

Assessment of global and local neural network's performance for model-free estimation of flow angles

Original

Assessment of global and local neural network's performance for model-free estimation of flow angles / Lerro, A.; de Pasquale, L.. - In: THE AERONAUTICAL JOURNAL. - ISSN 0001-9240. - (2023), pp. 1-16. [10.1017/aer.2023.55]

Availability:

This version is available at: 11583/2981785 since: 2023-09-08T09:40:14Z

Publisher:

CAMBRIDGE UNIV PRESS

Published

DOI:10.1017/aer.2023.55

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

REGULAR PAPER

Assessment of global and local neural network's performance for model-free estimation of flow angles

A. Lerro  and L. de Pasquale 

Department of Mechanical and Aerospace Engineering, Polytechnic University of Turin, C.so Duca degli Abruzzi 24 Turin, Italy, 10129

Corresponding author: A. Lerro; Email: angelo.lerro@polito.it

Received: 27 February 2023; **Revised:** 31 May 2023; **Accepted:** 6 June 2023

Keywords: Synthetic sensors; Flow angle; Neural network; Model-free

Abstract

A synthetic flow angle sensor, able to estimate angle-of-attack and angle-of-sideslip, can exploit different methods to solve a set of equations modelling data fusion from other onboard systems. In operative scenarios, measurements used for data fusion are characterised by several uncertainties that would significantly affect the synthetic sensor performance. The off-line use of neural networks is not a novelty to model deterministic synthetic flow angle sensors and to mitigate issues arising from real flight applications. A common practice is to train the neural network with corrupted data that are representative of uncertainties of the current application. However, this approach requires accurate tuning on the target aircraft and extensive flight test campaigns, therefore, making the neural network tightly dependent on the specific aircraft. In order to overcome latter issues, this work proposes the use of neural networks to solve a model-free scheme, derived from classical flight mechanics, that is independent from the target aircraft, flight regime and avionics. It is crucial to make use of a training dataset that is not related to any specific aircraft or avionics to preserve the generality of the scheme. Under these circumstances, global and local neural networks are herein compared with an iterative method to assess the neural capabilities to generalise the proposed model-free solver. The final objective of the present work, in fact, is to select the neural technique that can enable a flow angle synthetic sensor to be used on board any flying body at any flight regime without any further training sessions.

Nomenclature

ADS	air data system
ADHRS	air data, heading reference system
AoA	angle-of-attack
AoS	angle-of-sideslip
ASSE	angle-of-attack and -sideslip estimator
EMRAN	extended minimal resource-allocating network
FCS	flight control systems
GRBF	generalised radial basis function
MLP	multilayer perceptron
MRAN	minimal resource-allocating network
NN	neural network
RAN	resource-allocating network
RBF	radial basis function
TAS	true airspeed
UAV	unmanned aerial vehicles
UAS	unmanned aircraft systems
VMS	vehicle management systems

1.0 Introduction

Redundancy is usually applied to safety critical systems to avoid single points of failure and to be compliant with the applicable airworthiness regulations. Analytical redundancy [1] can guarantee the same level of reliability without installing any additional (or redundant) physical systems. The whole system can be also monitored by means of using analytical, or synthetic, sensors with the aim, for example, to accommodate possible failures [2, 3].

A synthetic sensor implements a data fusion technique that can provide indirect measurements of flight data. The use of synthetic sensors can be beneficial to reduce the number of system units to be installed on the aircraft and to introduce the dissimilarity feature that is crucial to avoid common failure modes [4, 5] and incorrect failure diagnosis from pilots or autonomous systems [6].

Modern control systems rely on vehicle management systems (VMS), or flight control systems (FCS), to autonomously control, navigate and complete the air vehicle's mission. The VMS, or the FCS, encompasses several sub-systems in order to measure all necessary flight data to perform the required tasks. A set of flight data is directly measured from the external environment by means of the air data system (ADS) that is based on physical probes and vanes installed on the aircraft fuselage to be immersed in the surrounding airfield. The external sensors can provide direct measurements of pressures, temperatures and flow angles.

Flow angles, angle-of-attack (AoA) and angle-of-sideslip (AoS), can be estimated using several synthetic sensors [7–16] that can be grouped into two main categories, related to the approach used: model-based (e.g. Kalman filters) and data-based (e.g. neural networks). The latter approaches suffer from common disadvantages. In fact, they are designed on the target aircraft and, moreover, they need to be reconfigured for any change in aircraft configuration or flight regime. The latter drawbacks are overcome using a model-free approach.

In the present work, the adopted method for flow angle estimation aims to be independent from the target aircraft and avionics [17]. The proposed method, in fact, is based on a set of nonlinear equations derived from the theory of flight dynamics by means of fusing together several inputs: inertial accelerations, airspeed and angular rates. The nonlinear scheme, named ASSE, can be solved using iterative solvers that can lose accuracy in operative scenarios [18] dealing with actual signals that are affected by uncertainties [19, 20] due to several sources (electronics noise, structural vibrations, etc.).

The manufactured technological demonstrator encompasses several sensors (to provide direct measurements to the ASSE scheme) that are characterised [21]. The iterative method that is currently being used to solve the ASSE scheme may lose accuracy in realistic applications. Therefore, alternative solvers, such as neural ones, are considered in order to mitigate the effect of realistic uncertainties on the input signals. The multi-layer perceptron (MLP) and generalised radial basis function (GRBF) neural networks (NNs) are proposed in this work as possible solvers of the ASSE scheme. Both neural solvers are pre-trained in order to guarantee the deterministic feature. On the other side, training the neural networks with flight data from a specific aircraft would hinder the generality of the method. The objective of the present work is to investigate if it is possible to train the neural network as the ASSE solver using uncertainty-free inputs and, at the same time, preserving acceptable performance when using data corrupted by realistic uncertainties. With the proposed approach, the model-free synthetic sensor is maintained generic as it is fully decoupled from the target aircraft. In fact, modelling uncertainties, considering worst scenarios, would lead to pre-trained neural networks that are ready to be used as flow angle synthetic sensors on board any flying body at any flight regime without any further training sessions.

The ASSE scheme and the corresponding reliability criteria are introduced in Section 2. The characterisation study of the ASSE technological demonstrator is briefly reported in Section 3. The neural approach is justified in Section 4 along with the performance evaluation criteria and the description of training and test dataset. Neural network trainings are documented in Section 5. Before concluding the work, results are presented in Section 6 comparing the three methods.

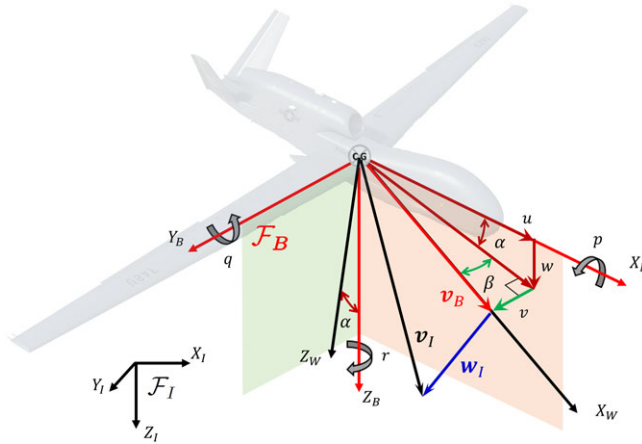


Figure 1. Representation of inertial and body reference frames with positive flow angles (α, β), linear relative velocities (u, v, w), angular rates (p, q, r), inertial velocity \mathbf{v}_I , relative velocity \mathbf{v}_B and wind velocity \mathbf{w}_I .

