ISSN 1648-7788 / eISSN 1822-4180

AVIATION

2023
Volume 27
Issue 2
Pages 75-85
https://doi.org/10.3846/aviation.2023.18909

FLIGHT PHASE CLASSIFICATION FOR SMALL UNMANNED AERIAL VEHICLES

Jakub LEŠKO [1✉], Rudolf ANDOGA [1], Róbert BRÉDA [1], Miriam HLINKOVÁ [1], Ladislav FŐZŐ [2]

[1]Department of Avionics, Faculty of Aeronautics, Technical University of Košice, Košice, Slovakia
[2]Department of Aviation Engineering, Faculty of Aeronautics, Technical University of Košice, Košice, Slovakia

Article History:
received 20 May 2022
accepted 12 October 2022

Abstract. This article describes research on the classification of flight phases using a fuzzy inference system and an artificial neural network. The aim of the research was to identify a small set of input parameters that would ensure correct flight phase classification using a simple classifier, meaning a neural network with a low number of neurons and a fuzzy inference system with a small rule base. This was done to ensure that the created classifier could be implemented in control units with limited computational power in small affordable UAVs. The functionality of the designed system was validated by several experimental flights using a small fixed-wing UAV. To evaluate the validity of the proposed system, a set of special maneuvers was performed during test flights. It was found that even a simple feedforward artificial neural network could classify basic flight phases with very high accuracy and a limited set of three input parameters.

Keywords: flight phase, classification, fuzzy logic, artificial neural network, unmanned aerial vehicles.

✉Corresponding author. E-mail: *jakub.lesko@tuke.sk*

Introduction

The loss of control of an unmanned aerial vehicle can be caused by unfavorable wind shear, incorrect configuration of the take-off flaps, icing on the leading edge of the wings, faults in the control system, or incorrect input from the pilot (Gill et al., 2013). Loss of control can be described as motion that is outside of the normal operating flight envelope, due to the UAV's inability to maintain altitude or heading, unpredictably altered by pilot control inputs, likely to result in high angular rates and displacements, etc. (Wilborn & Foster, 2004). Fast and automatic compensation that can detect abnormal situations can dramatically increase the safety of UAV operation (Perhinschi et al., 2010).

Flight phase classification is often associated with flight envelope protection and flight control systems. It is generally a problem that needs to be solved in the multidimensional space of flight parameters. Ideal candidates for solving problems of flight phase classification and flight envelope setting are artificial neural networks (ANNs), which have been successfully used to calculate the flight envelope and determine whether the aircraft is within or outside it (Basheer & Hajmeer, 2000). This classification methodology can be implemented in a real-time environment, as shown in the work of Norouzi et al. (2019).

On the other hand, Tang et al. (2009) used linear extrapolation to update model parameters in order to change flight envelope parameters. The advantage of this system is that it allows the aircraft to fly closer to its limits while maintaining controllability. However, certain limitations in computational power when using embedded controllers do not allow such a complex system to be applied to small affordable UAVs. In 2016, Lombaerts et al. (2016) developed an adaptive algorithm to achieve flight envelope protection during icing conditions (Lombaerts et al., 2018). This system can evaluate the error state and subsequently adjust the flight envelope boundaries; however, it relies on too many parameters and is computationally demanding.

To mitigate the loss of control, it is also possible to use nonlinear dynamics modeling, Bayesian system identification, and known aircraft limitations to estimate the flight envelope, as researched by Schuet et al. (2016). However, these methods can be computationally demanding and thus not usable in small UAVs.

Therefore, many approaches to flight phase classification perform the classification in the offline environment and are designed to work only offline, such as a system of advanced classifiers utilizing operators' EEG signals (Wang et al., 2019). Similar classification of flight phases using radar data was applied to solve air traffic problems, with the use classifiers on recorded flight data (Paglione & Oaks, 2006).

Flight phase classification can be carried out with a control algorithm or flight envelope protection system to achieve a higher degree of safety and controllability of the UAV, as researched by Donato et al. (2017), who created an envelope-aware management system that, based on the flight phase, could efficiently improve the response to dangerous scenarios by generating recovery solutions.

As previously mentioned, solving flight phase classification problems involves the use of algorithms with a focus on computational intelligence. For example, Shin and Kim created an adaptive flight envelope protection system that was aimed at improving the speed and accuracy of classifying standard and critical flight phases (Shin & Kim, 2016). They carried out their experiments in a simulation environment and used a model adaptive reference-like control scheme focused mainly on preventing departures from controlled flight.

In 2019, Liu et al. (2020) studied flight phase recognition using up to nine input parameters (Chin et al., 2019) and a Gaussian mixture model to classify flight phases in the offline environment. Using this methodology, they achieved an average accuracy of 90%. Kovarik et al. (2020) also applied machine learning methodologies to perform flight phase identification of civil aircraft using radar track data and four-dimensional GPS data.

Tian et al. (2017) described flight phase classification with the use of a decision tree; 30 input parameters were used with data obtained from a simulator and accuracy of 100% was achieved with the use of pruning.

The results of previous research show that flight phase classification problems are usually solved in the offline environment using rather complex algorithms and are mostly oriented to civil passenger aircraft and use many input parameters. There are not many papers evaluating this task with small unmanned aerial vehicles with very limited sets of parameters. The aim of the research in this paper was to find a computationally simple yet effective methodology that could identify the flight phases of a fixed-wing UAV with very limited computational power and a limited set of input parameters, in contrast to the more complex methodologies using large scales of input data from ADSB and aimed mostly at civilian aviation that have recently been successfully applied for this task (Kovarik et al., 2020; Kim et al., 2022; Zhang et al., 2022).

