

## Data Selection for Training the Neural Fuser Applied to Autonomous UAV Navigation

G. PENHA NETO<sup>1</sup>, H. F. CAMPOS VELHO<sup>2\*</sup> and E. H. SHIGUEMORI<sup>3</sup>

Received on December 10, 2020 / Accepted on July 29, 2022

**ABSTRACT.** Over the past few years, there has been a steady increase in the use of aircraft vehicles, in particular unmanned aerial vehicles (UAV). UAV navigation is generally controlled by a human pilot. But the challenge for the scientific community is to carry out autonomous navigation. Some solutions have been proposed for the UAV autonomous navigation. Studies indicate as a solution to use data fusion and/or image processing navigation. Kalman Filter (KF) can be employed as a data fuser, but the KF has disadvantages. An alternative to the KF is based on artificial intelligence. Here, the KF is replaced by a self-configured neural network. This work investigates a way to select data for training the neural fuser, based on cross-validation techniques. The results are compared to the data fusion done by a KF.

**Keywords:** self-configured neural network, unmanned aerial vehicle, cross-validation, k-fold.

### 1 INTRODUCTION

Over the past few years, there has been a steady increase in the use of vehicles aircraft, in particular, unmanned aerial vehicles (UAV). UAV applications take place in several areas, ranging from civil and military activities, such as surveillance operations, environmental monitoring, aerial survey, and cargo transportation. Particularly, the application in agricultural monitoring and precision agriculture stands out [11, 12, 14, 15, 28, 31, 32].

UAV navigation is generally controlled by a human pilot. This pilot controls the UAV through a ground station, which sends commands to the UAV by a radio signal [30]. Another form of navigation of UAV is automatic navigation. Automatic navigation does not require human pilot for flight control, because this type of navigation is made from a pre-established route [30]. But

---

\*Corresponding author: Haroldo Fraga de Campos Velho – E-mail: haroldo.camposvelho@gmail.com

<sup>1</sup> Faculdade de Tecnologia (FATEC), Av. Cesare Monsueto Giulio Lattes, 1350, Distrito Eugênio de Melo, São José dos Campos, SP, 12247-014, Brazil – E-mail: gerson.penha@inpe.br <https://orcid.org/0000-0002-5161-9255>

<sup>2</sup> Instituto Nacional de Pesquisas Espaciais (INPE), Av. dos Astronautas, 1758 – Jardim da Granja, São José dos Campos, SP, 12227-010, Brazil – E-mail: haroldo.camposvelho@gmail.com <https://orcid.org/0000-0003-4968-5330>

<sup>3</sup> Instituto de Estudos Avançados (IEAv), Trevo Coronel Aviador José Alberto Albano do Amarante 01, Putim, São José dos Campos, SP, 12228-001, Brazil – E-mail: elcio@ieav.cta.br <https://orcid.org/0000-0001-5226-0435>

the challenge for the scientific community is to define the autonomous navigation methodology [14].

Autonomous navigation eliminates the control of a human pilot. So you need sensors, like a Global Navigation Satellite System (GNSS) and an Inertial Navigation System (INS). These sensors assist an embedded control system in locating and positioning the UAV during navigation [30].

However, there are problems associated with the use of these sensors. The main troubles are interference, which can be caused intentionally or by natural actions, making autonomous navigation unfeasible. The main ways of intentional interference are called jamming and spoofing [13,22]. These attacks affect the estimation of the UAV location and position, provided by the GNSS receiver [5, 18].

In addition to problems with the GNSS signal, contingencies with the INS must also be considered. The problem that occurs with the INS, in particular with low-quality equipment, is the accumulation of the drift error, which, if not corrected, can imply a great divergence between the estimated position and the actual position of the aircraft [16].

Beyond intentional interferences, natural interferences must also be highlighted. In this sense, two phenomena stand out, which are ionospheric scintillation and the magnetic anomaly of the South Atlantic [17,29]. From these natural phenomena, scintillation significantly affects the GNSS signal transmission, impacting the autonomous navigation of UAVs.

The aforementioned problems have already been discussed and some solutions have been proposed for the autonomous navigation of UAVs when the GNSS signal is not available. Studies indicate as a solution the use of data fusion and/or image processing navigation [2, 19,36],

The displacement of a UAV during the flight is something that happens in fractions of a second. Therefore, data fusion techniques must process information with maximum efficiency, that is, at the highest speed possible. Kalman Filter (KF) is one technique applied as a data fuser [9], but it has high computational complexity, becoming a disadvantage. Another issue linked to the KF is the assumption of Gaussian statistics. The high computational effort of the KF on small UAVs can become a problem if there is no onboard computer capable of supporting KF timely execution for data fusion [20].

Methods based on artificial intelligence can be an alternative to KF [1, 10, 36]. Cintra and co-authors pointed out the Kalman filter and neural network as  $O(N^3)$  and  $O(N^2)$  [8], respectively. The complexity analysis of these algorithms shows the interest for searching a strategy with a lower computational effort. In a previous study, the KF was replaced by a self-configured neural network [24]. This neural network was called a neural fuser.

Data fusion operates with detection, association, correlation, estimation, and combination of data from different sensors, with different types of data [6]. In this sense, data fusion becomes an interesting alternative for application in the autonomous navigation of UAVs. With more sensors and data types combined, a larger amount of information can be used in the estimations, applying the

fuser system for the autonomous UAV navigation during the absence of the GNSS signal [34]. Data fusion applied to autonomous UAV navigation uses data from several sensors, where the selection of data for training the neural fuser becomes important. This work investigates an automatic way to select data for training the neural fuser based on cross-validation techniques. The results are compared to the data fusion done by a KF.