2.0 Nonlinear ASSE scheme

In this section, some crucial information is introduced about the necessary inputs of the ASSE scheme.

In Fig. 1, the inertial and body reference frames, $\mathcal{F}_I = \{X_I, Y_I, Z_I\}$ and $\mathcal{F}_B = \{X_B, Y_B, Z_B\}$ respectively, are illustrated. The subscript denotes the reference frame where the vector components are represented. Generally speaking, the inertial velocity \mathbf{v}_I can be expressed as a function of the relative velocity \mathbf{v}_B and the wind velocity \mathbf{w}_I as

$$\mathbf{v}_I = \mathbf{C}_{B2I} \mathbf{v}_B + \mathbf{w}_I \tag{1}$$

where $\mathbf{C}_{B2I} = \mathbf{C}_{I2B}^T$ is the direction-cosine matrix to express vector components in the inertial reference frame from the body one [22]. The vector \mathbf{v}_B can be written as a function of TAS and the flow angles, α and β , as

$$\mathbf{v}_B = V_\infty \hat{\mathbf{i}}_{WB} = V_\infty (\cos \beta \cos \alpha \hat{\mathbf{i}}_B + \sin \beta \hat{\mathbf{j}}_B + \cos \beta \sin \alpha \hat{\mathbf{k}}_B) \tag{2}$$

where V_∞ is the magnitude of the true airspeed (TAS), $\hat{\mathbf{i}}_B, \hat{\mathbf{j}}_B$ and $\hat{\mathbf{k}}_B$ are the three unit vectors defining \mathcal{F}_B and $\hat{\mathbf{i}}_{WB}$ is the unit vector of \mathbf{v}_B whose direction is only related to α and β .

From the definition of the skew-symmetric matrix of body angular rates $\mathbf{\Omega}_B$ [23], the coordinate acceleration can be written as

$$\mathbf{a}_B = \mathbf{C}_{I2B} \mathbf{a}_I = \dot{\mathbf{v}}_B + \mathbf{\Omega}_B \mathbf{v}_B + \mathbf{C}_{I2B} \dot{\mathbf{w}}_I = a_{X_B} \hat{\mathbf{i}}_B + a_{Y_B} \hat{\mathbf{j}}_B + a_{Z_B} \hat{\mathbf{k}}_B \tag{3}$$

It is worth noting that the coordinate acceleration vector \mathbf{a}_B can be calculated by means of removing the effect of the gravity acceleration from the inertial acceleration \mathbf{a}_I that is directly measured by inertial reference systems.

The fundamental equation of the ASSE scheme [17] can be introduced as

$$n_\tau = \hat{\mathbf{i}}_{WB,t}^T \mathbf{m}_\tau \tag{4}$$

where t is the current time, τ is a generic previous time, and

$$n_\tau = V_{\infty,\tau} \dot{V}_{\infty,\tau} + \left[\int_\tau^t \mathbf{a}_B d\mathcal{T} - \int_\tau^t \mathbf{C}_{I2B} \dot{\mathbf{w}} d\mathcal{T} \right]^T (\mathbf{a}_B - \mathbf{C}_{I2B} \dot{\mathbf{w}})_\tau \tag{5}$$

and

$$\mathbf{m}_\tau = V_{\infty,t} (\mathbf{I} - \mathbf{\Omega}_{B,t} \Delta t) (\mathbf{a}_B - \mathbf{C}_{I2B} \dot{\mathbf{w}})_\tau = h_\tau \hat{\mathbf{i}}_B + l_\tau \hat{\mathbf{j}}_B + m_\tau \hat{\mathbf{k}}_B \tag{6}$$

It is worth underlining that in Equations (5) and (6) all parameters are measured between time τ and t . Whereas, in Equation (4), AoA and AoS only appear at current time t . Moreover, Equation (4) can be rewritten considering different starting time τ_i with $i \in [0, 1, \dots, n]$ and $\tau_0 \equiv t$. The latter process leads to the ASSE scheme of $n + 1$ nonlinear equations:

$$\begin{cases} n_t = \hat{\mathbf{i}}_{WB,t}^T \mathbf{m}_t \\ n_{\tau_1} = \hat{\mathbf{i}}_{WB,t}^T \mathbf{m}_{\tau_1} \\ \vdots \\ n_{\tau_n} = \hat{\mathbf{i}}_{WB,t}^T \mathbf{m}_{\tau_n} \end{cases} \quad (7)$$

where α and β (at current time t) are the only unknowns of the system. The wind acceleration $\dot{\mathbf{w}}$ is considered negligible in the time interval and all other coefficients can be directly measured using dedicated systems (as described in Section 3). Therefore, the flow angle estimation of Equation (7) requires direct measurements of: (1) body angular rates p , q and r ; (2) TAS and its time derivative; (3) coordinate acceleration vector \mathbf{a}_B .

2.1 ASSE solvers

The flow angle estimation approach of Equation (7) can be solved using the most suitable solver for nonlinear schemes. The Levenberg-Marquardt algorithm [24] is used in Ref. [17] to solve the system of two, three and four nonlinear equations. In Ref. [18], it is shown that the iterative solver is highly affected by uncertainties on the inputs and neural solvers can mitigate the propagation of input uncertainties on the flow angle estimations.

Three nonlinear equations are used to assemble the ASSE scheme in the present work. The same number of equations is used to build the input layer of the neural networks as described in Section 2.1.

2.2 ASSE reliability criteria

In Ref. [25], the flight conditions where the ASSE scheme is reliable are defined as:

1. Vertical/lateral coordinate acceleration:
 - $a_z > a_{thr} = 0.5 \text{ ms}^{-2}$ for AoA estimation
 - $a_y > a_{thr} = 0.5 \text{ ms}^{-2}$ for AoS estimation
2. $\tilde{D} > D_{thr} = 0.2 \text{ m}^4/\text{s}^6$

where $\tilde{D} = l_{t,\dot{\mathbf{w}} \approx 0} m_{\tau,\dot{\mathbf{w}} \approx 0} - m_{t,\dot{\mathbf{w}} \approx 0} l_{\tau,\dot{\mathbf{w}} \approx 0}$. In order to guarantee the flow angle estimation during dynamic manoeuvres, the aforementioned criteria are considered satisfied only if they are verified for (at least) 100 consecutive samples, equivalent to 1 s at the output rate adopted in this work (as described in Section 3).

3.0 Uncertainty characterisation

A dedicated demonstrator of the ASSE technology is designed and manufactured, as illustrated in Fig. 2, that encompasses an air data, inertial and heading reference system (ADAHRS).

The technological demonstrator can provide all the required measurements for the ASSE scheme at 100 Hz. The demonstrator is interfaced with: (i) a Pitot probe able to sense the dynamic and absolute pressures and (ii) an outside ambient temperature sensor. All sensors are characterised in terms of global uncertainty under static and dynamic condition (21). The sensor's uncertainty is modelled using a bias (B_0) and white noise (WN) with null mean and the standard deviation 1σ is evaluated as a function

Table 1. Uncertainty models for the ASSE technological demonstrator

Sensor	B_0	$\sigma_{WN,0}$	$\sigma_{WN,1}$
gyroscope	0° s^{-1}	$0.05^\circ \text{ s}^{-1}$	5×10^{-4}
accelerometer	0 ms^{-2}	0.07 ms^{-2}	0.02
true airspeed	$\pm 0.47 \text{ ms}^{-1}$	$1.3 \times 10^{-3} \text{ ms}^{-1}$	0
TAS time derivative	$\pm 0 \text{ ms}^{-2}$	0.073 ms^{-2}	0.4