Based on the state-of-the-art research and from the methodological point of view, fuzzy inference systems and artificial neural networks were identified as good candidates for this task, as they are simple and have been successfully applied. The goal of the initial design of the system was to find the most important parameters obtained from the UAV sensors to reliably classify its basic flight phases with tests using real data, and the designed methodology had to be applicable in an embedded system with limited computational power. This means identification of a set of 3–5 input parameters and design of a minimal neural network architecture with feed-forward structure and a minimal rule-base of a fuzzy inference system applicable in real-time systems. The aim of the research was to evaluate the feasibility of such minimalistic design, with further prospects of improving it for more complex uses and applications. The developed approach should be unique in this regard as development of such system for small UAVs has not been published so far and can serve as a starting point for development of more complex flight phase classification systems.

The paper is organized according to the performed workflow as follows: Section 1 describes the UAV and sensors that were used to design the flight phase classification system perform testing flights and create a basic outline of the methodology. Section 2 describes the computational design of the flight phase classification algorithms and the experimental design for their testing using the Skydog UAV. Section 3 presents the testing results of the developed methodology using the obtained flight data, and the last two sections discuss and evaluate the results, and mention future research on simple flight phase classification algorithms and their applications.

1. Materials and methods

To design a flight phase classification system for a small affordable UAV, it is necessary to take into account limited computational resources and limited types and numbers of sensors with corresponding flight parameters. To resolve this parametric deficiency, which can result in some uncertainties, methodologies from the area of computational intelligence can be successfully applied. In this study, to cope with low computational demand, the authors selected two approaches, fuzzy inference systems and artificial neural networks, which will be explored further.

The aim was to create an initial design of a flight phase classifier that can be used in small, fixed-wing UAVs. The goal was to create a simple classifier with the use of advanced computational methods with a minimal set of input data. The advantage of the designed fuzzy inference system (FIS) is its simplicity. The FIS works with a predetermined set of rules that are directly aimed at the decision-making process. The fuzzy logic provides linguistic expressions as the output, which can be displayed directly to the pilot.

Artificial neural networks allow the use of obtained flight data and selected parameters, which can be used to train the network in flight phase classification tasks. Because of the low computational demand and simplicity of the system, a feedforward artificial neural network with a relatively low number of neurons and standard training algorithms was used. The use of artificial neural networks with the selection of parameters to be used for flight phase classification can be a proof of concept.

To test the methodology and the selection of main parameters for flight phase classification, a small, fixed-wing SkyDog UAV was used. This UAV is equipped with an electric engine and onboard avionics, which ensures the real-time collection of flight data. The aircraft has a takeoff weight of 7 kilograms and wingspan of 2 meters. The Pixhawk PX-4 autopilot, which has an embedded processor for smooth operation of control algorithms and collection of flight data, was selected as the control unit. The use of this advanced control unit is crucial in the implementation of adaptive algorithms for small UAVs, as it has limited input data due to a limited number of sensors (Lassak et al., 2020; Kurdel et al., 2022; Leško et al., 2019). The workflow of the design of flight phase classification algorithms with limited input data is shown in Figure 1.

The Pixhawk PX4 control unit is the main part of the SkyDog avionics equipment. It enables the collection of flight data from individual sensors and performs and processes calculations. One of the functions that it offers is the recording of flight data on an SD card. It also offers the ability to control the aircraft in various modes, including automatic mode. The connection of sensors and implementation of the Pixhawk in the SkyDog is shown in Figure 2. The sensors listed in Table 1 were used to obtain the parameters for the flight phase classification system. The airspeed from an MS4525DO sensor (Measurement Specialties, 2013), to which the pitot-static tube was connected, was used as an input parameter for the developed system. A built-in MS5611 barometric sensor was used to obtain the barometric altitude (Measurement Specialties, 2012), which was used with the altitude from the GPS module to more accurately determine the altitude at which the UAV is located. An MPU6000 3-axis accelerometer/gyroscope sensor (InvenSense, 2013) was used to determine angular velocities.

2. Design of flight phase classification system

In the initial flight phase classification system design, it was decided to test the reliability of the methodologies from the perspective of computational intelligence in classifying the current state of flight of UAV into 3 basic situational frames: take-off, flight, and landing. If the system performed reliably, it could be expanded further for other flight situations. The design should also work with a