## 2 SELF-CONFIGURED NEURAL NETWORK

The displacement of a UAV during the flight is something that happens in fractions of a second. Therefore, data fusion techniques must process information with maximum efficiency, that is, at the highest speed possible. Because of this issue and also for seeking to reduce the computational complexity related to the KF, a self-configured multi-layer perceptron (MLP) neural network was used to perform the data fusion.

The use of neural networks has been employed to several applications. For this reason, neural networks can be considered as a set of techniques already consolidated, under constant evolution or adaptation [25]. MLP is widely used by the scientific community, designed to solve linearly inseparable problems, which could not be solved by the single layer perceptron network. Due to their robustness, MLPs are also applied in data fusion [7, 35].

It is important to inform why a self-configured neural network was used. Given that the displacement of a UAV during flight is something that happens in fractions of a second, the response of the neural fuser needs to be as fast as possible. Thus, a solution to achieve the desired performance is the construction of the neural fuser in dedicated hardware by field programmable gate array (FPGA) [23].

The size of the topology of the neural network can make its construction in dedicated hardware unfeasible. For this reason, an existing approach delivers an optimized topology for the neural fuser. This approach consists of applying the Multiple Particle Collision Algorithm (MPCA) [4, 21, 23]. The MPCA is a version of the meta-heuristic particle collision algorithm (PCA), and this, in turn, was inspired by the physics of particle collision reactions in a nuclear reactor, in particular in the scattering and absorption behaviors.

The objective of the MPCA is to find the optimized architecture of an MLP [4]. The algorithm determines the ideal values for the neural network parameters. This optimization occurs by minimizing a cost function, which is defined as:

$$J(Q) = \Phi(x, y) \times \frac{\rho_1 E_t(Q) + \rho_2 E_g(Q)}{\rho_1 + \rho_2} \quad (2.1)$$

$$\Phi(x, y) = \theta_1 [e^{x^2}] + \theta_2 [y] + 1 \quad (2.2)$$

where  $Q$  is the unknown set of parameters: number of hidden layers, number of neurons per each hidden layer, type of activation function, learning ratio and momentum parameters for the training process;  $E_t$  and  $E_g$  are training and generalization errors, respectively;  $\rho_1$  and  $\rho_2$  are parameters related to the balance between training and generalization errors – square differences

between the output neural network and the reference data, where the reference is the Kalman filter results;  $\Phi(x,y)$  is a measure of the complexity of the neural network, with  $x$  being the number of neurons and  $y$  the number of epochs until convergence during the training;  $\theta_1$  and  $\theta_2$  are adjustment parameters.

### 3 DATA USED

The data set was obtained from experiments carried out by the research group from the Wallenberg Laboratory for Information Technology and Autonomous Systems (WITAS), Department of Informatics and Information Science at the University of Linköping – Sweden [9]. The WITAS experiments were carried out by using autonomous UAV based on the Yamaha R-MAX model<sup>1</sup>. The total length of the UAV trajectory was about 3.6 km, and the drone is capable of taking off with a maximum weight of 95 kg.

The used sensors were loaded into an Inertial Measurement Unit (IMU), containing three accelerometers and three gyroscopes, providing the acceleration and angular rate of the UAV along the three axes; a barometric altitude sensor; and a monocular video camera mounted on a panoramic unit. Along with these sensors, latitude and longitude measurements from INS were also obtained.

The UAV control system was performed by 3 embedded computers. The flight computer was a Pentium-III PC-104 700 MHz. This computer was responsible for sending commands for takeoff, landing, and other basic movements during the flight. The second computer, the same model as the first, implemented image processing and features to control the camera's tilt. Finally, the last computer, a PC104 Pentium-M 1.4 GHz, implemented the data fusion – in this computer, the processing from the KF was performed. All internal communication between the computers and the UAV control loop was carried out using IEEE 802.3 (Ethernet) connections [9].

During the trajectory, 12 types of uncertainty data were collected, which are latitude and longitude – obtained by the inertial sensor, 3 angular rates, 3 accelerometer measurements, directions: north, west, climb, and height measurements. This information, together with the image processing estimation, was applied to a KF-based data fuser to estimate the UAV position [9]. These data were stored in the flight logs in a tabular structure with 66,483-time samples.

The performance between data fuser strategies based on Kalman filter and self-configured neural network was evaluated comparing the estimated UAV positioning with the differential GPS real time Kinematic (GPS-RTK) sensor signal. Neto et al. [24] showed errors for the UAV position for each coordinates (Latitude, Longitude)  $\leq 1.5 \times 10^{-5}$ .

---

<sup>1</sup>See: Yamaha Motorsports: Yamaha R-MAX model – webpage accessed at 23/Nov/2020:  
[www.yamahamotorsports.com/motorsports/pages/precision-agriculture-rmax](http://www.yamahamotorsports.com/motorsports/pages/precision-agriculture-rmax).

## 4 TRAINING AND VALIDATION

Before starting the description of the data selection steps, training and validation of the neural fuser, it is important to inform a reader that throughout all the experiments presented here, the uncertainties were calculated on the difference between the results by the self-configured neural network and those obtained by a reference source - here the reference source is from the KF. This KF was developed by the WITAS research group.

Once the data from the WITAS research group were used, where the reference for the Swedish experiments was given by the INS data, the average error in the trajectory was 10 m, between the latitude and longitude measurements. Data fusion made by KF, combining INS measurements and image processing estimation, reduced this difference to less than 5 m [9].

Sensors embedded in the UAV have different acquisition rates. There are several reasons for this lack of synchronization. For example, the communication channels used in the embedded electronics can vary in type and technology, each with its own transfer rate. This variation in acquisition rates also occurred with data from WITAS [30].

