further reduction in error was not feasible, i.e., the program went into an infinite loop. The final results for the port servo, starboard servo, and throttle control parameters show reasonably close prediction, as depicted in Figs. 10d–10f.

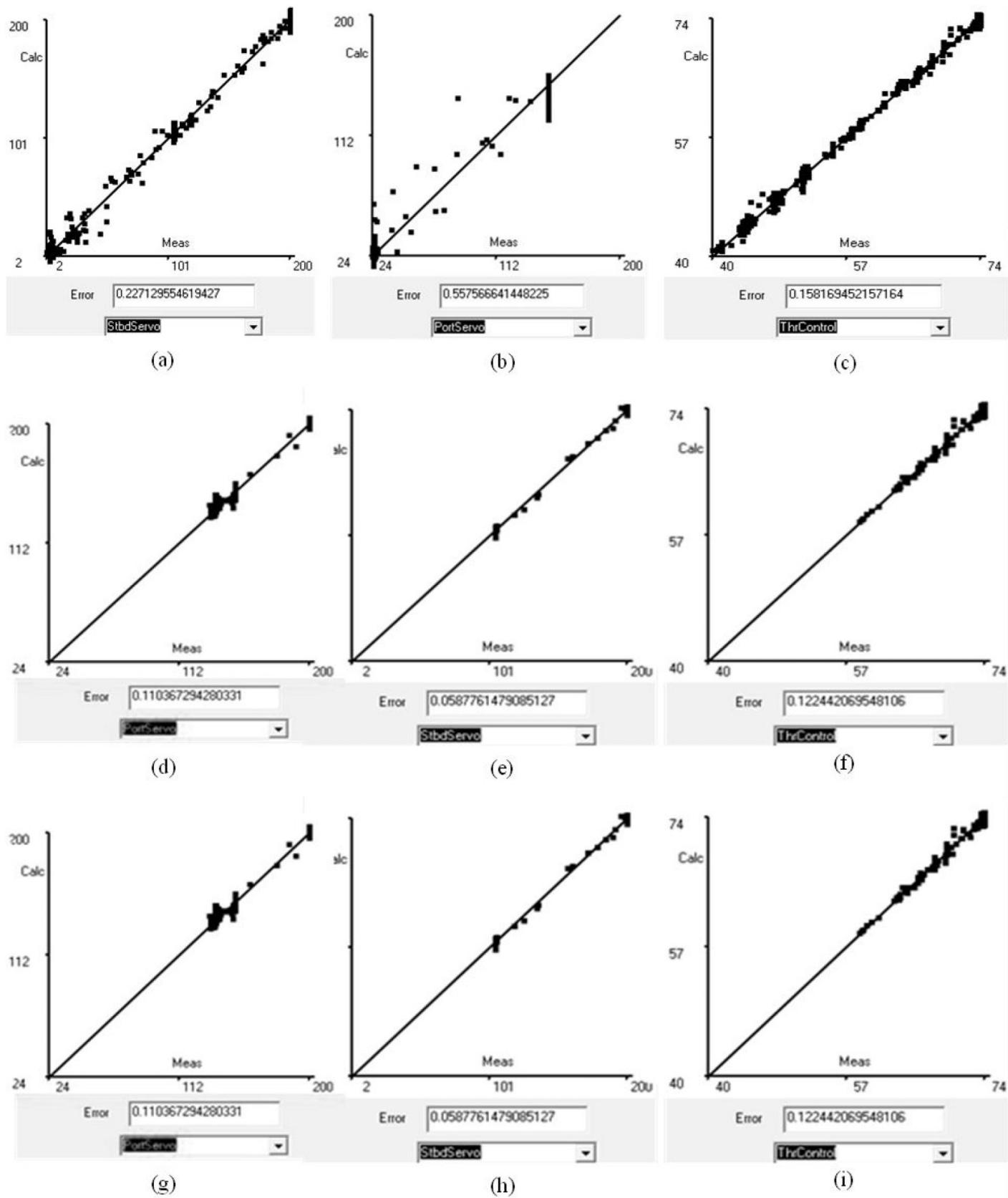
The trained weights were saved and subsequently tested with the dataset of 200 fields kept aside for testing. The trained network is now tested with the new set of data, to which it has never been exposed before. The results are shown in Figs. 10g–10i, which clearly shows that although the quantum of data is significantly less (200 compared with 1000 earlier), there is an acceptable correlation.