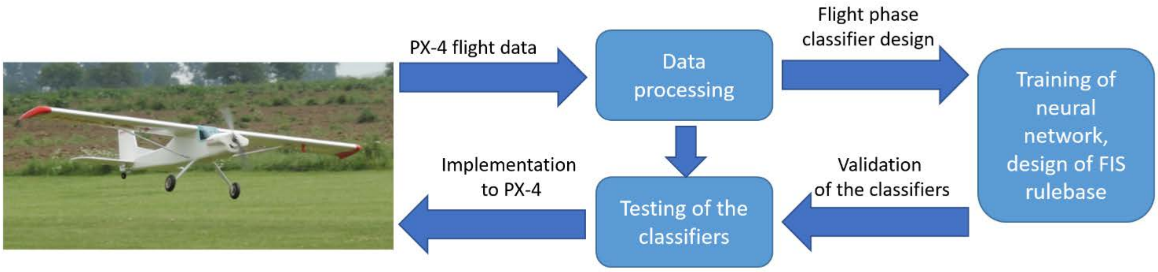

Figure 1. SkyDog UAV and the workflow of the design

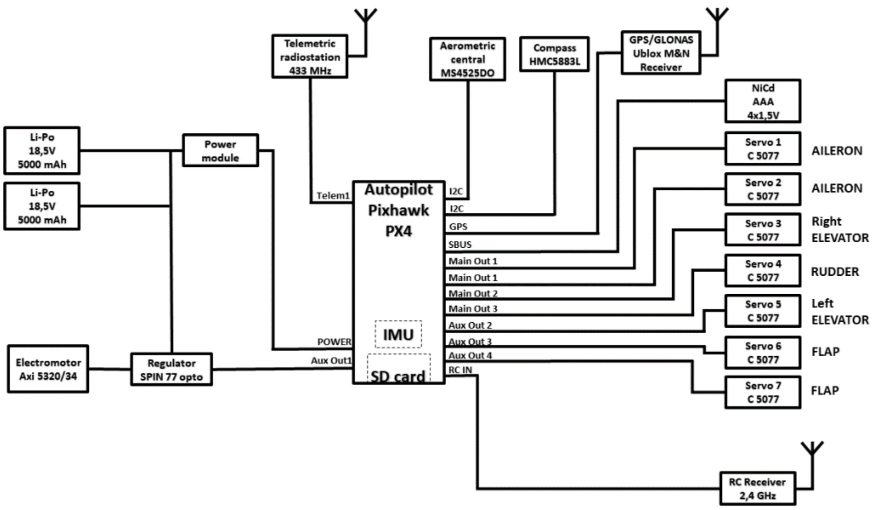

Figure 2. Block diagram of avionics in SkyDog aircraft (Leško et al., 2019)

Table 1. Sensors used for data collection

Parameter	Sensor	Error
Airspeed	MS4525DO	±0,25 %
Barometric pressure	MS5611	±2.5 mbar
N/A	N/A	Nonlinearity
Acceleration/angular velocity	MPU6000 3-axis accelerometer/gyroscope	Accelerometer 0.5% Gyroscope 0.2%

minimal set of input parameters. After a careful analysis, 3 basic parameters were selected as inputs for the fuzzy inference system and the artificial neural network:

- Velocity: Indicated aircraft velocity (km/h) as measured by the airspeed sensor (Table 1);
- Altitude: Barometric altitude (m) as measured by the sensor measuring barometric pressure;
- Current: Actual value of current (A) supplied to the electric engine by the batteries.

The selection of these parameters and situational classes implies that both the FIS system and the ANN will contain 3 inputs and 3 outputs, thus meeting the goals of simplicity of the system and real-time operability.

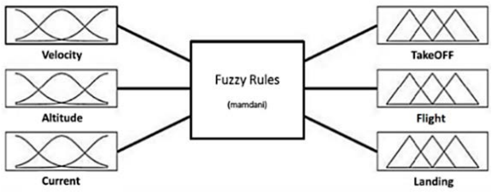

Figure 3. Fuzzy inference system for flight phase classification

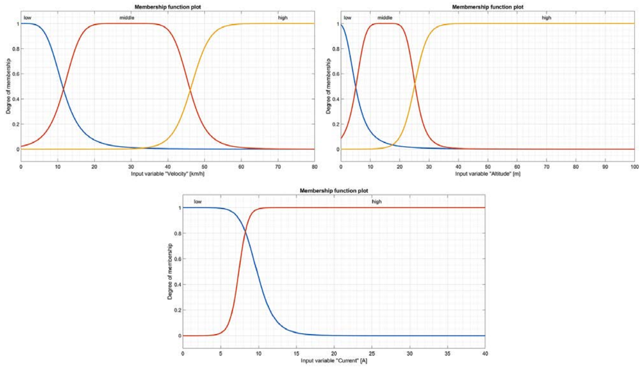

Figure 4. FIS classifier of membership functions for 3 input variables

2.1. Design and implementation of the system using fuzzy logic

The fuzzy inference system (FIS) uses non-Boolean fuzzy logic and IF-THEN rules defined by an expert (Kaleva, 1987). Fuzzy inference systems were derived from expert systems, which make them ideal candidates for solving flight phase classification problems. The use of fuzzy logic implies that transitions between the output states of the system are smooth. The diagram in Figure 3 shows a fuzzy inference system for flight phase classification with selected inputs and outputs (Klir & Yuan, 1995). The Mamdani type has been selected for application in flight phase classification, as it is desirable to have the outputs of the system as fuzzy membership functions. The goal of the design is to create an FIS with minimum rules so that it can be easily modified and applied in real-time control systems. In the design phase a heuristic expert approach using flight envelope setting has been used to create rules and membership functions of the FIS system.

The FIS classification block evaluates the combination of input parameters velocity, altitude, and current, represented by linguistic variables, and their corresponding values, defined by individual membership functions low, middle, and high. All combinations create a rule-based design, as shown in Table 2. The output of the system consists of linguistic variables Take-off, Flight, and Landing, with the

Table 2. FIS classifier rules

No.	Flight phase classification conditions of fuzzy inference system					
	INPUTS			OUTPUTS		
	AIRSPEED	ALTITUDE	CURRENT	Take-off	Flight	Landing
1.	Low	Low	Low	No	No	Yes
2.	Low	Low	High	Yes	No	No
3.	Low	Medium	Low	No	No	Yes
4.	Low	Medium	High	Yes	No	No
5.	Low	High	Low	No	Yes	No
6.	Low	High	High	No	Yes	No
7.	Medium	Low	Low	No	No	Yes
8.	Medium	Low	High	Yes	No	No
9.	Medium	Medium	Low	No	No	Yes
10.	Medium	Medium	High	No	Yes	No
11.	Medium	High	Low	No	Yes	No
12.	Medium	High	High	No	Yes	No
13.	High	Low	Low	No	Yes	No
14.	High	Low	High	Yes	No	No
15.	High	Medium	Low	No	Yes	No
16.	High	Medium	High	No	Yes	No
17.	High	High	Low	No	Yes	No
18.	High	High	High	No	Yes	No

corresponding values represented by membership functions yes and no. Figure 4 shows the membership functions for the variables velocity, altitude, and current. They are of generalized bell-shaped membership function (gbellmf) type to secure smooth transitions between states.

Table 2 lists the rules used in the FIS classification block. Based on these rules, the system classifies flight phases, with the output given in the interval <0.25–0.75>.

2.2. Design and implementation of the system using artificial neural networks

As in the previously designed FIS, 3 input parameters were fed into the system: velocity, altitude, and electric current. This indicates that the network has 3 neurons in its input and 3 neurons in its output, corresponding to 3 flight phases: Take-off, Flight, and Landing.

