POLITECNICO DI TORINO
Repository ISTITUZIONALE

Robust Synthetic Air Data Estimation via Kalman-Aided Deep Learning Approach for Analytical Redundancy

Original
Robust Synthetic Air Data Estimation via Kalman-Aided Deep Learning Approach for Analytical Redundancy / Bang, H., Lerro, A., Youn, W., Ahn, H.. - In: IEEE SENSORS JOURNAL. - ISSN 1530-437X. - 26:6(2026), pp. 9347-9355. [10.1109/jsen.2026.3658151]

Availability:
This version is available at: 11583/3007356 since: 2026-02-04T16:56:59Z

Publisher:
IEEE

Published
DOI:10.1109/jsen.2026.3658151

Terms of use:

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

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

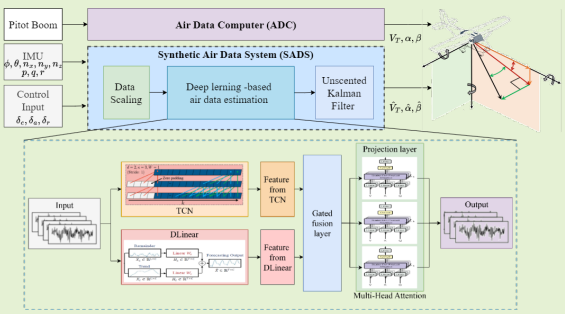
07 November 2026

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

Robust Synthetic Air Data Estimation via Kalman-Aided Deep Learning Approach for Analytical Redundancy

Hyuntae Bang, Angelo Lerro, Wonkeun Youn*, and Hyojung Ahn*

***Abstract*— Reliable air data is essential for safe and stable flight operations. However, physical air data sensors are susceptible to failure due to environmental disturbances, especially in ultralight manned aircraft where hardware redundancy is often impractical due to strict size, weight, and power constraints. Although model-based synthetic air data systems (SADS) have been proposed to reduce sensor reliance, they require precise aerodynamic coefficients and are sensitive to modeling errors. To address these limitations, this study proposes a lightweight, data-driven SADS framework based on a hybrid deep learning model that combines temporal and trend-based features. An unscented Kalman filter (UKF) is applied as a post-processing step to enhance robustness against noise and anomalous inputs. The system is trained and validated on real-world flight data and demonstrates improved accuracy and stability over conventional deep learning baselines. These results suggest that the proposed method offers a robust and complementary alternative to model-based SADS, particularly in resource-constrained flight environments.**



***Index Terms*— Synthetic Air Data System (SADS), Deep Learning, Temporal Convolutional Network (TCN), Decomposition-based linear (DLinear), Unscented Kalman Filter (UKF), Analytical Redundancy.**

I. Introduction

AIR data parameters such as airspeed, angle of attack (AOA), and sideslip angle (SSA) are essential for maintaining flight stability, enabling precise control, and ensuring the overall safety of aircraft operations. These parameters are typically obtained through physical sensors mounted externally on the airframe, including pitot-static systems, vane sensors, and porous probes [1]. However, such sensors are inherently vulnerable to environmental hazards such as icing, moisture, contamination, and physical damage, which can lead to sensor failure and compromise flight safety [2]. While conventional aircraft often employ hardware redundancy to mitigate these risks, this approach is often impractical for ultralight manned aircraft due to strict limitations on size, weight, and power.

Alternatively, synthetic air data systems (SADS) have emerged to estimate air data parameters virtually, using information from onboard sensors (e.g., global navigation satellite systems (GNSS), inertial measurement units (IMUs), and inertial navigation systems (INS)) rather than relying solely on dedicated airflow sensors [3]. By leveraging mathematical models, sensor fusion algorithms, or machine learning techniques, SADS can support analytical redundancy (i.e., estimating air data virtually rather than relying on physical sensors), enhance system reliability, detect anomalies, and enable continued operation under abnormal conditions [4], [5].

Research on SADS has primarily followed two directions. Model-based approaches estimate air data using aerodynamic equations and aircraft dynamics, often incorporating state estimators and sensor fusion schemes [6], [7]. However, these methods rely on accurate aerodynamic coefficients (e.g., lift and drag parameters), which are often challenging to obtain for diverse aircraft platforms.