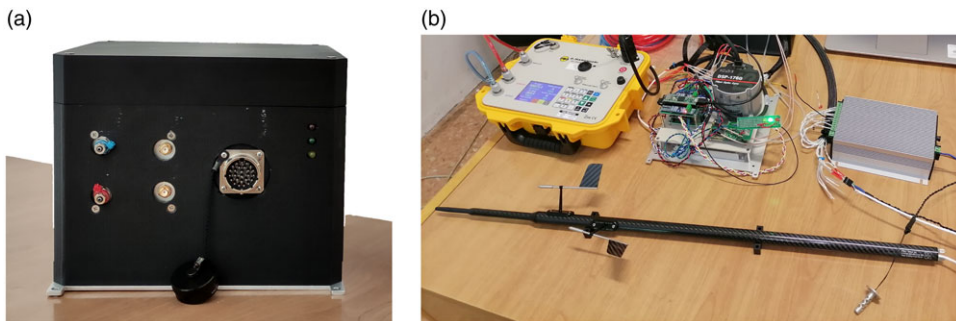


Figure 2. A view of the boxed (a) un-boxed (b) technological demonstrator under pressure tests. The demonstrator is connected to the Pitot boom with flow angle vanes, the outside ambient temperature sensor and the power supply.

of the current value itself. Therefore, the sensor uncertainty is modelled as $\pm B_0 + WN$, where $\sigma_{WN} = \frac{1}{2} \sqrt{\sigma_{WN,0}^2 + v^2 \sigma_{WN,1}^2}$ and v is the current value of the signal. The uncertainty models used in the present work are collected in Table 1.

For example, the inertial acceleration along the X direction, a_x , is corrupted using an unbiased white noise with 1σ calculated as $a_{x,1\sigma} = \frac{1}{2} \sqrt{0.007^2 + a_x^2 0.02^2} \text{ ms}^{-2}$. Whereas, the TAS uncertainty is modelled with a bias $B_{0,TAS} = \pm 0.47 \text{ ms}^{-1}$ and a white noise with constant 1σ value, $TAS_{1\sigma} = 1.3 \times 10^{-3} \text{ ms}^{-1}$.

4.0 Selected neural approaches

It can be demonstrated that MLP [26] and GRBF [27] NNs with a single hidden layer can approximate any continuous nonlinear input-output mapping function. However, the MLP and GRBF represent two different neural approaches as they belong to “global” and “local” approximators, respectively.

As far as a “global” neural network is concerned, the activation function is part of the ridge function class (e.g. Equation (A2)) and, therefore, the output of a single hidden unit (or neuron) is calculated as a function of all the available inputs. On the other hand, the GRBF activation functions are radial ones (e.g. Equation (B2)) and each centre is committed to a specific area of the input space. In fact, the output of a single hidden unit computes the Euclidean norm (or distance) between the input vector and the neuron centre.

Although the relation between the MLP and GRBF is demonstrated in Ref. [28], the degree of accuracy between the MLP and GRBF approaches cannot be preliminarily defined. For example, in Ref. [29], the MLP is shown to achieve the same accuracy of the RBF neural networks but using less hidden units.

As the present work deals with a physics-based method solved using “global” and “local” neural networks, the input layer is defined by the ASSE scheme. Moreover, as the proposed GRBF-NN is defined with a single hidden layer, the proposed MLP-NN is also based on a single hidden layer.

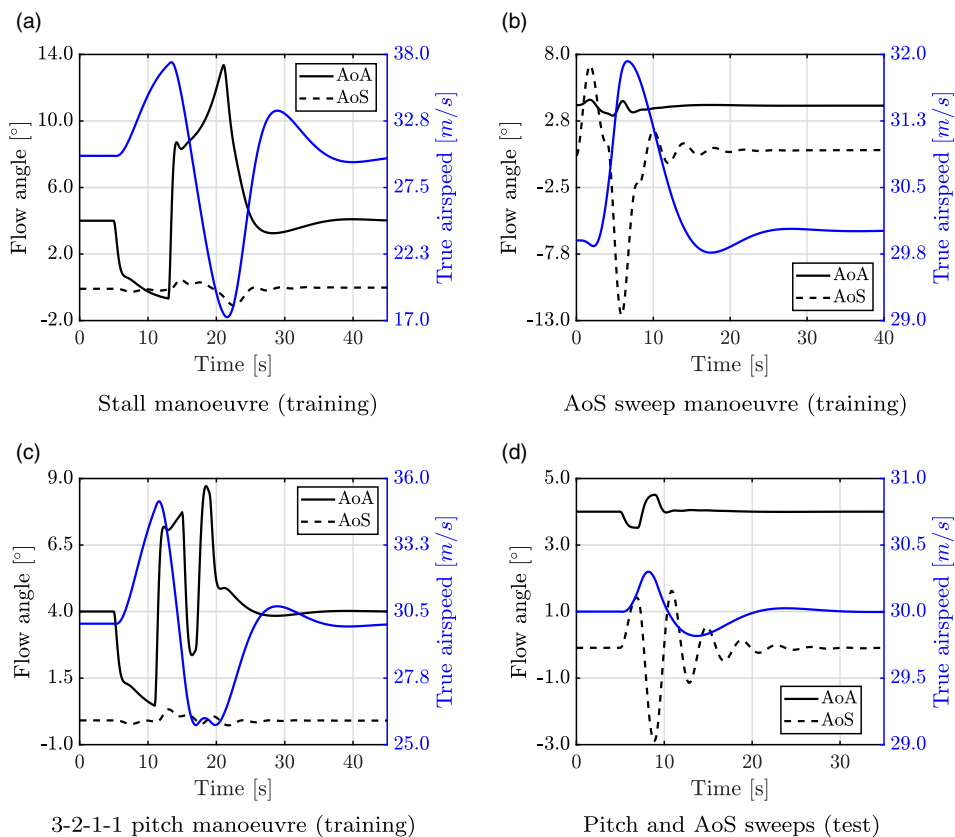


Figure 3. Training and test manoeuvres.

4.1 Training and test dataset

The definition of a suitable training dataset can be challenging. As far as model-free methods are concerned, a possible strategy is to generate coherent uncertainty-free data using a reliable flight simulator. According to the latter potential strategy, it would be possible to control an adequate mapping of the domain where the neural networks are defined. However, building a suitable training dataset is not addressed in this work and the simulated flight manoeuvres only define a training and test dataset suitable for the scope of the work.

Simulated data are labelled as “nominal data” because they are not affected by any uncertainties. Simulated data are also corrupted using uncertainty models defined according to the characterisation study of a dedicated technological demonstrator. Latter data are labelled as “realistic data”. Firstly, the proposed networks are trained using nominal data, whereas they are tested using both nominal and realistic data. Secondly, the networks are re-trained using input data affected by realistic uncertainties and they are tested using both nominal and realistic data.

Training and test data are obtained with flight simulations and input uncertainties are artificially added in post processing according to Section 3. The flight simulator is a six-degree of freedom nonlinear aircraft model with nonlinear aerodynamic and thrust models. The simulation is executed with 10 ms fixed time steps aiming to be coherent with the output rate of the dedicated demonstrator.

Training and test manoeuvres considered in this work are introduced in [18] and available in Ref. [30]. Three manoeuvres (Figs. 3(a)–(c)) belong to the training dataset, whereas, the independent manoeuvre of Fig. 3(d) belongs to the test dataset conceived to evaluate the ability of the NNs to solve the ASSE scheme.