## 4. Results and discussion

Finally, the 100 records preserved for testing were individually charged, and the predicted values recorded. The graph of the actual and predicted values is shown in Fig. 11. The lines are almost overlapping and the calculated correlation between the predicted and measured values for port servo control is 0.999883, for starboard servo control is 0.999744, and for throttle control is 0.982832.

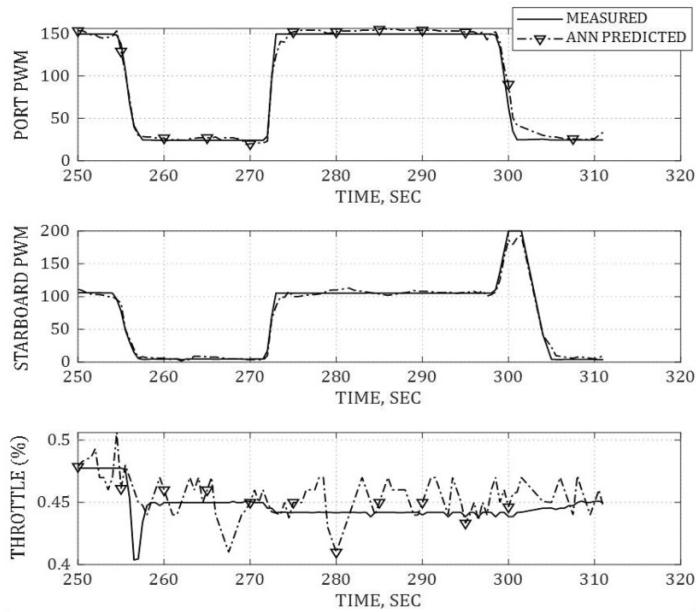
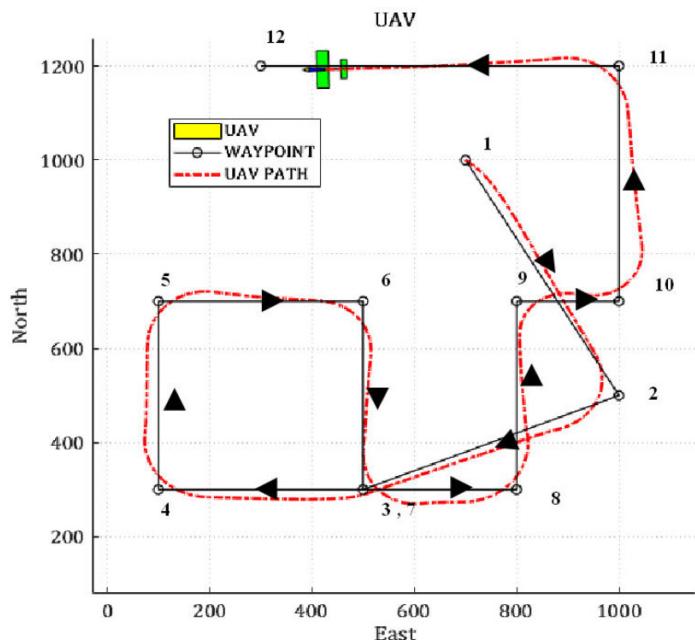
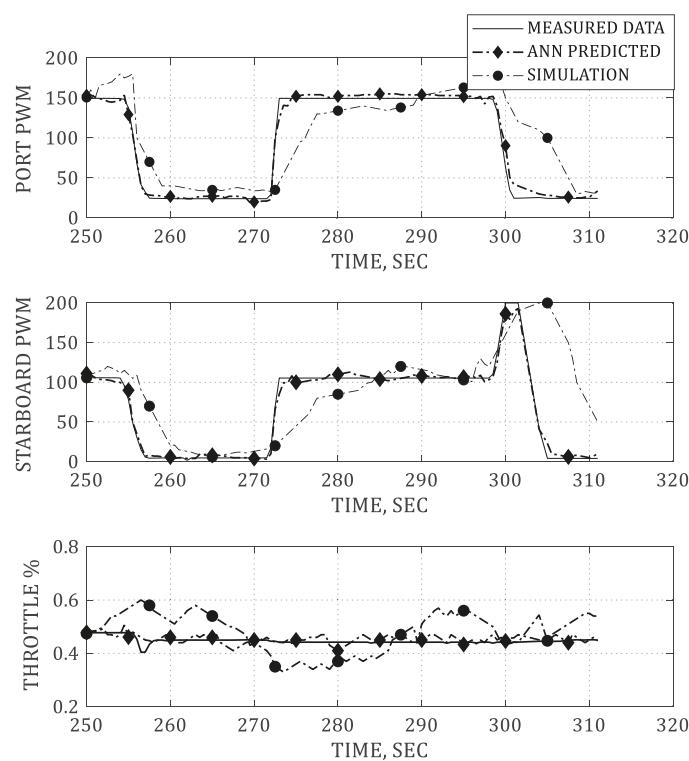
The powered paraglider was simulated for a similar flight condition, and the mission plan was chosen to be similar to the actual flight path flown, as shown in Fig. 12. The mission consists of waypoints in the following sequence 1 (start point) → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 (end point). Control architecture as mentioned in the previous section (Fig. 2) is employed, and the trajectory followed by the UAV is shown in red.