Figure 5 shows individual layers in the neural network used for flight phase classification. Several configurations of the feedforward neural network were tested in its design. In training experiments using flight data obtained from the SkyDog drone, the 3-20-10-3 architecture was found to be optimal when trained by the Levenberg–Marquardt backpropagation algorithm (Kanzow et al., 2004). The training algorithm converged on average after 12 training iterations given the training datasets from individual flights. Hidden layer 1 consisted of 20 neurons, hidden layer 2 consisted of 10 neurons, and the output layer consisted of 3 neurons. Given the low complexity of the trained network, which was its design goal the computational time for a single iteration was on average 0.9 seconds using the LM algorithm. A higher numbers of layers and neurons in hidden layers did not provide better classification results on training sets as well as training algorithms based on standard gradient descent methods, which needed more than 300 training iterations.

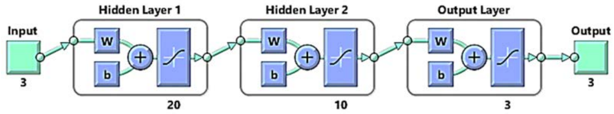

Figure 5. Feedforward neural network with two hidden layers

2.3. Experimental setup

The goal of the experiment was to compare two created classification systems that could classify flight phases with a minimal set of input data. Flights were performed based on predetermined flight plans, consisting of specific maneuvers to test the functionality of the designed systems. The flight plans are described in Table 3. The duration of each flight was measured from the start of the Take-off phase until the end of the Landing phase. Flight 1 took 142 s, flight 2 lasted 131 s, and flight 3 took 109 s to accomplish.

In the first flight, the flaps were set to 60%, which is classed as an incorrect UAV Take-off configuration. That was purposely set to monitor the systems' flight phase recognition and ensure correct classification. During the Flight phase in Flight 1, the correct classification was monitored while maximum flight velocity was exceeded. The entire flight took 142 seconds from the beginning of the Take-off phase until the end of the Landing phase.

Flight 2 took 131 s for total flight phase classification. It consisted of 2 Take-off, 2 Cruise, and 2 Landing phases, since it had a touch-and-go maneuver. Thus, all values were divided into 2 parts (duration of first part of flight + duration of second part of flight). The purpose of this flight was to create an unnatural condition for the system and monitor its classification.

Table 3. Flight plans for individual flights

Flight 1	
Take-off phase	Set flaps to 60% (incorrect take-off configuration setting) to monitor successful phase recognition using individual systems Longer and steeper take-off
Flight phase	Exceed maximum velocity of 105 km/h to monitor correct flight phase classification at higher velocity
Landing phase	Set flaps from 0 to 60%
Flight 2	
Take-off phase	Set flaps to 30%
Flight phase	Set flaps to 0% Perform simple circuits with minimum tilt, reasonable flight velocity and altitude
Landing phase	Set flaps to 60% Perform touch-and-go maneuver Perform last circuit and land, then wait until UAV completely stops
Flight 3	
Take-off phase	Set flaps to 30% Perform smooth start with flaps at lateral inclination of 20–40%
Flight phase	Set flaps to 0% Perform flight at reasonable speed at different flight altitudes
Landing phase	Set flaps to 60% Perform landing at very low velocity (under 38 km/h)

In flight No. 3, limit situations and the correct classification of individual flight phases were tested. The plan was to land at a low speed (below 38 km/h), but with the given flap configuration it was not possible to keep the velocity lower than 38 km/h. This phase was therefore performed as a normal landing.

3. Results

The output data were compared to etalons that were created based on the input data and predetermined flight plans. The etalon represents the ideal flight phase duration that should be reached for algorithms to correctly classify individual phases.

Table 4 lists the etalon values for the individual flight phases during the 3 flights. The duration parameter represents how long each individual flight phase lasted during the whole flight. The start of phase parameter represents the second when the flight phase started, and the end of phase parameter represents when the corresponding flight phase ended. The total column shows the duration of all the flight phases together and the whole duration of the flight from the Take-off phase until the end of the Landing phase.

Equations (1)–(3) were used to calculate the statistical parameters recall, precision, and accuracy (Olson & Delen, 2008):

$$Recall = \frac{Duration\ of\ \boldsymbol{TP}\ phase\ classification\left(Take\text{-}off,\ Flight,\ Landing\right)}{Total\ duration\ of\ \boldsymbol{TP}\ and\ \boldsymbol{FN}\ phase\ classification}; \tag{1}$$

$$Precision = \frac{Duration\ of\ \boldsymbol{TP}\ phase\ classification\left(Take\text{-}off,\ Flight,\ Landing\right)}{Total\ duration\ of\ \boldsymbol{TP}\ and\ \boldsymbol{FP}\ phase\ classification}; \tag{2}$$

$$Accuracy = \frac{Duration\ of\ \boldsymbol{TP}+\boldsymbol{TN}\ phase\ classification\left(Take\text{-}off,\ Flight,\ Landing\right)}{Total\ duration\ of\ \boldsymbol{TP}+\boldsymbol{TN}+\boldsymbol{FP}+\boldsymbol{FN}\ phase\ classification}, \tag{3}$$

where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives (Olson & Delen, 2008).

Flight 1

The graphs displayed in Figure 6 show input values of Flight 1, together with etalon markings. The duration of Flight 1 was 142 s. The whole course of the flight is given in Table 3.