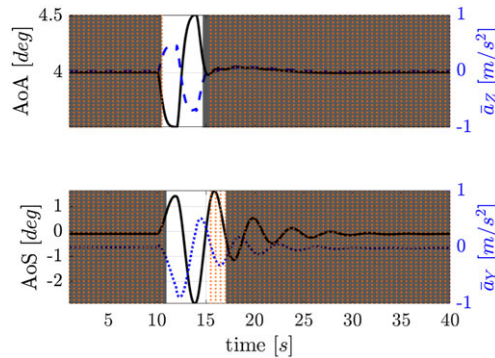


Figure 4. Test manoeuvre with ASSE reliability limits.

The training is made up of: (1) a stall manoeuvre (Fig. 3(a)) to excite the AoA up to maximum values under quasi-steady state flight conditions; (2) an AoS sweep manoeuvre (Fig. 3(b)) to excite lateral-directional modes; (3) 3-2-1-1 pitch manoeuvre (Fig. 3(c)) to excite longitudinal aircraft modes. The test manoeuvre, not related to the training dataset, is a pitch sweep superimposed to an angle-of-sideslip sweep manoeuvre to excite at the same time both longitudinal and lateral-directional aircraft modes (Fig. 3(d)).

The simulations only involve synchronised and noise-free signals. With the aim to preliminarily evaluate the ASSE performance with realistic input signals, the uncertainty models are used to corrupt the simulated flight data as introduced in Section 3.

The ASSE reliability criteria of Section 2.2 are applied to the test manoeuvre of Fig. 3(d) in order to limit the performance assessment during flight conditions that are reliable for ASSE applications. In Fig. 4, the shaded area represents the flight conditions where the acceleration criteria are not met, whereas, the dotted areas indicate where the determinant criteria is not satisfied. For simplicity, in this work, the acceleration criteria are only considered for the ASSE application. Therefore, the AoA will be evaluated in the time frame [10.0 s ÷ 14.5 s], whereas the AoS in [10.0 s ÷ 17.0 s]. It is important to underline that the ASSE solution is always computed even when the aforementioned criteria are not met in order to show the flow angle estimations outside the reliability area.

4.2 Performance evaluation criteria

As metrological standards for synthetic sensors are under evaluation [31], some criteria are defined in this work to evaluate the flow angle estimation. The mean, maximum, 1σ , 2σ and 3σ errors are used to conduct the performance analysis. In the present work, 1σ , 2σ and 3σ are the values such that the probability $\Pr(-\sigma \leq X \leq \sigma) = 68.3\%$, $\Pr(-2\sigma \leq X \leq 2\sigma) = 95.4\%$ and $\Pr(-3\sigma \leq X \leq 3\sigma) = 99.7\%$ also in case the error does not have a normal distribution.

The rationale behind the latter strategy is to compare the maximum and 3σ errors in order to evaluate the presence of possible spike errors. The 1σ and 2σ errors are those commonly used to assess the sensor's performance. From an operative standpoint, the maximum and the 2σ errors should be specified according to their target applications.

For example, if the synthetic flow angle sensor is used to display data on board or to monitor actual systems, maximum absolute errors up to 5° (or even 10°) can be accepted. Whereas, for analytical redundancy of UAS systems, e.g. VMS or FCS applications, the performance requirements can be more demanding. The latter scenario is assumed to define the following performance requirements:

- maximum absolute error $< 2^\circ$
- 2σ error $< 1^\circ$

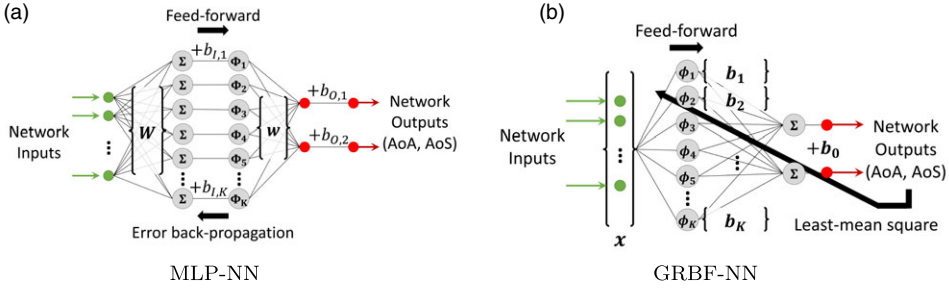


Figure 5. Overview of the neural networks with a single hidden layer considered in the present work.

5.0 Neural networks as ASSE solvers

As aforementioned, in the present work NNs are used as solvers of the ASSE scheme presented in Equation (7). The neural solvers are trained to learn the input-output nonlinear mapping between the ASSE inputs (\mathbf{m} and n coefficients) and the output (α and β). Exploiting the model-free feature of the ASSE scheme, the NNs can be trained independently from the target aircraft and, for the latter reason, using flight data obtained with a coherent data generator. Therefore, even though a particular flight simulator is used to generate training data, the trained NN is generic, i.e. independent from the aircraft model adopted in the simulations and applicable to any flying bodies.

The neural network input layer is shaped according to the ASSE scheme composed of three nonlinear equations. The latter choice is justified in [17] where the minimum set of equations is established. Considering Equation (7), the input layer, or vector, comprises the coefficients $([\mathbf{m}_t^T, \mathbf{m}_{\tau_1}^T, \mathbf{m}_{\tau_2}^T]^T)$ and known terms $([n_t, n_{\tau_1}, n_{\tau_2}]^T)$ at the current time t and two previous time steps, τ_1 and τ_2 . As the demonstrator’s operating frequency is 100 Hz, $\tau_1 = t - 10$ ms and $\tau_2 = t - 20$ ms. Therefore, recalling Equation (7), the overall network response is a mapping $f_{NN} : \mathbb{R}^{12} \rightarrow \mathbb{R}^2$ for flow angle estimation at current time t that can be written as

$$\tilde{\mathbf{y}} = [\tilde{\alpha}(t), \tilde{\beta}(t)]^T = f_{NN} \{ [\mathbf{m}_t^T, n_t, \mathbf{m}_{\tau_1}^T, n_{\tau_1}, \mathbf{m}_{\tau_2}^T, n_{\tau_2}]^T \} \tag{8}$$

where the neural networks of the present work have 12 inputs (n_i), 2 outputs (n_o) and the reference output is denoted as $\mathbf{y} = [\alpha(t), \beta(t)]^T$. The MLP and GRBF NNs are selected as examples of “global” and a “local” approximators respectively. A schematic overview is presented in Fig. 5.

As all presented neural networks are pre-trained on conventional personal computers, the operative life is expected to be performed on commercial hardware, e.g. for drone applications.

5.1 MLP trade-off

As the number of MLP-NN’s neurons is not defined a priori, a trade-off analysis is necessary to define the number of hidden units as presented in Fig. 6. The 3σ error is used because it will provide information about the largest errors excluding possible spike errors.

Once the number of hidden units is set, the MLP-NN is re-trained 10 times starting from random network weights. Among the ten re-trainings, the network that minimises the mean quadratic sum of the AoA and AoS 3σ errors is selected.

In order to define the best MLP-NN architecture, eight different neural networks are considered with several numbers of neurons ([10, 15, 20, 25, 30, 35, 40, 45]) as represented in Fig. 6. As far as the MLP-NN is considered, the best performance is achieved with 25 neurons (Fig. 6(a)). Conversely, the best performance is obtained with 30 hidden units for the MLP-NN-U (Fig. 6(b)). The latter configurations are used in this work.