Since the sensors had different acquisition rates, the data collection time were disunited. One of the metadata in the flight log is the time the information recorded by the sensor, where the time is measured in milliseconds. The embedded KF in the UAV used in Sweden had an activation rate of 50Hz. But the gyroscope and accelerometer sensors had an acquisition rate of 200Hz. Therefore, the measurements obtained by the system are repeated at various times. So, it was necessary to pre-select the data to interpolate the measurements obtained by the sensors so that they would be related temporally.

The original data were organized in 66,483 temporal samples, based on the sensor with the lowest acquisition rate (10Hz), and a pre-selection was made on the data. This pre-selection meant neglecting mismatched time measurements considering the sensor with the lowest acquisition rate. The data were reorganized into a set of 3,319 temporal samples. These data were used for training and validation of the neural fuser. In the experiments, 12 types of inertial data were used, which are latitude and longitude obtained by the inertial sensor, 3 angular rates, 3 accelerometer measurements, north, west, uphill, and last height measurements.

Here an MPCA implementation was used for the automatic configuration of the neural network [3]. Parameters to be determined for a neural network are the variation in the learning rate of the neural network, the variation in the number of training periods, the number of training iterations, the moment rate, types of the activation function, number of hidden layers, and the number of neurons per hidden layer. For the number of neurons, the selected variation was from 1 to 32 neurons – this number was selected after evaluations of some FPGA devices used in previous studies [23].

#### 4.1 Cross-validation training - hold out

Regarding the data made available by the WITAS research group, it was necessary a division into sets for training an artificial intelligence model. In the literature, there is a study that investigated the influence of the number of training periods on this data set. The study points out the methodology applied in the investigation was not an MLP but an approach using a Fuzzy system [26]. The investigation concludes that a low number of training iterations, less than 50 iterations, for example, does not have statistical security to guarantee quality in the estimation. In addition, the amount of training data used influences the results, indicating the best results were in the range between 60% and 80% of the data.

In the MPCA neural network configuration, the data partition were into two datasets: 90% of the data for training, and 10% for generalization. For the training dataset, an amount of 20% employed to cross validation. The performance errors with the network topology found by the MPCA were 10% of the data.

For the first experiment, the interval for the learning rates and moment was from 0.1 to 0.9. The number of epochs varied from 100 to 300 epochs, and the hyperbolic tangent and sigmoid activation functions were tested. The network outputs are real values for latitude and longitude. The information in Table 1 are the parameters values found by the MPCA, which presented the best result for the generalization set.

Table 1: Parameters selected by the MPCA for the first experiment.

Parameter	Value or Type
Training epoch	300
Neurons in the hidden layer	11
Learning Rate	0.7
Moment rate	0.8
Activation function	sigmoid

The errors found for the latitude and longitude coordinates in this experiment were 47.57 cm and 62.83 cm, respectively. Table 2 presents the uncertainties found for the generalization set at each coordinate. The average error in the trajectory was estimated at 87.38 cm – the actual displacement is considered here: the value obtained from the estimates by the KF. Regarding the trajectory error, it must be considered that each location in space is defined by a pair (latitude, longitude). Thus, there is a distance associated with two points, which in the case of this experiment corresponds to the difference in the actual displacement and the displacement estimated by the neural fuser. For a trajectory, there is a set of distances between the real coordinates and the estimated coordinates, with the trajectory error the average of these distances.

In this experiment, high values were obtained for standard deviation and variance. This can be indicative of sensitivity to noise. For latitude and longitude measurements, a variation in degrees,

Table 2: Uncertainties in centimeters for each coordinate.

Latitude	
Uncertainty type	Value
Mean error	47.57 cm
Standard deviation	45.74 cm
Variance	2,092.44 cm
Longitude	
Uncertainty type	Value
Mean error	62.83 cm
Standard deviation	49.46cm
Variance	2,447.09 cm

although small, can also be expressed in meters. For example, the distance of one degree in latitude is approximately equivalent to 111.12 km [33].

Figure 1 shows the result of the estimation applied to validation and generalization sets. It is important to note that the latitude and longitude values are in degree. This is expected for this type of data, where the variation usually occurs from the fifth decimal place, considering this small flight area. We note that the values in Table 2 are in centimeters, while values in Figure 1 are in degrees.

#### 4.2 Cross-validation training – k-fold

In the previous experiment, the data sets were separated sequentially in time. That is why the last 10% of the data has always been used in generalization. However, this way of dividing the data can be inefficient for training, since it is not known which part of the data is the most significant content. In other words, where is the part of the information is the best one to train the neural network. In addition, training a neural network using sequential data can generate learning that is addicted to trends, which increases the generalization error when the set applied for testing does not follow the same trend as the training set. To avoid this issue, it was decided to apply another partitioning approach known in the literature as k-fold [27].

The goal of k-fold cross-validation is to investigate the performance of a network based on the variation in training and generalization sets. Therefore, it is relevant for the validation of all k-sets to have the same neural network architecture, meaning the same number of hidden layers, the same number of neurons, the same activation function, and the same initial synaptic weights.