Table 4. Etalon values of three flights

Etalon (Flight 1)	Duration (s)	Start of phase (s)	End of phase (s)	Total (s)
Take-off	6	25	31	142
Flight	100	32	132	
Landing	36	133	169	
Etalon (Flight 2)	N/A	N/A	N/A	N/A
Take-off	3/2	6/101	9/103	N/A
Flight	88/21	10/104	98/125	93/38
Landing	2/15	99/126	101/141	N/A
Etalon (Flight 3)	N/A	N/A	N/A	N/A
Take-off	8	10	18	N/A
Flight	81	19	100	109
Landing	20	101	121	N/A

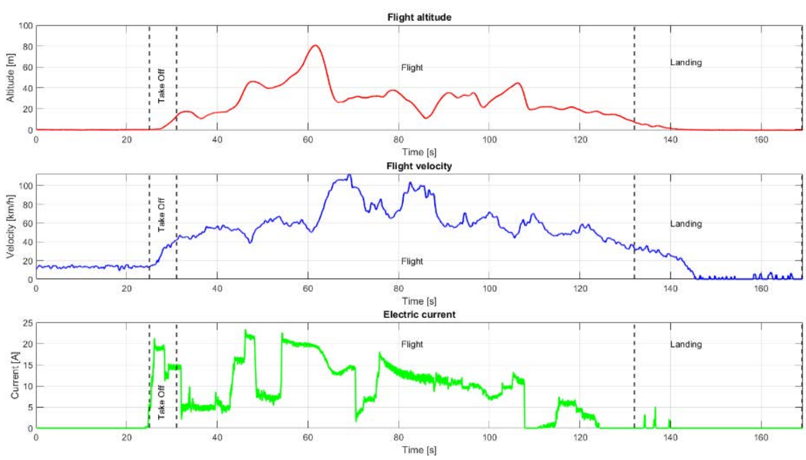

Figure 6. Input data for Flight 1

Table 5 lists the flight phase classification values using the mentioned systems. The delay value represents the flight phase determination delay in seconds. For example, the FIS had the longest delay in the Flight phase, which took >7 s to classify, and it classified the end of the phase 6 s sooner. For the feedforward neural network, the delays of the Flight phase were +0 s for the start of the phase and +1 s for the end of the phase.

The most accurate was the feedforward neural network, with 97.18% accuracy. It was able to accurately classify 138 s out of a total time of 142 s. The ANN also had the least delays in classifying the individual flight phases. As can be seen from the graphs in Figure 7 and the values in Table 5, the feedforward neural network had the least delays.

Figure 7 shows that both systems classified the Take-off phase correctly. The most accurate system can be considered the ANN, which deviated minimally from the etalon. The Flight phase was classified correctly by both methods, but the difference was in the classification delay.

Flight 2

Input values from Flight 2 are displayed in Figure 8, and the whole course of the flight is given in Table 3. It took 131 s from the start of the Take-off phase until the end of the Landing phase. The main goal of flight 2 was to monitor the classifier's output during the touch-and-go maneuver.

As can be seen in Table 6, the most accurate classifier was the ANN. It was able to accurately classify not only the individual phases but also the phase of secondary Flight

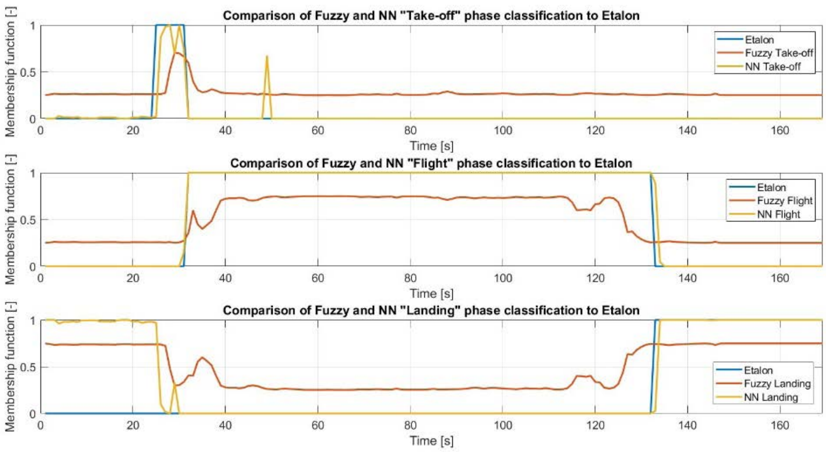

Figure 7. Comparison of NN output with etalon values from flight course 1

Table 5. FIS classifier output values for Flight 1

Phase	Delay				Precision		Recall		Accuracy (%)		Overall accuracy (%)	
	Start of phase (s)		End of phase (s)									
	FIS	NN	FIS	NN	FIS	NN	FIS	NN	FIS	NN	FIS	NN
Take-off	+4	+3	+1	–1	1	1	0.6	1	50	50	88.73	97.18
Flight	+7	+0	–6	+1	1	1	0.94	1	87	100		
Landing	–6	+1	+0	+0	0.82	1	1	1	81.81	97		

Table 6. FIS classifier output values for flight course 2

Phase	Delay				Precision		Recall		Accuracy (%)		Overall accuracy (%)	
	Start of phase (s)		End of phase (s)									
	FIS	NN	FIS	NN	FIS	NN	FIS	NN	FIS	NN	FIS	NN
Take-off	+2/Not identified	+1/+3	+1/Not identified	+0/+1	1	0.75	0.5	0.6	20	60	74.05	92.37
Flight	+4/Not identified	+2/+2	–5/Not identified	+1/+2	1	0.98	0.76	0.99	72.47	97.25		
Landing	–5/Not identified	+1/+3	Not identified	+2/+0	0.38	0.86	1	0.86	100	70,59		