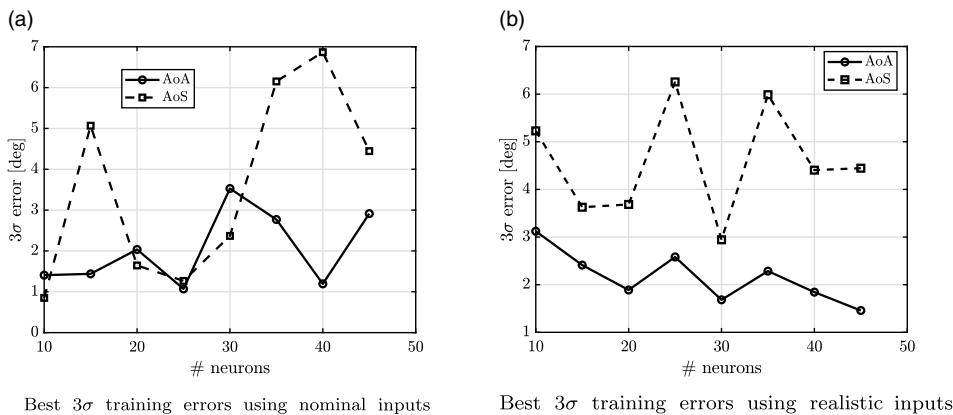


Figure 6. Tradeoff of the MLP-NN's hidden units.

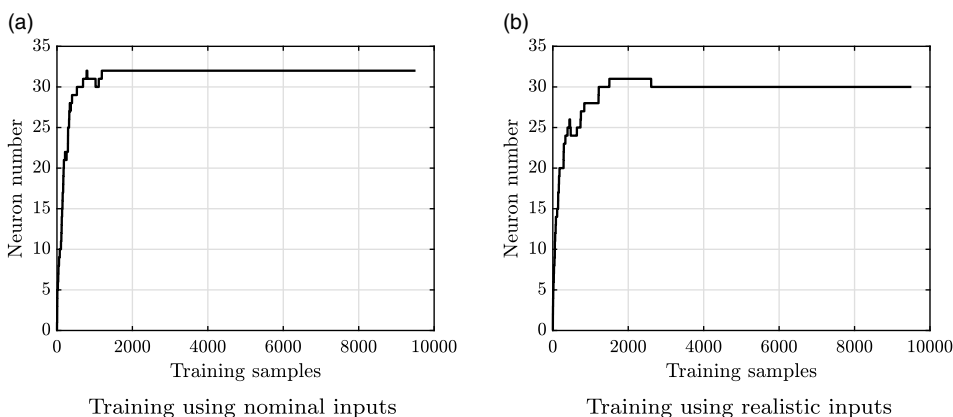


Figure 7. Neuron number evolution during the GRBF-NN training.

5.2 GRBF trade-off

Adopting the minimal resource-allocating network (MRAN) growing algorithm, described in Appendix B, the neuron number is not defined a priori because the algorithm can populate, or prune, the single hidden layer following the logics introduced in Appendix C.

The maximum number of neurons is limited to $K_{max} = 100$. The training evolution of the neuron number is reported in Fig. 7. The stability of the number of neurons below the upper limit K_{max} is a solid indication to determine the convergence of the GRBF-NN. The training of the GRBF-NN (Fig. 7(a)) converged on 32 hidden units, whereas the GRBF-NN-U (Fig. 7(b)) on 30 hidden units. It can be noted that the final number of hidden units is comparable to those selected for the MLP-NNs in Section 5.1.

6.0 Results

Firstly, the ASSE scheme based on three nonlinear equations is solved using: (1) the iterative Levenberg-Marquardt algorithm [17], denoted as LM-3; (2) the MLP-NN trained with nominal data; (3) the GRBF-NN trained with nominal data. Additionally, the neural networks are re-trained using realistic data leading to MLP-NN-U and GRBF-NN-U and their performance are compared to those achieved with NNs trained with uncertainty-free data.

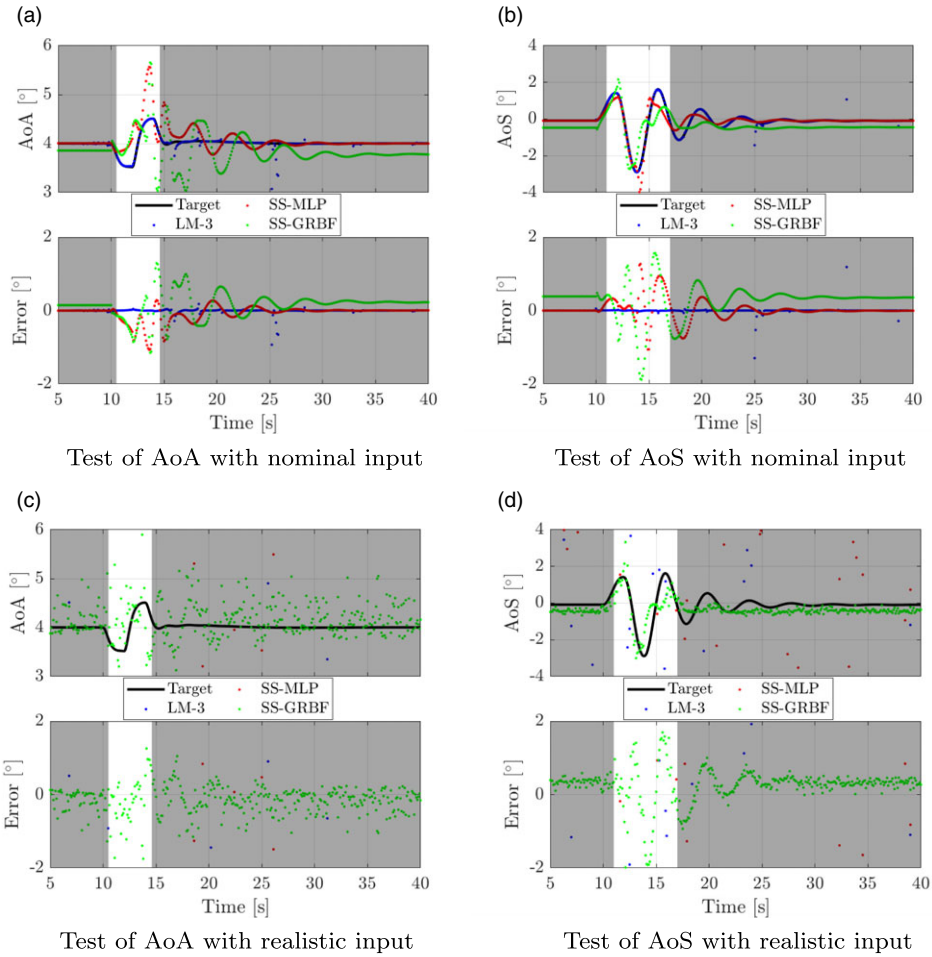


Figure 8. Test of flow angle estimations using the iterative method and neural networks trained with nominal input data. Only for visualisation purposes, results are presented at 10Hz.

In this section, results only belong to the time intervals where the ASSE scheme’s reliability criteria are satisfied as described in Section 4.1.

6.1 Validation of neural networks trained with uncertainty-free input

The AoA and AoS estimation are plotted in Fig. 8 for the test manoeuvre, the error statistics are presented in Table 2. From Fig. 8(a), it is clear that the LM-3 shows the best accuracy for flow angle estimation as both maximum and 2σ errors are acceptable. As far as the neural solvers are concerned, the GRBF-NN exhibits the largest 2σ errors with acceptable maximum errors. Whereas, the MLP-NN shows 2σ errors less than 20 beyond the defined thresholds but with very large maximum errors. The latter aspect suggests the presence of some spike errors. The presence of spike errors, however, does not represent a real issue, in fact, spike errors can be managed using adequate filters. Therefore, as far as the simulated scenario is concerned, the best accuracies are achieved with the LM-3. Whereas, both the neural methods cannot be acceptable for analytical redundancy of UAS’s flow angle sensors considering the thresholds defined in Section 2.1.