Figure 13 shows the response of the actual actuator control, ANN-predicted controls, and the simulated control response

**Fig. 10.** Results of training and testing the artificial neural network for two hidden layers.

**Table 4.** Hidden layers and the number of combinations trained on the designed ANN architecture.

Combination variables	Combination results
No. of hidden layers used	1    2    3    4
No. of combinations of ANN architectures achieved	3    12    29    51

**Fig. 11.** Actual versus estimated values of control parameters for the tested ANN model for port servo, starboard servo, and throttle control.**Fig. 12.** PPUAV simulated flight path for the control logic.**Fig. 13.** The graph of the actual, ANN estimation, and simulation values of port servo, starboard servo, and throttle control parameters.**Table 5.** Correlation between the ANN estimated and the simulation estimated values of the three control parameters.

Control parameter	ANN prediction	Simulation prediction
Actual port servo	0.9962	0.8219
Actual starboard servo	0.9963	0.5365
Actual throttle control	0.8203	0.4906

of the port servo, starboard servo, and throttle control for a similar trajectory maneuver as depicted in Fig. 12.

In Table 5, the correlation between the ANN and simulated prediction is presented. As can be seen, ANN-predicted response is accurate and matches with the actual response, whereas the predicted response of the simulation somewhat lags in accuracy. This can be due to the simplified 6-DoF dynamics employed to design the control of PPUAV. This is adequate to confidently use this trained ANN architecture to predict the trajectory of a PPUAV. The trained ANN can be used online with the 9 (12 – 3) parameters as direct online inputs and the desired values of the trajectory as predetermined latitude, longitude, and altitude inputs to generate the controls for the parafoil propelled vehicle (PPV) in autonomous mode.

## 5. Conclusion

This paper attempts to estimate the throttle and servo controls of PPUAV using neural networks. The main contribution of the present work is the fabrication of a small-scale PPV that

can be flown confidently in calm wind conditions. When it comes to the instrumentation part, putting sensors for data acquisition is challenging as the canopy is vulnerable to deformation if sensors are put on it. Hence, the sensors were put in the gondola. The acquired data were then processed using data compatibility test before applying to neural networks. Thereafter, it required collecting sufficient flight data and development of neural models using back propagation network, which can be used to predict the controls. It was found that a complicated system like PPUAV with 12 independently varying input parameters can best be controlled with an ANN model with a divergent-convergent design. Several combinations were tested with the gathered flight data, and it was found that the minimum MSE was obtained for three hidden layers with 12-20-14-7-3 architecture. A sophisticated 6-DoF nonlinear mathematical model was created by making certain assumptions. By reducing the nonlinear 6-DoF model to a low-fidelity model, we were able to derive a set of linear equations that describe the lateral dynamics of the paramotor and are therefore useful for controlling it. Parallel efforts resulted in the creation of more basic longitudinal dynamic models. The highly applicable feedback approach has been used to increase stability by successively closing the loop. Finally, simulation results showed that lateral and longitudinal paramotor control may be successfully separated by employing asymmetric brake deflection for lateral control while keeping altitude and speed control via throttle and pitch control. The control system's response was tested for similar flight conditions and same flight path. The result of the DNN and the simulation was compared with the actual control signal required. The pattern and nature of the plots suggest the efficacy of the neural network's divergent-convergent type design in estimating the throttle and servo controls of a small-scale powered paraglider. Before autonomous powered paragliders can be widely used, they must overcome a few specific hurdles. A more stable airframe with an easier-to-implement control design might be achieved with more investigation into control techniques. Using a simple PID-based control and neural network-based dynamic modeling, this article demonstrated the feasibility of incorporating these cutting-edge adaptive control technologies into future endeavors through experimentation and simulation. Using a neural network's feedback loop, complex, nonlinear relationship dynamics can be accurately captured. Improvements in flying performance relative to simulation testing should be possible as new experimental data are put into a neural network controller. In the future, the PPUAV can be scaled up, and used as an autonomous vehicle to deliver desired payloads, vide a predetermined trajectory to an actual location.