The result obtained from the k-fold approach showed improvements compared to previous experiment. Although the average error is close to that obtained in the first experiment, there was a reduction in the values of standard deviation and variance. The reduction in variance over latitude was 46%, for longitude it was 36%. Table 3 shows the uncertainties obtained for this cross-

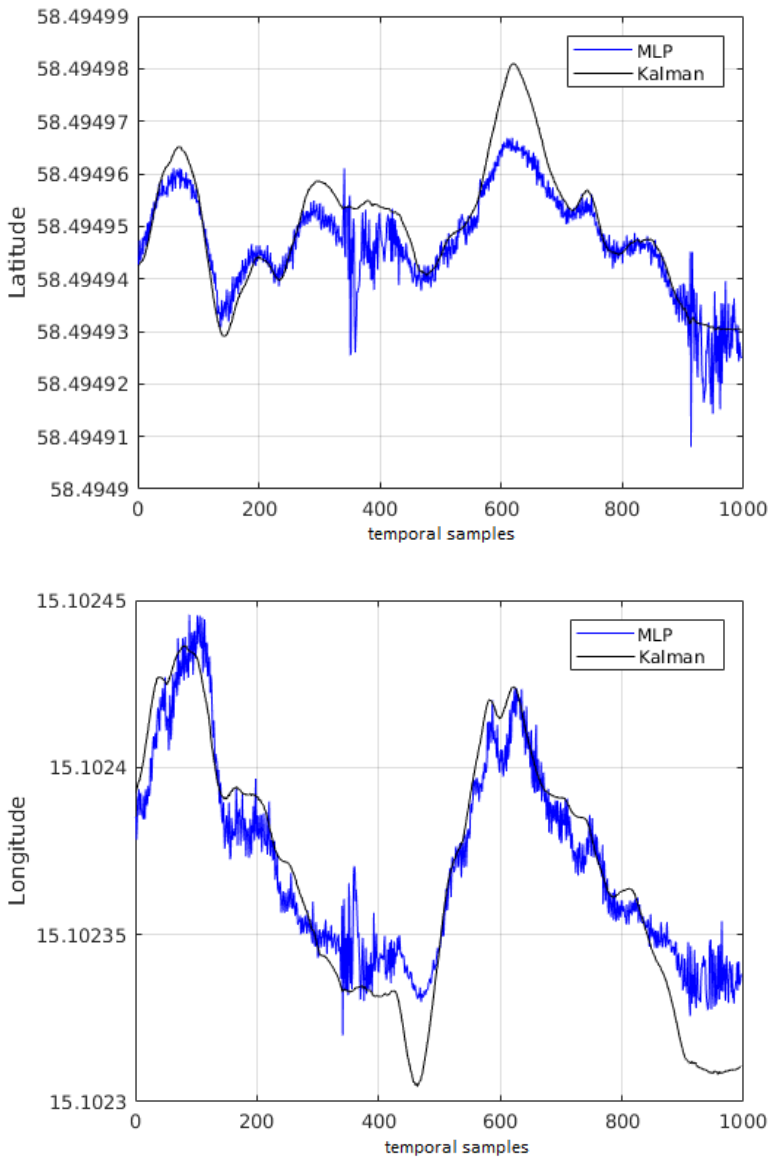


Figure 1: Result of the estimation on the validation and generalization sets.

validation approach. The reduction values were calculated on the results of the first experiment. For this comparison, the average of the average errors of each set was used.

In this experiment, original dataset has a division into 10 sets, and it is still considered the data acquisition sequence, where each set represents a time sequence of 10% of the data. The first set has the first 332 data, and so on.



Table 3: Uncertainties in centimeters for each coordinate - third experiment - k-fold.

Latitude		
Uncertainty type	Value	Uncertainty reduction
Mean error	43.92 cm	7.67%
Standard deviation	31.28 cm	31.61%
Variance	1,113.26 cm	46.79%
Longitude		
Uncertainty type	Value	Uncertainty reduction
Mean error	59.30 cm	5.61%
Standard deviation	37.53 cm	24.12%
Variance	1,562.86 cm	36.13%

As previously mentioned, the k-fold approach allows to identify which sets obtain the best and the worst result for generalization. In this way, the mean error and standard deviation were verified for each set. Figures 2 and 3 show the mean error and standard deviation of each set for both coordinates, respectively.

Some sets obtained higher mean error and standard deviation. Set-6 had the worst results for mean error and standard deviation, denoting a relevant portion of the data, and it would be interesting to leave this portion within the training set. Similar conclusion can be applied to the last two sets. It is also possible to see the sets with the best result for generalization. For example, set-3 obtained the smallest mean error for both coordinates and set-8 the smallest standard deviation considering the two coordinates. The last set, with 10% of the data, was used for generalization in the previous experiments, resulting in a bad influence for the first experiment. Therefore, if this set were in the data portion for training, the previous results could be better.

Table 3 shows a reduction in the values of standard deviation and variance. Obviously, these measures are related. The variance is information for the confidence interval. The confidence intervals can be high considering the context of UAV autonomous navigation.

Despite this experiment pointing out the sets with relevance, it has not yet been possible to determine which is the best combination among the sets, in order to have the best set for training. In addition, the way the data is organized directly affects the final result for generalization. More relevant, we are dealing with complex data, such as ones obtained during a UAV flight, making this a not trivial task. This shows the definition importance of the best data organization based on the available data.

Since the first experiment, the organization of the data continued using 70% for training, 20% for validation, and 10% generalization. These percentages are common in problems of supervised artificial intelligence models. But the first experiment, using the k-fold approach, did not perform all possible combinations between the data sets. The experiment did not consider the possible combinations among the validation and generalization sets – which add up to 30% of the data.