Table 2. Test results using the LM-3 method, MLP-NN and the GRBF-NN with nominal and realistic input data. The neural networks are trained with nominal data. Data presented here are only evaluated when the ASSE reliability criteria are satisfied

Test input	Solver	Mean error		Absolute max error		2 σ error	
		AoA	AoS	AoA	AoS	AoA	AoS
Nominal	LM-3	0.00°	0.00°	0.06°	0.08°	0.02°	0.02°
	MLP-NN	-0.41°	53.8°	53.8°	> 90°	1.11°	1.19°
	GRBF-NN	-0.13°	0.08°	1.30°	1.89°	1.23°	1.77°
Realistic	LM	-1.99°	1.15°	27.2°	42.3°	21.0°	37.4°
	MLP-NN	-24.4°	0.49°	>90°	>90°	>90°	>90°
	GRBF-NN	-0.15°	0.03°	2.41°	2.48°	1.42°	1.91°

The same solvers (LM-3, MLP-NN and GRBF-NN) are tested using realistic input signals that are obtained according to Section 3. Time histories of AoA and AoS estimations are reported in Fig. 8(c) and (d) respectively. Error statistics are presented in Table 2.

By means of comparing the 2 σ errors obtained using nominal and realistic inputs for a particular solver, it is clear that the LM-3 and the MLP-NN are not satisfactory to be used in a realistic scenario. In fact, their performance degrade significantly when the realistic input uncertainties are involved.

On the other hand, the GRBF-NN is only marginally affected by realistic inputs. In fact, the 2 σ errors are comparable with those achieved with nominal inputs, even though the maximum errors increase significantly. However, the GRBF-NN cannot satisfy the aforementioned performance requirements as AoA and AoS 2 σ errors are larger than defined limits. In order to improve the GRBF-NN accuracy, several actions can be adopted (e.g. retrain, optimise the training coefficients, etc.) but they are beyond the scope of the present work. To conclude the analysis, it is clear that the iterative solver cannot be adopted to solve the ASSE scheme of three nonlinear equations with realistic input signals.

6.2 Validation of neural networks trained with input uncertainties

This section deals with neural networks that are trained using the same training dataset of Section 6.1 but corrupted with realistic uncertainties. Time histories of the test manoeuvre are reported in Fig. 9, whereas a quantitative error analysis is presented in Table 3. From a qualitative point of view, it can be noted that, comparing Figs 9(c) and (d) with respect to Figs. 8(c) and (d), the MLP-NN-U is able to approximate the realistic test data only if trained with realistic inputs. In other words, aiming to solve the ASSE scheme, the MLP can be predictive in a realistic context only if trained in a similar scenario. Conversely, the GRBF does not show the same sensitivity to input uncertainties.

If error data are compared with respect to those collected in Table 2, the GRBF-NN-U has an improved AoA estimation accuracy with a penalisation of the AoS accuracy. In fact, the GRBF-NN-U satisfies the defined requirements for AoA as both maximum and 2 σ errors are below the thresholds. However, the GRBF-NN-U's AoS performance are worse than those observed with the GRBF-NN and, therefore, the AoS estimation is still not suitable for the aforementioned scopes.

As far as the MLP is concerned, the use of training data corrupted with realistic uncertainties is beneficial to improve the performance with realistic data even though beyond the accuracy requirements.

Both neural solvers trained with realistic data are not affected by the input test type (nominal or realistic). In fact, it can be noted in Table 3 that errors achieved with realistic inputs are comparable with those obtained with nominal inputs.

Although both neural solvers benefit from a training dataset corrupted with realistic uncertainty, their estimation accuracy is not adequate to comply with the performance requirements defined in Section 2.1 for UAS applications. Moreover, in order to assess the neural solver accuracy, training and test

Table 3. Test results using MLP and GRBF neural networks trained with realistic data. Data presented here are only evaluated when the ASSE reliability criteria are satisfied

Test input	Solver	Mean error		Absolute max error		2σ error	
		AoA	AoS	AoA	AoS	AoA	AoS
Nominal	MLP-NN-U	-0.16°	-1.21°	3.15°	3.29°	2.47°	3.09°
	GRBF-NN-U	-0.05°	0.38°	1.00°	2.22°	0.94°	2.18°
Realistic	MLP-NN-U	-0.14°	-1.21°	3.33°	3.54°	2.44°	3.11°
	GRBF-NN-U	-0.08°	0.42°	1.31°	2.74°	0.98°	2.23°

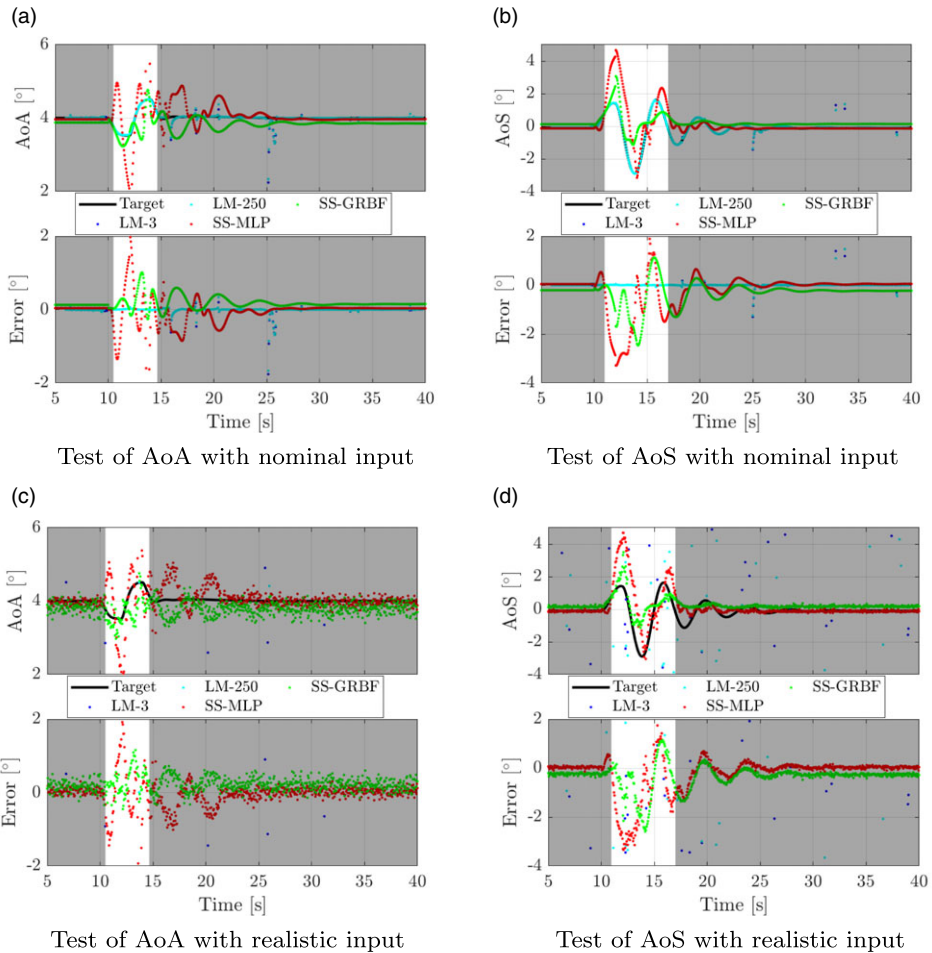


Figure 9. Test of flow angle estimations using the iterative method and neural networks trained with inputs corrupted by realistic uncertainties. Only for visualisation purposes, results are presented at 10Hz.

manoeuvres should cover a realistic flight envelope, e.g. a common flight envelope for UAS applications. The latter topic will be addressed to training optimisation of the ASSE neural solver.