## Article information

### History dates

Received: 13 August 2022

Accepted: 13 April 2023

Accepted manuscript online: 4 May 2023

Version of record online: 19 May 2023

### Copyright

© 2023 The Author(s). This work is licensed under a [Creative Commons Attribution 4.0 International License](#) (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

### Data availability

Data generated or analyzed during this study are not available due to the nature of this research (as they are proprietary or confidential in nature and may only be provided with restrictions).

## Author information

### Author ORCIDs

Prashant Kumar <https://orcid.org/0000-0003-2208-7891>

### Author contributions

Conceptualization: PK, AKG

Formal analysis: AKG

Investigation: PK, JR

Methodology: PK, DP, AKG

Resources: DP

Supervision: AS, JR, DP

Validation: AS

Writing – original draft: PK, BC

Writing – review & editing: PK, BC, AS, JR, AKG

### Competing interests

The authors declare that there are no competing interests.

### Funding information

The authors declare no specific funding for this work.

## References

- Beard, R.W., and McLain, T.W. 2010. Navigation, guidance, and control of small and miniature air vehicles. Brigham Young University.
- Beard, R.W., and McLain, T.W. 2012. Small unmanned aircraft: theory and practice. Princeton University Press, Princeton, NJ.
- Ben Mosbah, A., Botez, R.M., and Dao, T.M. 2016. A hybrid original approach for prediction of the aerodynamic coefficients of an ATR-42 scaled wing model. *Chin. J. Aeronaut.* **29**(1): 41–52. doi:[10.1016/j.cja.2015.12.022](https://doi.org/10.1016/j.cja.2015.12.022).
- Breivik, M. 2005. Principles of guidance-based path following in 2D and 3D. In Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain, 12–15 December 2005. pp. 627–634.
- Egorchev, M.V., and Tiumentsev, Y.V. 2018. Neural network identification of aircraft nonlinear aerodynamic characteristics. In IOP Conference Series Materials Science and Engineering. **312**(1). doi:[10.1088/1757-899X/312/1/012008](https://doi.org/10.1088/1757-899X/312/1/012008).
- Gang, Y. 2015. Nine-degree of freedom modeling and flight dynamic analysis of parafoil aerial delivery system. *Procedia Eng.* **99**: 866–872. doi:[10.1016/j.proeng.2014.12.614](https://doi.org/10.1016/j.proeng.2014.12.614).
- Gauche, E., Coelho, J., and Teive, R.C.G. 1999. Mixed back-propagation/Marquardt-Levenberg algorithm for optimizing the distribution electrical systems operation. In Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference. **2**. pp. 753–758. doi:[10.1109/tdc.1999.756144](https://doi.org/10.1109/tdc.1999.756144).