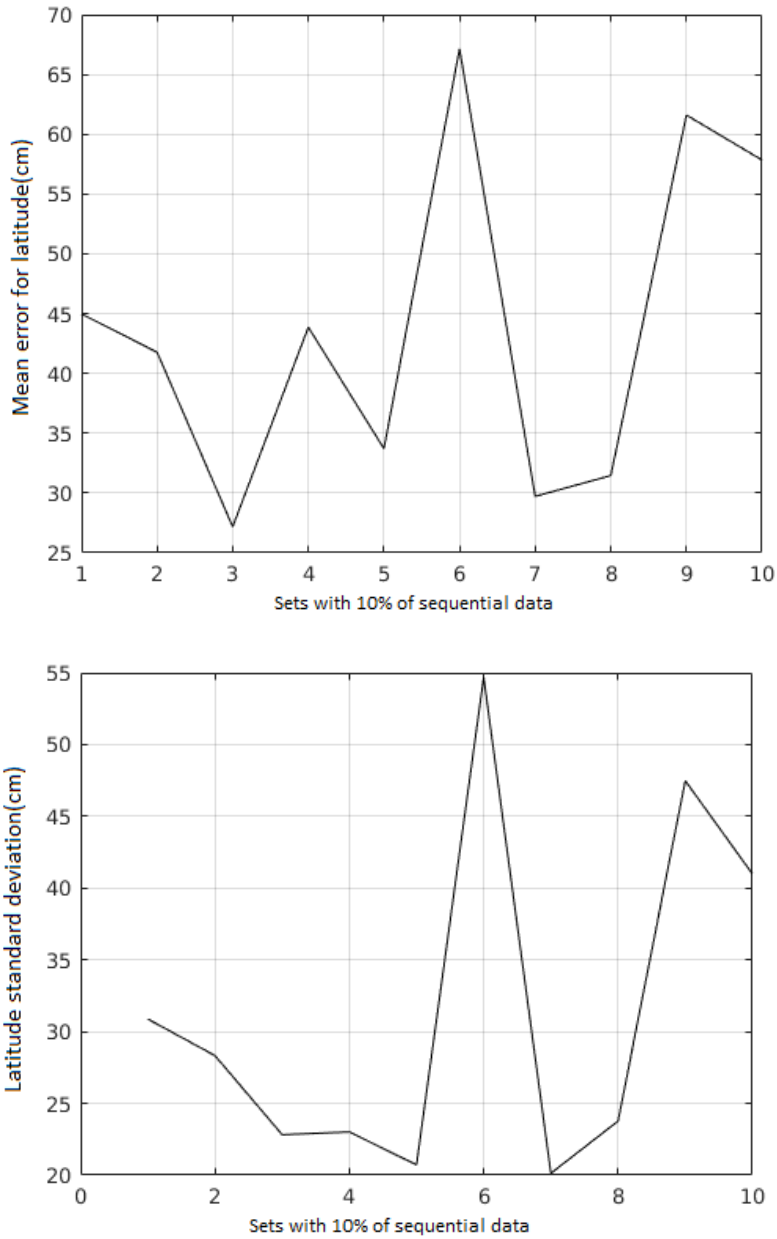


Figure 2: Mean error and standard deviation in latitude for each  $k$ -set.

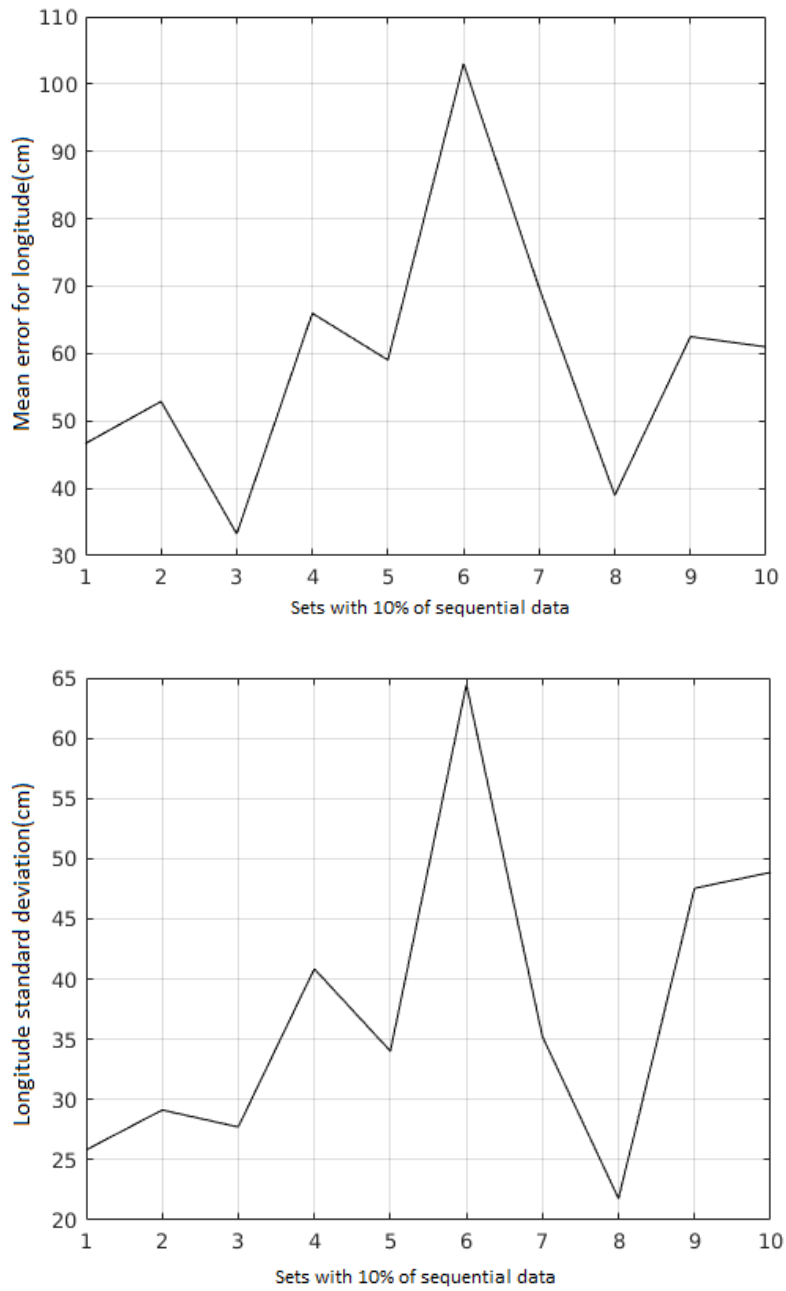


Figure 3: Mean error and standard deviation in longitude for each  $k$ -set.