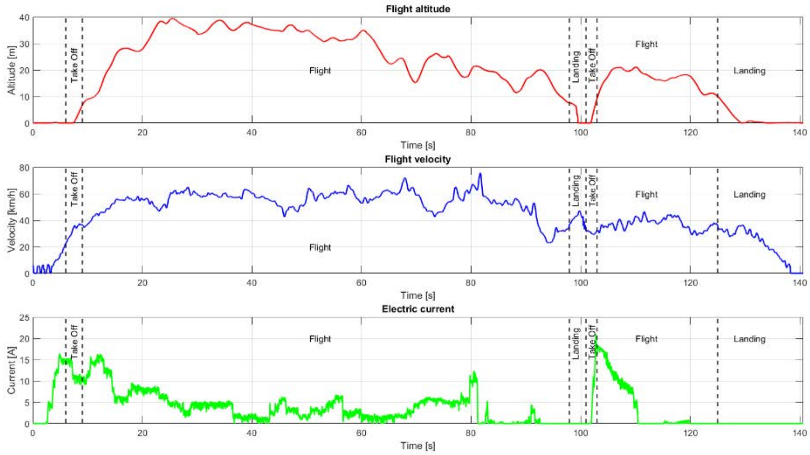

Figure 8. Input data for Flight 2

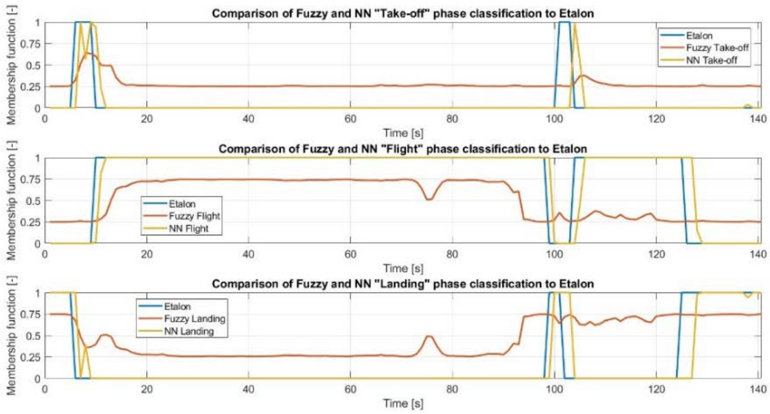

Figure 9. Comparison of NN output with etalon values from flight course 2

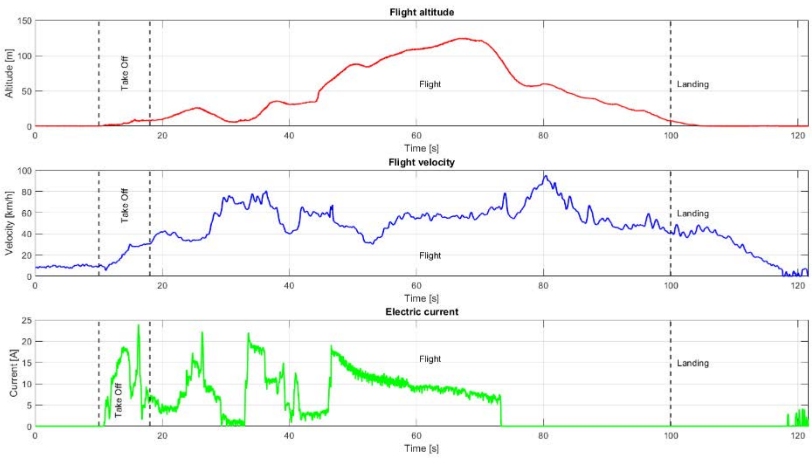

Figure 10. Input data for Flight 3

and Take-off after the touch-and-go maneuver. The only drawback was the delayed determination of the Landing phase when approaching the runway during the touch-and-go maneuver.

As shown in Table 6 and Figure 9, the FIS was unable to classify the correct transition from the Landing phase to the secondary Take-off phase. As these had lower velocities and lower altitudes during the flight circuit, the FIS evaluated this part as Landing. The most accurate was the ANN, which classified flight phases with more than 92% overall accuracy.

Flight 3

Flight 3 took 109 s from the beginning of the Take-off phase until the end of the Landing phase. The whole flight plan for Flight 3 is given in Table 3. The graphs in Figure 10 display input data used for flight phase classification.

Table 7. FIS classifier output values for flight course 3

Phase	Delay				Precision		Recall		Accuracy (%)		Overall accuracy (%)	
	Start of phase (s)		End of phase (s)									
	FIS	NN	FIS	NN	FIS	NN	FIS	NN	FIS	NN	FIS	NN
Take-off	+6	+4	–2	+1	1	0.8	0.25	0.57	12.5	50.00	81.65	92.66
Flight	+11	+3	–1	+1	1	0.99	0.93	0.99	85.19	97.53		
Landing	+2 s	+1	+0	+0	0.70	0.86	1	0.95	95.00	90.00		

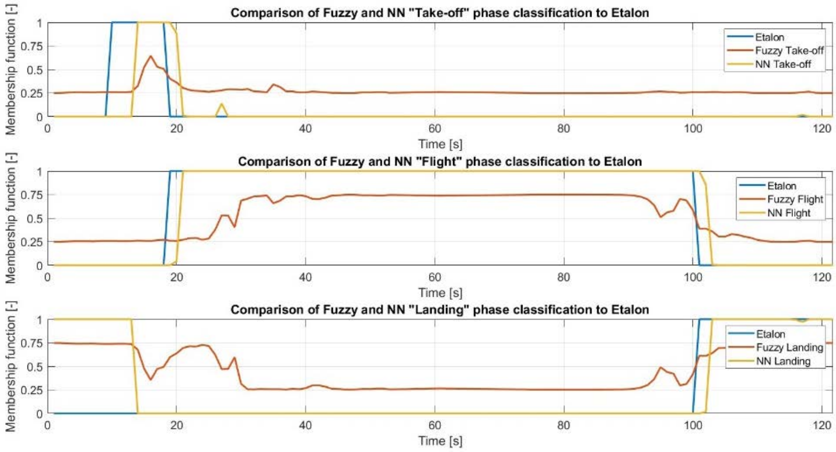

Figure 11. Comparison of NN output with etalon values from flight course 3

As can be seen from Table 7, the NN represents a very accurate system for determining individual flight phases, with accuracy of 92.66%.