- Goodrick, T., Murphy, J.R., and Pearson, A. 1973. Analysis of various automatic homing techniques for gliding airdrop systems with comparative performance in adverse winds. 73. doi:[10.2514/6.1973-462](https://doi.org/10.2514/6.1973-462).
- Harth, E. 1983. Order and chaos in neural systems: an approach to the dynamics of higher brain functions. In *IEEE Transactions on Systems, Man and Cybernetics*. SMC-13(5). pp. 782–789. doi:[10.1109/TSMC.1983.6313072](https://doi.org/10.1109/TSMC.1983.6313072).
- Hess, R. 2013. On the use of back propagation with feed-forward neural networks for the aerodynamic estimation problem. doi:[10.2514/6.1993-3638](https://doi.org/10.2514/6.1993-3638).
- Hua, Y., Lei, S., Cheng, L., and Jun, H. 2013. Study on powered-parafoil longitudinal flight performance with a fast estimation model. *J. Aircr.* 50(5): 1660–1668. doi:[10.2514/1.C032176](https://doi.org/10.2514/1.C032176).
- Iacomini, C.S., and Cerimele, C.J. 1999. Longitudinal aerodynamics from a large scale parafoil test program. In *15th Aerodynamic Decelerator Systems Technology Conference*. pp. 229–239. doi:[10.2514/6.1999-1732](https://doi.org/10.2514/6.1999-1732).
- Ignat'yev, D.I., and Khrabrov, A.N. 2015. Neural network modeling of unsteady aerodynamic characteristics at high angles of attack. *Aerospace Sci. Technol.* 41: 106–115. doi:[10.1016/j.ast.2014.12.017](https://doi.org/10.1016/j.ast.2014.12.017).
- Jaiswal, R., Prakash, O., and Chaturvedi, S.K. 2020. A preliminary study of parameter estimation for fixed wing aircraft and high durability parafoil aerial vehicle. *12(4)*: 95–109. doi:[10.13111/2066-8201.2020.124.9](https://doi.org/10.13111/2066-8201.2020.124.9).
- Kumar, P. 2014. Design and development of autonomous light weight powered paraglider. Indian Institute of Technology.
- Kumar, P., Sonkar, S., Ghosh, A.K., and Philip, D. 2020. Dynamic waypoint navigation and control of light weight powered paraglider. In *IEEE Aerospace Conference Proceedings*. p. 208016. doi:[10.1109/AERO47225.2020.9172564](https://doi.org/10.1109/AERO47225.2020.9172564).
- Kumar, P., Sonkar, S., George, R.C., Ghosh, A.K., and Philip, D. 2022a. Longitudinal aerodynamic parameters estimation using machine learning with neuro artificial bee colony fusion algorithm (NABC). In *2022 13th Asian Control Conference (ASCC)*. pp. 500–505. doi:[10.23919/ASCC56756.2022.9828108](https://doi.org/10.23919/ASCC56756.2022.9828108).
- Kumar, P., Sonkar, S., Ghosh, A.K., and Philip, D. 2022b. Lateral aerodynamic parameters estimation using neuro artificial bee colony fusion algorithm (NABC). In *2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR)*. pp. 1–6. doi:[10.1109/ICAITPR51569.2022.9844223](https://doi.org/10.1109/ICAITPR51569.2022.9844223).
- Li, B., He, Y., Han, J., and Xiao, J. 2019. A new modeling scheme for powered parafoil unmanned aerial vehicle platforms: theory and experiments. *Chin. J. Aeronaut.* 32(11): 2466–2479. doi:[10.1016/j.cja.2019.04.001](https://doi.org/10.1016/j.cja.2019.04.001).
- Luo, S., Sun, Q., Wu, W., Sun, M., and Chen, Z. 2019. Accurate flight path tracking control for powered parafoil aerial vehicle using ADRC-based wind feedforward compensation. *Aerospace Sci. Technol.* 84: 904–915. doi:[10.1016/j.ast.2018.11.022](https://doi.org/10.1016/j.ast.2018.11.022).
- Mooij, E., Wijnands, Q.G.J., and Schat, B. 2003. 9 Dof parafoil/payload simulator development and validation. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 1–12 August. doi:[10.2514/6.2003-5459](https://doi.org/10.2514/6.2003-5459).
- Murray, J.E., Sim, A.G., Neufeld, D.C., Rennich, P.K., Norris, S.R., and Hughesb, W.S. 2013. Further development and flight test of an autonomous precision landing system using a parafoil. doi:[10.2514/6.1994-2141](https://doi.org/10.2514/6.1994-2141).
- Nicolaides, J.D. 1970. A review of para-foil applications. Patent 3285546.
- Ortega, G., Mulder, A.~J., and Verbruggen, H. 1999. Fuzzy logic for space-craft control: a European approach. In *Artificial intelligence, robotics and automation in space*. Edited by M. Perry. Vol. 440. p. 471.
- Phillips, C., Karr, C.L., and Walker, G. 1996. Helicopter flight control with fuzzy logic and genetic algorithms. *Eng. Appl. Artif. Intell.* 9(2): 175–184. doi:[10.1016/0952-1976\(95\)00008-9](https://doi.org/10.1016/0952-1976(95)00008-9).