Thus, another experiment was performed with the data, following the scheme of grouping the data by combining 10 taken 3 to 3. Thus, the percentage of 70% for training was maintained, but varying the combination of the 30% used in the cross-validation and generalization sets, totaling 120 possible and independent combinations.

The results obtained using this approach revealed a combination of data presenting the lowest mean error in the generalization is in set-75. For this set, the lowest mean error in the estimates for the UAV trajectory, and also the smallest errors for estimating latitude and longitude. Figures 4 and 5 respectively show the average errors in the estimates made for each set, for both coordinates and for trajectory. There is a range of sets, starting above set-60 and ending at set-80, which provides the best results – the smallest average errors. For latitude, the best result is not in this range, but in a set close to the last one shown in Figure 4. However, this is not reflected in the other coordinate or the trajectory error. So, this value was not considered for a possible classification, in case it is necessary to list the sets to choose one of them as the best or most appropriate.

Considering the first experiment, before using the k-fold approach, the mean error in the trajectory estimation was 87.28 cm. In the approach with 120 sets, for set-75, the mean error decreased to 69.41 cm. The reduction was 17.97 cm, representing a mean error of 20% less. The same happened with the coordinates in the first experiment, where the mean errors for latitude and longitude estimation were, respectively, 47.57 cm and 62.83 cm. For set-75, the mean errors were 35.74 cm and 53.73 cm. The result for set-75 represents a reduction of 24% and 14% for the mean error of estimating latitude and longitude.

From the average values obtained for each set, the mean estimation error for the trajectory would be 104.04 cm. The mean estimation error for coordinates would be 51.65 cm for latitude and 81.68 cm for longitude. Values were considerably higher than those obtained in set-75, and even higher than those obtained in the first experiment, before the k-fold approach. This clearly shows the impact of the partition in the data sets selected to compose the training, cross-validation, and generalization sets.

## 5 CONCLUSIONS AND FINAL CONSIDERATIONS

The experiments carried out aimed at a better understanding of the data set partition with a focus on the application of neural networks for UAV autonomous navigation. The experiments also served to verify the functioning of the methodology without or with the minimal treatment of the data.

The experiments showed the drift error from the INS sensor. Applying the k-fold approach, There is a data partition used for the training, which reflects an improvement in the generalization results.

Figure 4 shows that the last sets present the average estimation error for the longest trajectory. These last sets use the data of the end of the trajectory traversed by the UAV for generalization. Consequently, it becomes interesting in the training set to always put data that reflects the drift error. Therefore, the step using k-fold was important to find the sets where learning is improved.

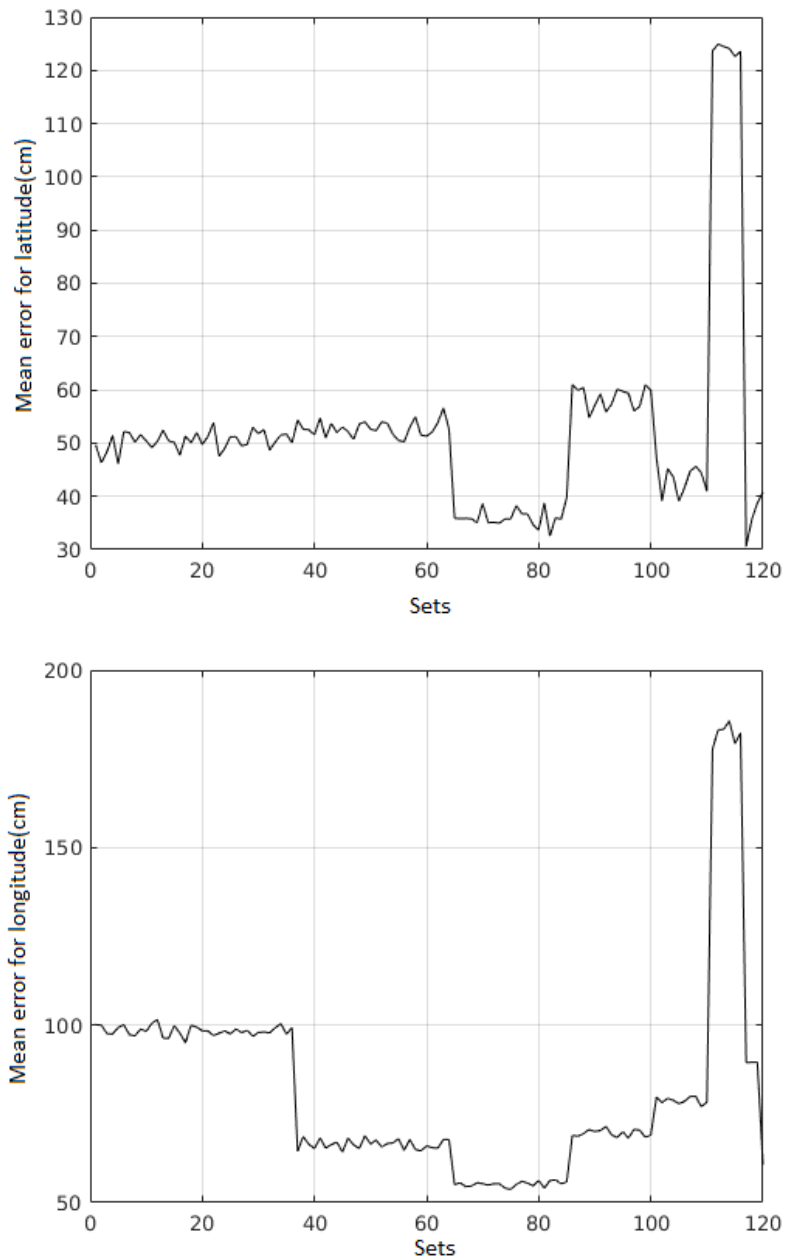


Figure 4: Estimates for each coordinate of each  $k$  set, following the scheme of 120 combinations.