The detailed performance of both classifiers can be seen in Figure 11 presenting the third flight as a time plot. As in previous flights, the FIS was the slowest and least accurate. The accuracy of flight phase identification was 78.84%. In addition, the delays that occurred in the flight-phase classification were up to 11 s, which is a considerable amount of time. The most accurate in flight 3 was the neural network, which again confirmed its superiority over the FIS.

4. Discussion

The data from three flights with special maneuvers, as described in Section 3.3, were used as inputs for the flight phase classification system, which used two systems for classification, fuzzy inference and artificial neural network. After being tested in an offline environment, it can be said that the system worked properly and could correctly identify each flight phase. With the proposed methodology, it was also possible to choose the three optimal parameters for reliable flight phase classification and create a lean classifier, which can be implemented in embedded controllers with limited computational power. The trained neural networks and fuzzy inference system are suitable for use on other available autopilots like ardu-pilot or other embedded platforms, which are able to measure the selected input parameters.

As the research in this paper shows, each system has advantages and disadvantages. Fuzzy logic brings a smooth transition between each phase. Thus, with further improvement, it could be used to predict upcoming situations. A big disadvantage of the FIS is that the algorithm must be programmed and it consists of a strict set of rules to correctly classify the flight phase. This means that the user of the algorithm must manually set boundaries, which, if met, will signify specific flight phases. Its overall average accuracy was calculated as 81.48% for three chosen flights. Table 8 shows a deficiency of correct take-off classification, with only 27.5% accuracy. It can be improved by including more input parameters and programming them together with existing rules. The FIS was also unable to correctly identify the touch-and-go maneuver. It would probably be more successful with added input parameters and an expanded rule base.

The feedforward neural network is one of the best options for performing the task of classifying flight phases with minimal input parameters. If there existed flight data with all possible situations, it could serve the neural network in terms of further improving the already high success rate. The overall average accuracy of even a simple feedforward ANN was 94.07%. The neural network was able to correctly classify 98.26% of the Flight phase, which can be considered as reasonable accuracy.

Table 8. Overall average accuracy of FIS and NN

FIS	Average accuracy (per phase)		Overall average accuracy	
	FIS	NN	FIS	NN
Take-off	27.50%	53.33%	81.48%	94.07%
Flight	81.55%	98.26%		
Landing	92.27%	85.94%		

In Table 8, the average accuracy of the classification of individual phases by the FIS and ANN is calculated. The ANN had the highest average accuracy, reaching 94.07%. This system managed to recognize all flight phases correctly. It was also able to recognize a touch-and-go maneuver, which FIS was unable to do. A big advantage of the ANN is the possibility of training it further with expanded training datasets. By training the network, it should be possible to improve the accuracy of flight phase evaluation, even with the limited architecture of the network.

Conclusions

Both the fuzzy inference system and the artificial neural network were subjected to several tests and, under normal conditions, performed adequately, as shown in the results. However, when tested against special flight conditions with the SkyDog UAV, only the neural network performed adequately; it can be concluded that this approach met the design goal of working with a only three input parameters, which were identified and a lean network architecture, which are the main contributions of the study. Contrary to most papers, which base their results on simulations the performance of the designed classifiers has been evaluated using real-world data and were implemented in the Pixhawk PX-4 autopilot. Performance could be improved by further training on an even larger dataset with more training flights with different conditions. The fuzzy inference system in the role of flight phase classifier would need more input parameters and a larger rule base to be equally successful, but it can serve in a support role to improve switching between the individual situational frames.

In conclusion, it can be stated that selecting three input parameters is sufficient for reliable flight phase classification when using even a small feedforward neural network. The research can be expanded by creating sub-classifiers in the form of other small artificial neural networks, in order to further decompose the three basic flight situations into micro situations, such as climb/descent, loiter, or abnormal flight situations, such as an aerodynamic stall. Fusion of the FIS and neural network classifiers could be another path of follow-up research. The resulting classifier can be connected to a diagnostic system or be used in advanced drone situational control systems. The developed classification system can be taken as proof of concept for implementation in embedded real-time control systems with limited computational power and resources, and it shows that even a simple neural network can produce adequate results in such an application with correctly identified inputs and architecture.

Funding

This work was funded by the project Mobile Monitoring System for the Protection of Isolated and Vulnerable Population Groups against Spread of Viral Diseases, ITMS code 313011AUP1, co-funded by the European Regional Development Fund under the Operational Programme Integrated Infrastructure.

This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-20-0546.

Author contributions

JL and RA were responsible for the conceptualization and formal analysis and for preparing the original draft. JL, RA, RB, and LF were responsible for methodology and validation. JL, RB, and MH were responsible for data curation. JL and MH were responsible for editing and visualization. RA was responsible for project administration and for supervision.

References

Basheer, I. A., & Hajmeer, M. N. Artificial neural networks: Fundamentals, computing, design, and application. *Journal of Microbiological Methods, 43*(1), 3–31. https://doi.org/10.1016/S0167-7012(00)00201-3

Chin, H. J. H., Payan, A., Johnson, C., & Mavris, D. N. (2019). Phases of flight identification for rotorcraft operations. In *AIAA Scitech 2019 Forum*. San Diego, California. https://doi.org/10.2514/6.2019-0139

Donato, P. F., Balachandran, S., McDonough, K., Atkins, E. M., & Kolmanovsky, I. V. (2017). Envelope-aware flight management for loss of control prevention given rudder jam. *Journal of Guidance Control and Dynamics*, *40*(4), 1027–1041. https://doi.org/10.2514/1.G000252

Gill, S. J., Lowenberg, M. H., Neild S. A., Krauskopf, B., Puyou, G., & Goetzee, E. (2013). Upset dynamics of an airliner model: A nonlinear bifurcation analysis. *Journal of Aircraft, 50*(6), 1832–1842. https://doi.org/10.2514/1.C032221