- Prakash, O., and Ananthkrishnan, N. 2006. Modeling and simulation of 9-DOF parafoil-payload system flight dynamics. In *Collection of Technical Papers – 2006 Atmospheric Flight Mechanics Conference*. 1(August). pp. 140–165. doi:[10.2514/6.2006-6130](https://doi.org/10.2514/6.2006-6130).
- Qian, K., and Chen, Z. 2010. Dynamic inversion based on neural network applied to nonlinear flight control system. In *Proceedings of the 2010 2nd International Conference on Future Computer and Communication, ICFCC 2010*. 1(4). pp. V1–699-V1-703. doi:[10.1109/ICFCC.2010.5497341](https://doi.org/10.1109/ICFCC.2010.5497341).
- Slegers, N.J. 2010. Effects of canopy-payload relative motion on control of autonomous parafoils. 33(1). doi:[10.2514/1.44564](https://doi.org/10.2514/1.44564).
- Smith, J., Bennett, T., and Fox, R. 1999. Development of the NASA x-38 parafoil landing system. In *15th Aerodynamic Decelerator Systems Technology Conference*. pp. 205–217. doi:[10.2514/6.1999-1730](https://doi.org/10.2514/6.1999-1730).
- Talwar, A., Lokhande, G., Jain, R., and Singh, S. 2017. Estimation of aerodynamic parameters using Cascade Forward Back Propagation. In *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, 1–6 January, 2018. doi:[10.1109/TEL-NET.2017.8343508](https://doi.org/10.1109/TEL-NET.2017.8343508).
- Tan, P., Sun, M., Sun, Q., and Chen, Z. 2020. Dynamic modeling and experimental verification of powered parafoil with two suspending points. *IEEE Access*, 8: 12955–12966. doi:[10.1109/ACCESS.2020.2965541](https://doi.org/10.1109/ACCESS.2020.2965541).
- Tao, J.I.N., Dehmer, M., and Xie, G. 2019. A generalized predictive control-based path following method for parafoil systems in wind environments. *IEEE Access*, 7: 42586–42595. doi:[10.1109/ACCESS.2019.2905632](https://doi.org/10.1109/ACCESS.2019.2905632).
- Tepedelenlioglu, N., and Scalero, S.R. 1992. A fast new algorithm for training feedforward neural networks. *IEEE Trans. Signal Process.* 40(1): 202–210. doi:[10.1109/78.157194](https://doi.org/10.1109/78.157194).
- Tsai, C.C., Huang, H.C., and Lin, S.C. 2010. Adaptive neural network control of a self-balancing two-wheeled scooter. *IEEE Trans. Ind. Electron.* 57(4): 1420–1428. doi:[10.1109/TIE.2009.2039452](https://doi.org/10.1109/TIE.2009.2039452).
- Umenberger, J., and Göktoğan, A.H. 2012. Guidance, navigation and control of a small-scale paramotor. In *Australasian Conference on Robotics and Automation, ACRA*. pp. 3–5.
- Wailes, W., and Harrington, N. 2013. The guided parafoil airborne delivery system program. doi:[10.2514/6.1995-1538](https://doi.org/10.2514/6.1995-1538).
- Wang, T., Gao, H., and Qiu, J. 2016. A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. *IEEE Trans. Neural Netw. Learn. Syst.* 27(2): 416–425. doi:[10.1109/TNNLS.2015.2411671](https://doi.org/10.1109/TNNLS.2015.2411671). PMID: 25898246.
- Wise, K.A. 2006. Dynamics of a UAV with parafoil under powered flight. In *Collection of Technical Papers – AIAA Guidance, Navigation, and Control Conference 2006*. 8(August). pp. 5376–5398. doi:[10.2514/6.2006-6795](https://doi.org/10.2514/6.2006-6795).
- Wu, S.-F., Engelen, C.J.H., Chu, Q.-P., Babuška, R., Mulder, J.A., and Ortega, G. 2001. Fuzzy logic based attitude control of the spacecraft X-38 along a nominal re-entry trajectory. *Control Eng. Pract.* 9(7): 699–707. doi:[10.1016/S0967-0661\(01\)00036-3](https://doi.org/10.1016/S0967-0661(01)00036-3).
- Yakimenko, O.A. 2005. On the development of a scalable 8-DoF model for a generic parafoil-payload delivery system. In *Collection of Technical Papers – 18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*. pp. 642–654. doi:[10.2514/6.2005-1665](https://doi.org/10.2514/6.2005-1665).
- Zhu, E., and Gao, H. 2020. Guidance-based path following control of the powered parafoil. *Control Eng. Appl. Informatics*, 22(1): 42–50.
- Zhu, E., Sun, Q., Tan, P., Chen, Z., Kang, X., and He, Y. 2015. Modeling of powered parafoil based on Kirchhoff motion equation. 617–629. doi:[10.1007/s11071-014-1690-9](https://doi.org/10.1007/s11071-014-1690-9).
- Zou, K., Zhang, Q., and Zhang, Z. 2018. Parafoil path tracking control based on GPC parameters self-correction with interval type-2 fuzzy logic. In *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*. pp. 1144–1149. doi:[10.1109/CCDC.2018.8407301](https://doi.org/10.1109/CCDC.2018.8407301).