As a final observation, it is worth noting that the GRBF-NNs are more tolerant to input uncertainties even though they are trained with nominal inputs. This latter aspect is crucial when dealing with

the ASSE scheme because it would give the chance to train the neural solver independently from the uncertainty models of the target avionics.

7.0 Conclusion

The ASSE scheme is a model-free approach suitable for developing synthetic sensors for flow angle estimations, e.g. for monitoring and redundancy purposes. The ASSE scheme is defined using three nonlinear equations and it is solved with: (i) the Levenberg-Marquardt algorithm; (ii) the multi-layer perceptron (“global”) neural network; (iii) the generalised radial basis function (“local”) neural network. In order to achieve deterministic solvers, neural networks are pre-trained. In operative scenarios, the iterative method is inadequate. On the contrary, training neural networks with real flight data would hinder the generality of the ASSE scheme. Therefore, as a first step, the neural networks are trained with uncertainty-free data. Considering nominal test data, the iterative solver is able to satisfy the defined performance requirements, whereas the neural solvers do not show the same capability. On the other hand, dealing with test data corrupted with realistic uncertainty, the iterative solver and the global neural network are inadequate because they develop very large errors, whereas the local neural network is more tolerant to input uncertainties. In addition to the uncertainty-free training, both neural networks are re-trained with the same training dataset but corrupted with realistic uncertainties. The “global” neural network takes great advantage of the presence of uncertainties in the training dataset. Whereas, the “local” approximator partially benefits from a realistic training dataset. Between the two neural approaches considered to solve the ASSE scheme, results show that the “local” neural network is more effective in a realistic environment even though it is trained with nominal data. The latter feature is crucial to maintain the generalisation of the synthetic flow angle sensors. In fact, as the ASSE scheme is model-free, i.e. independent from the aircraft application, the solver should also not depend on the target avionics. Therefore, in order to solve the ASSE scheme, the “local” approximator is the most suitable candidate to be used in operative scenarios.

Acknowledgements. The present work is an extended version of “Neural Network Techniques to Solve a Model-Free Scheme for Flow Angle Estimation”, A. Lerro, A. Brandl, P. Gili, presented at the International Conference on Unmanned Aircraft Systems ICUAS 2021. Data considered in this work are available in Ref. [30].

Declarations.

- Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
- Conflict of interest/Competing interests: The authors have no relevant financial or non-financial interests to disclose.
- Ethics approval: not applicable
- Consent to participate: not applicable
- Consent for publication: not applicable
- Availability of data and materials: <https://doi.org/10.5281/zenodo.5938860>
- Code availability: not available
- Competing interests: The authors declare none
- Authors’ contributions: All authors contributed to the study conception. Design, coding, material preparation, data collection and analysis were performed by Angelo Lerro. The manuscript was prepared and reviewed by all authors. All authors read and approved the final manuscript.

References

- [1] Pouliezios, A.D. and Stavrakakis, G.S. *Analytical Redundancy Methods*, vol. 12, Springer Netherlands, 1994.
- [2] Gertler, J. Analytical redundancy methods in fault detection and isolation – survey and synthesis, *IFAC Proc. Vol.*, 1991, **24**, (6), pp 9–21. *IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS’91)*, 10–13 September 1991, Baden-Baden, Germany.

- [3] Taiebat, M. and Sassani, F. Distinguishing sensor faults from system faults by utilizing minimum sensor redundancy, *Trans. Can. Soc. Mech. Eng.*, 2017, **41**, (3), pp 469–487.
- [4] Eubank, R., Atkins, E. and Ogura, S. Fault detection and fail-safe operation with a multiple-redundancy air-data system, *AIAA Guidance, Navigation, and Control Conference*, number August 2010, Reston, Virigina, pp 1–14. American Institute of Aeronautics and Astronautics.
- [5] Lu, P., Van Eykeren, L., Van Kampen, E.-J. and Chu, Q. Air data sensor fault detection and diagnosis with application to real flight data. *AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum (AIAA 2015-1311)*, January 2015, pp 1–18.
- [6] Komite Nasional Keselamatan Transportasi Republic of Indonesia. Aircraft accident investigation report - pt. lion mentari airlines boeing 737-8 (max), 2018. KNKT.18.10.35.04.
- [7] Dendy, J.B. and Transier, K.G. Angle-of-Attack Computation Study, Technical report, Air Force Flight Dynamics Laboratory, 1969. AFFDL-TR-69-93.
- [8] Freeman, D.B. Angle of Attack Computation System, Technical report, Air Force Flight Dynamics Laboratory, 1973. AFFDL-TR-73-89.
- [9] Rohloff, T.J., Whitmore, S.A. and Catton, I. Air data sensing from surface pressure measurements using a neural network method, *AIAA J.*, 1998, **36**, (11), pp 2094–2101.
- [10] Samara, P.A., Fouskitakis, G.N., Sakellariou, J.S. and Fassois, S.D. Aircraft angle-of-attack virtual sensor design via a functional pooling NARX methodology. *2003 European Control Conference (ECC)*, September 2003, vol. 1, pp 1816–1821. IEEE.
- [11] Wise, K.A. Computational Air Data System for Angle-Of-Attack and Angle-Of-Sideslip, 2005. US 6,928,341 B2.
- [12] Crowther, W.J. and Lamont, P.J. A neural network approach to the calibration of a flush air data system, *Aeronaut. J. (1968)*, 2001, **105**, (1044), pp 85–95.
- [13] Yeung, A., Bramesfeld, G., Chung, J. and Foster, S. Measuring low-altitude wind gusts using the unmanned aerial vehicle Gustav, *J. Unmanned Veh. Syst.*, 2018, **6**, (4), pp 235–248.
- [14] Lu, P., Van Eykeren, L., van Kampen, E., de Visser, C.C. and Chu, Q.P. Adaptive three-step kalman filter for air data sensor fault detection and diagnosis, *J. Guid. Control Dyn.*, 2016, **39**, (3), pp 590–604.
- [15] Prem, S., Sankaralingam, L. and Ramprasadh, C. Pseudomeasurement-aided estimation of angle of attack in mini unmanned aerial vehicle, *J. Aerosp. Inform. Syst.*, 2020, **17**, (11), pp 603–614.
- [16] Valasek, J., Harris, J., Pruchnicki, S., McCrink, M., Gregory, J. and Sizoo, D.G. Derived angle of attack and sideslip angle characterization for general aviation, *J. Guid. Control Dyn.*, 2020, **43**, (6), pp 1039–1055.
- [17] Lerro, A., Brandl, A. and Gili, P. Model-free scheme for angle-of-attack and angle-of-sideslip estimation. *J. Guid. Control Dyn.*, 2021, **44**, (3), pp 595–600.
- [18] Lerro, A., Brandl, A. and Gili, P. Neural network techniques to solve a model-free scheme for flow angle estimation, *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp 1187–1193.
- [19] Plataniotis, K.N. and Venetsanopoulos, A.N. Nonlinear filtering of non-Gaussian noise, *J. Intell. Rob. Syst.*, 1997, **19**, (1), pp 207–231.
- [20] Grigorie, L.T. and Botez, R.M. Modeling and numerical simulation of an algorithm for the inertial sensors errors reduction and for the increase of the strap-down navigator redundancy degree in a low cost architecture, *Trans. Can. Soc. Mech. Eng.*, 2010, **34**, (1), pp 1–16.
- [21] Lerro, A., Gili, P. and Pisani, M. Verification in relevant environment of a physics-based synthetic sensor for flow angle estimation, *Electronics*, 2022, **11**, (1). <https://www.mdpi.com/2079-9292/11/1/165>
- [22] Schmidt, D.K. *Modern Flight Dynamics*. Asia Higher Education Engineer. McGraw-Hill, 2011.
- [23] Etkin, B. and Reid, L.D. *Dynamics of Flight: Stability and Control*, 3rd Edition, Wiley, 1995.
- [24] Marquardt, D.W. An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Indus. Appl. Math.*, 1963, **11**, (2), pp 431–441.
- [25] Lerro, A. Physics-based modelling for a closed form solution for flow angle estimation, *Adv. Aircr. Spacecr. Sci.*, December 2021, **8**, (4), pp 273–287.
- [26] Cybenko, G. Approximation by superpositions of a sigmoidal function, *Math. Control Signals Syst.*, December 1989, **2**, (4), pp 303–314.
- [27] Park, J. and Sandberg, I.W. Universal approximation using radial-basis-function networks. *Neural Comput.*, 1991, **3**, (2), pp 246–257.
- [28] Maruyama, M., Girosi, F. and Poggio, T. Connection between GRBF and MLP, *Artif. Intell. Memo*, April 1992, **1**, (1291).
- [29] Haykin, S. *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, 1994.
- [30] Lerro, A. and Brandl, A. Simulated data for ASSE verification: noise-free and noisy data, February 2022.
- [31] SAE International. Minimum performance standards, for angle of attack (AoA) and angle of sideslip (AoS), 2019.
- [32] Beale, M.H., Hagan, M.T. and Demuth, H.B. *Deep Learning Toolbox*, The MathWorks, 2021.
- [33] Fabri, S.G. and Kadirkamanathan, V. Dynamic structure neural networks for stable adaptive control of nonlinear systems. *IEEE Trans. Neural Netw.*, 1996, **7**, (5), pp 1151–1167.
- [34] Platt, J. A resource-allocating network for function interpolation, *Neural Comput.*, 1991, **3**, (2), pp 213–225.
- [35] Yingwei, L., Sundararajan, N. and Saratchandran, P. Identification of time-varying nonlinear systems using minimal radial basis function neural networks, *IEE Proc., Control Theory Appl.*, March 1997, **144**, (6), pp 202–208.