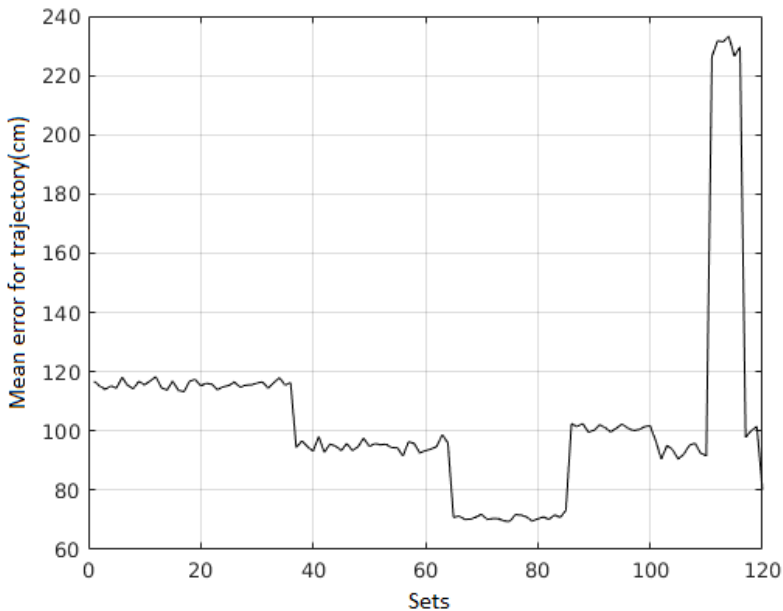


Figure 5: Estimates for the trajectory of each set of  $k$ , following the scheme of 120 combinations.

It is worth remembering that 12 types of inertial data were applied to data fusion and that each data has its contribution in terms of influence on the organization of training, cross-validation, and generalization data. Mainly, sensors such as gyroscope and accelerometer, present high sensitivity to any variation in the attitude of the aircraft during the flight, which can reflect in the performance during the training of the neural network. Still, on attitude sensors, it should be noted that their accuracy or sensitivity depends on the components used in the manufacture of the UAV.

Regarding the number of training seasons, the number of iterations between 100 and 300 was based on observations on preliminary tests. The results of the trajectory estimation tended to be equal to or greater than 1 m, for both coordinates, when the number of seasons was less than 100. The number of seasons for training with seasons above 300 has a worsening in the results, showing overtraining.

Evaluations on the impact on the number of neurons and/or hidden layers have not been analyzed, because the MPCA methodology was applied. This methodology already delivers an optimized architecture. Therefore, it was not necessary to investigate these parameters.

Experiments have shown the impact on how training, validation, and generalization sets are chosen. In addition, something important is the sensitivity to noise. Therefore, in the next work, the impact caused by noise will be evaluated, mainly in the confidence interval of the estimates.

## REFERENCES

- [1] S. Adusumilli, D. Bhatt, H. Wang, V. Devabhaktuni & P. Bhattacharya. A novel hybrid approach utilizing principal component regression and random forest regression to bridge the period of GPS outages. *Neurocomputing*, (2015).
- [2] A. Al-Kaff, D. Martín, F. García, A. Escalera & J. María Armingol. Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*, (2018).
- [3] J. Anochi, H.F. Campos Velho, H. Furtado & E.P. Luz. Self-configuring Two Types of Neural Networks by MPCA. *Journal of Mechanics Engineering and Automation*, (2015).
- [4] J.A. Anochi & H.F. Campos Velho. Neural network for seasonal climate precipitation prediction on the Brazil. *Ciência e Natura*, **42**(0) (2020), 15.
- [5] M.P. Arthur. Detecting Signal Spoofing and Jamming Attacks in UAV Networks using a Lightweight IDS. *International Conference on Computer, Information and Telecommunication Systems (CITS)*, (2019).
- [6] K. Bahador, K. Alaa, F.O. K. & S.N. R. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, **14**(1) (2013).
- [7] X. Chen, Z. Xuelong, Z. Wang, M. Li, Y. Ou & S. Yufan. Multi-Frequency Data Fusion for Attitude Estimation Based on Multi-Layer Perception and Cubature Kalman Filter. *IEEE Access*, (2020).
- [8] R.S. Cintra, H.F. Campos Velho & R. Todling. Redes Neurais Artificiais na Melhoria de Desempenho de Métodos de Assimilação de Dados: Filtro de Kalman. *TEMA Tendências em Matemática Aplicada e Computacional*, (1) (2010), 29–39.
- [9] G. Conte & P. Doherty. Vision-based unmanned aerial vehicle navigation using geo-referenced information. *Journal on Advances in Signal Processing*, (2009).
- [10] C. Corrado & K. Panetta. Data fusion and unmanned aerial vehicles (UAVs) for first responders. In “IEEE International Symposium on Technologies for Homeland Security (HST)” (2017).
- [11] H. Daryanavard & A. Harifi. Implementing Face Detection System on UAV Using Raspberry Pi Platform. *Iranian Conference on Electrical Engineering*, (2018).
- [12] J.P. Dash, M.S. Watt, G.D. Pearse, M. Heaphy & H.S. Dungey. Assessing very high resolution UAV imagery for monitoring forest health during a simulated disease outbreak. *ISPRS Journal of Photogrammetry and Remote Sensing*, (2017).
- [13] E. Elezi, G. Cankaya, A. Boyaci & S. Yarkan. The effect of Electronic Jammers on GPS Signals. *16th International Multi-Conference on Systems, Signals Devices (SSD)*, (2019).
- [14] X. Gao, H. Jia, Z. Chen, G. Yuan & S. Yang. UAV security situation awareness method based on semantic analysis. *IEEE International Conference on Power, Intelligent Computing and Systems*, (2020).
- [15] C.M. Gevaert, J. Tang, F.J. García-Haro, J. Suomalainen & L. Kooistra. Combining hyperspectral UAV and multispectral Formosat-2 imagery for precision agriculture applications. *6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, (2014).