InvenSense. (2013). *MPU-6000 and MPU-6050 product specification revision 3.4*. https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

Kaleva, O. (1987). Fuzzy differential equations. *Fuzzy Sets and Systems*, *24*(3), 301–317. https://doi.org/10.1016/0165-0114(87)90029-7

Kanzow, Ch., Yamashita, N., & Fukushima, M. (2004). Levenberg–Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *Journal of Computational and Applied Mathematics, 172*(2), 375–397. https://doi.org/10.1016/j.cam.2004.02.013

Kim, D., Seth, A., & Liem, R. P. (2022). Data-enhanced dynamic flight simulations for flight performance analysis. *Aerospace Science and Technology, 121*, 107357. https://doi.org/10.1016/j.ast.2022.107357

Kovarik, S., Doherty, L., Korah, K., Mulligan, B., Rasool, G., Mehta, Y., Bhavsar, P., & Paglione, M. (2020). Comparative analysis of machine learning and statistical methods for aircraft phase of flight prediction. In *The 9th International Conference on Research in Air Transportation*. National Science Foundation.

Klir, G. J., & Yuan, B. (1995). Fuzzy sets and fuzzy logic – theory and applications. *Choice Reviews*. https://doi.org/10.5860/CHOICE.33-2786

Kurdel, P., Češkovič, M., Gecejová, N., Adamčík, F., & Gamcová, M. (2022). Local control of unmanned air vehicles in the mountain area. *Drones, 6*(2), 54. https://doi.org/10.3390/drones6020054

Lassak, M., Draganova, K., Blistanova, M., Kalapos, G., & Miklos, J. (2020). Small UAV camera gimbal stabilization using digital filters and enhanced control algorithms for aerial survey and monitoring. *Acta Montanistica Slovaca, 25*(1), 127–137. https://doi.org/10.46544/AMS.v25i1.12

Leško, J., Schreiner, M., Megyesi, D., & Kovács, L. (2019, 28–29 November). Pixhawk PX-4 autopilot in control of a small unmanned airplane. In *2019 Modern Safety Technologies in Transportation (MOSATT)*. Kosice, Slovakia. https://doi.org/10.1109/MOSATT48908.2019.8944101

Liu, D., Xiao, N., Zhang, Y., Peng, X. (2020). Unsupervised flight phase recognition with flight data clustering based on GMM. In *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (pp. 1–6). IEEE. https://doi.org/10.1109/I2MTC43012.2020.9128596

Lombaerts, T., Looye, G., Seefried, A., Neves, M., & Bellmann, T. (2016). Development and concept demonstration of a physics based adaptive flight envelope protection algorithm. *IFAC-PapersOnLine, 49*(5), 248–253. https://doi.org/10.1016/j.ifacol.2016.07.121

Lombaerts, T., Looye, G., Seefried, A., Neves, M., & Bellmann, T. (2018). Proof of concept simulator demonstration of a physics based self-preserving flight envelope protection algorithm. *Engineering Applications of Artificial Intelligence, 67*, 368–380. https://doi.org/10.1016/j.engappai.2017.08.014

Measurement specialties. (2013). *MS4525DO*. https://www.mouser.com/datasheet/2/418/MS4525DO-710170.pdf

Measurement specialties. (2012). *MS5611*. https://datasheetspdf.com/pdf-file/921406/measurement/MS5611-01BA03/1

Norouzi, R., Kosari, A., & Sabour, M. H. (2019). Real time estimation of impaired aircraft flight envelope using feedforward neural networks. *Aerospace Science and Technology*, *90*, 434–451. https://doi.org/10.1016/j.ast.2019.04.048

Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques* (1st ed.). Springer.

Paglione, M., & Oaks, R. (2006). Determination of horizontal and vertical phase of flight in recorded air traffic data. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Aerospace Research Central. https://doi.org/10.2514/6.2006-6772

Perhinschi, M. G., Moncayo, H., & Davis, J. (2010). Integrated framework for artificial immunity-based aircraft failure detection, identification, and evaluation. *Journal of Aircraft, 47*(6), 1847–1859. https://doi.org/10.2514/1.45718

Schuet, S., Lombaerts, T., Acosta, D., Kaneshige, J., Wheeler, K., & Shish, K. (2016). Autonomous flight envelope estimation for loss-of-control prevention. *Journal of Guidance Control and Dynamics, 40*(4), 847–862. https://doi.org/10.2514/1.G001729

Shin, H., & Kim, Y. (2016). Flight envelope protection of aircraft using adaptive neural network and online linearisation. *International Journal of Systems Science, 47*(4), 868–885. https://doi.org/10.1080/00207721.2014.906769

Tang, L., Roemer, M. J., Ge, J., Crassidis, A. L., Prasad, J. V., & Belcastro, C. M. (2009, 10–13 August). Methodologies for adaptive flight envelope estimation and protection. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. Chicago, Illinois. https://doi.org/10.2514/6.2009-6260

Tian, F., Cheng, X., Meng, G., & Xu, Y. (2017). Research on flight phase division based on decision tree classifier. In *The 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)* (pp. 372–375). IEEE. https://doi.org/10.1109/CIAPP.2017.8167242

Wang, D., Wu, D., He, J., & Gu, H. (2019). A study of EEG-based flight phase recognition. In *2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT)* (pp. 640–645). IEEE. https://doi.org/10.1109/ICCASIT48058.2019.8972996

Wilborn, J. E., & Foster, J. V. (2004, 16–19 August). Defining commercial transport loss-of-control: A quantitative approach. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. Providence, Rhode Island. https://doi.org/10.2514/6.2004-4811

Zhang, Q., Mott, J. H., Johnson, M. E., & Springer, J. A. (2022). Development of a reliable method for general aviation flight phase identification. In *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 11729–11738. https://doi.org/10.1109/TITS.2021.3106774