8.0 Appendix

A. Multi-layer perceptron neural network

The mapping function of a conventional MLP-NN, represented in Fig. 5(a), can be expressed as

$$\tilde{\mathbf{y}} = \mathbf{w}\Phi(\mathbf{W}\mathbf{x} + \mathbf{b}_I) + \mathbf{b}_O, \tag{A1}$$

where:

- Φ is the activation function
- $\mathbf{b}_I \in \mathbb{R}^K$ is the vector with the hidden layer's bias terms
- $\mathbf{b}_O \in \mathbb{R}^{n_O}$ is the vector with the output layer's bias terms
- $\mathbf{w} \in \mathbb{R}^{K \times n_O}$ is the matrix of the connection weights between the hidden units and the output layer
- $\mathbf{W} \in \mathbb{R}^{n_I \times K}$ is the matrix of the connection weights between the input layer and the hidden units
- K is the number of hidden units (or neurons)

The activation function Φ is defined as:

$$\Phi(z_k) = \frac{1}{1 + e^{-z_k}}, \tag{A2}$$

where the input z_k to the k -th hidden unit is the k -th element of the vector defined as $\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}_I$.

The batch Back-Propagation algorithm [32] is adopted to train the MLP-NNs. The latter choice makes the training outcome not related to the data order. With the training, the neural network parameters (\mathbf{W} , \mathbf{W} , \mathbf{b}_I and \mathbf{b}_O) are optimised to minimise the mean square error over the entire training dataset. The Levenberg-Marquardt algorithm is chosen to train the MLP-NN.

When the MLP-NN is trained using training data corrupted with realistic uncertainties, it is denoted as MLP-NN-U.

B. Generalised radial basis function

The other neural solver considered in the present work is included in the group of GRBF-NNs that is illustrated in Fig. 5(b). Even though the GRBF-NN can be successfully applied for on-line and sequential learning [33], a sequential off-line training is adopted in this work making the GRBF-NN sensitive to the data order adopted during the training stage. For the latter reason, the training database is shuffled in a random order.

The selected radial basis function is the Gaussian function of Equation (B2) that can model the local function representation, as the radial basis function's output diminishes exponentially moving away from its centre (or mean value). The latter aspect is the most important distinction between the MLP and GRBF approaches.

In a GRBF-NN, the mapping function can be expressed as

$$\tilde{\mathbf{y}} = \mathbf{b}_0 + \sum_{k=1}^K \mathbf{b}_k \phi_k(\mathbf{x}), \tag{B1}$$

where \mathbf{b}_0 is the vector with the bias terms, \mathbf{b}_k is the weight vector associated to the k -th neuron. The k -th activation function ϕ_k is defined as:

$$\phi_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_k\|^2}{2\sigma_{NN,k}^2}\right), \tag{B2}$$

where \mathbf{x} is the input vector and $\boldsymbol{\mu}_k$ is the centre vector of the k -th hidden unit. The hidden units are placed on a uniform n_I -dimensional domain comprising the region of the input space.

C. Description of the minimal resource-allocating network algorithm

The Resource-Allocating Network (RAN) growth algorithm [34] is a sequential learning technique that is based on three criteria

$$\begin{aligned} & \|e_n\| = \|y(n) - \hat{y}(n)\| \geq E1 \\ & \|x(n) - \mu_{nr}\| \geq E2 \\ & e_{rsm} = \sqrt{\frac{\sum_{i=n-(M-1)}^n \|y(i) - \tilde{y}(i)\|^2}{M}} \geq E3 \end{aligned}, \quad (C1)$$

where μ_{nr} is the closest of the K neurons to the current input $x(n)$ and M represents the size of a sliding data window that covers a defined number of latest consecutive observations.

When all three growth criteria of Equation (C1) are satisfied, a new neuron ($K + 1$) is added to the GRBF-NN; otherwise, only the network parameters of the GRBF-NN are updated accordingly to Ref. [35]. When a new neuron is added, its centre μ_{M+1} , variance σ_{M+1} and weights w_{M+1} are updated as

$$\begin{aligned} b_{K+1} &= e_n \\ \mu_{K+1} &= x_n \\ \sigma_{NN,K+1} &= \kappa \|x(n) - \mu_{nr}\| \end{aligned}, \quad (C2)$$

where κ is the overlap factor that determines the overlap of the response in the input space of the added neuron and the closest one.

In order to avoid an excessive increase of the network size, a pruning strategy can also be applied; this modified training strategy is called minimal RAN (MRAN) [35]. The pruning criteria is defined as

$$r_k^n < \delta, \quad (C3)$$

where δ is a predefined threshold and r_k^n is the normalised output of the single k -th hidden unit [35]. In fact, when a certain neuron does not participate actively to the network output (with a normalised contribution less than δ) for M consecutive samples, it is considered useless and, hence, pruned.

Therefore, the number of neurons is not defined a priori but the training algorithm can populate the single hidden layer according to aforementioned logics.

In this work, training data are randomly shuffled and then used for the training. The resulting GRBF-NN has 32 neurons using the following training parameters:

- $E1 = 0.15$
- $E3 = 0.20$
- $\epsilon_{max} = 0.85$
- $\epsilon_{min} = 0.50$
- $\gamma_{err} = 0.997$
- $\kappa = 0.85$
- $M = 100$
- $\delta = 0.0001$

where $E2 = \max\{\epsilon_{max}\gamma^n, \epsilon_{min}\}$ and n is the number of the current sample. The values used in this work are inspired to literature examples [35] and tuned according to a trial and error approach. The training parameters are the same except for the $\kappa = 0.95$ used for the GRBF-NN-U.