- [16] M. Grewal, A. Andrews & C. Bartone. “Global Navigation Satellite Systems, Inertial Navigation, and Integration”. Wiley (2019).
- [17] I. Gulati, H. Li, S. Stainton, M. Johnston & S. Dlay. Investigation of Ionospheric Phase Scintillation at Middle-Latitude Receiver Station. In “International Symposium ELMAR” (2019).
- [18] W.C. Jin, K. Kim & W.J. Choi. Robust Jamming Algorithm for Location-Based UAV Jamming System. *IEEE Asia-Pacific Microwave Conference (APMC)*, (2019).
- [19] A. Khaldi, C. Bensalah, A.C. Braham & B. Cherki. UAV Attitude Estimation using Visual and Inertial Data Fusion based on Observer in SO(3). In “International Conference on Advanced Electrical Engineering (ICAEE)” (2019).
- [20] X. Kong, G. Fang, L. Liu & T.P. Tran. Low Computational Data Fusion Approach Using INS and UWB for UAV Navigation Tasks in GPS-Denied Environments. In “20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)” (2019).
- [21] E.F.P. Luz, J.C. Becceneri & H.F.d. Campos Velh. A new multi-particle collision algorithm for optimization in a high performance environment. *Journal of Computational Interdisciplinary Sciences*, (1) (2008), 3–10. URL <https://www.doi.org/10.6062/jcis.2008.01.01.0001>.
- [22] H. Mei, H. Changyi & W. Guangming. Novel method to evaluate the effectiveness of UAV navigation spoofing. *IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, (2019).
- [23] G.P. Neto, H.F. Campos Velho & E.H. Shiguemori. UAV autonomous navigation by data fusion and FPGA. *Mecânica Computacional*, **37** (2019), 609–618.
- [24] G.P. Neto, A.C. Paulino, H.F. Campos Velho, L.N.F. Guimarães & E.H. Shiguemori. Computational intelligence for UAV autonomous navigation with multisensor data fusion. In “Proceedings of the Conference of Computational Interdisciplinary Science (CCIS 2019)” (2019). URL <https://proceedings.science/ccis-2019/papers/computational-intelligence-for-uav-autonomous-navigation-with-multisensor-data-fusion>.
- [25] Z. Nian & C. Jung. CNN-Based Multi-Focus Image Fusion with Light Field Data. In “IEEE International Conference on Image Processing (ICIP)” (2019).
- [26] A. Paulino, L. Guimarães & E.H. Shiguemori. Hybrid Adaptive Computational Intelligence-based Multisensor Data Fusion applied to real-time UAV autonomous navigation. *INTELIGENCIA ARTIFICIAL*, **22** (2019), 162–195.
- [27] A. Rabinowicz & S. Rosset. Cross-Validation for Correlated Data. *Journal of the American Statistical Association*, (2020).
- [28] V.P. Subba Rao & S.G. Rao. Design and Modelling of an Affordable UAV Based Pesticide Sprayer in Agriculture Applications. *Fifth International Conference on Electrical Energy Systems*, (2019).
- [29] X. Sun, Z. Zhang, Y. Ji, S. Yan, W. Fu & Q. Chen. Algorithm of Ionospheric Scintillation Monitoring. In “7th International Conference on Digital Home (ICDH)” (2018).
- [30] K.P. Valavanis & G.J. Vachtsevanos. “Handbook of Unmanned Aerial Vehicles”. Springer Publishing Company, Incorporated (2014).



- [31] Y. Yamazaki, M. Tamaki, C. Premachandra, C.J. Perera, S. Sumathipala & B.H. Sudantha. Victim Detection Using UAV with On-board Voice Recognition System. *Third IEEE International Conference on Robotic Computing*, (2019).
- [32] X. Yang, D. Lin, F. Zhang, T. Song & T. Jiang. High Accuracy Active Stand-off Target Geolocation Using UAV Platform. *IEEE International Conference on Signal, Information and Data Processing*, (2019).
- [33] Z. Yingkun. Flight path planning of agriculture UAV based on improved artificial potential field method. In “Chinese Control And Decision Conference” (2018).
- [34] W. You, F. Li, L. Liao & M. Huang. Data Fusion of UWB and IMU Based on Unscented Kalman Filter for Indoor Localization of Quadrotor UAV. *20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, **8** (2020).
- [35] S. Yousuf & M.B. Kadri. Robot Localization in Indoor and Outdoor Environments by Multi-sensor Fusion. In “14th International Conference on Emerging Technologies (ICET)” (2019).
- [36] B. Zhang & C. Xu. Research on UAV Attitude Data Fusion Algorithm Based on Quaternion Gradient Descent. In “International Conference on Communications, Information System and Computer Engineering (CISCE)” (2019).