Although estimating these coefficients from flight data is theoretically possible, such estimation is typically limited by assumptions of steady flight, small angles of attack, or restricted maneuvering. Moreover, the process still depends on clean data and precise system identification, which may be impractical under real-world conditions involving turbulence, sensor noise, or complex flight dynamics.

In contrast, data-driven methods bypass the need for explicitly modeling aerodynamic coefficients by learning representations directly from flight data. While the learned models

This research was carried out with the support of the Jeju RISE Center, funded by the Ministry of Education and Jeju Special Self-Governing Province in 2025, as part of the "Regional Innovation System & Education (RISE): Glocal University 30" initiative.

H. Bang and W. Youn are with department of Autonomous Vehicle System Engineering, Chungnam National University, Yuseong-gu, Daejeon, 34134, Republic of Korea (e-mail: htbang@g.cnu.ac.kr; wkyoun@cnu.ac.kr).

A. Lerro is with department of Mechanical and Aerospace Engineering, Polytechnic University of Turin, Turin, Italy (e-mail: angelo.lerro@polito.it).

H. Ahn is with department of artificial intelligence, Jeju National University, Jeju, Republic of Korea (e-mail: hjahn@jejunu.ac.kr).

*Corresponding author: Wonkeun Youn and Hyojung Ahn

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

may contain a large number of parameters, they can generalize across dynamic conditions and adapt to varying flight regimes without requiring manual tuning of physical constants. This makes them a compelling alternative to classical methods, especially in scenarios where robustness and adaptability are critical.

Recent studies have demonstrated that intelligent and optimization-based control frameworks can significantly enhance UAV robustness under sensor noise and adverse environmental conditions, such as icing [8], [9]. In parallel with these advancements, several works have explored data-driven approaches specifically for air data estimation. Lim et al. employed a temporal convolutional network (TCN) to estimate airspeed as a backup in case of sensor failure, while Youn et al. proposed virtual AOA and SSA sensors based on deep learning to enhance redundancy in high-altitude UAVs [10], [11].

Furthermore, hybrid optimization-based intelligent methods have shown strong performance in UAV subsystems characterized by nonlinear dynamics and uncertainty, further motivating the use of combined learning and filtering architectures [12], [13]. While these approaches primarily focus on augmenting individual components of the air data system, the present work aims to integrate airspeed, AOA, and SSA estimates within a unified hybrid architecture.

To address these challenges, a deep learning-based SADS is proposed that considers the trade-offs between robustness and computational efficiency. The model is trained and validated on real flight data and features a hybrid architecture combining TCNs for capturing short-term dynamics with decomposition-based linear (DLinear) networks for modeling long-term trends. To further improve stability under uncertain conditions, an unscented Kalman filter (UKF) is applied as a post-processing module. This design enhances robustness compared to traditional methods while maintaining low computational cost, supporting its applicability to analytical redundancy in resource-constrained airborne systems.

II. Fundamentals of Air Data Estimation

Airspeed, AOA, and SSA can be derived based on aircraft equations of motion and aerodynamic coefficients. These variables can be used for analytical redundancy as they can be indirectly estimated in situations where sensor-based measurements are limited or impossible.

Based on Newton-Euler equations of motion, the dynamic equations of True airspeed (V_T), AOA (α), and SSA (β) derived using body-frame accelerations, attitude angles, and gravity components measured from IMU sensors can be expressed as follows [14], [15]:

$$\dot{V}_T = a_x \cos\alpha \cos\beta + a_y \sin\beta + a_z \sin\alpha \cos\beta + g\Delta V_T \tag{1}$$

$$\dot{\alpha} = \frac{1}{V_T \cos\beta}\left(-a_x \sin\alpha + a_z \cos\alpha + g\Delta\alpha\right) + q - \left(p\cos\alpha + r\sin\alpha\right)\tan\beta \tag{2}$$

$$\dot{\beta} = \frac{1}{V_T}\left(-a_x \cos\alpha \sin\beta + a_y \cos\beta - a_z \sin\alpha \sin\beta + g\Delta\beta\right) + p\sin\alpha - r\cos\alpha \tag{3}$$

In the equations, $\{a_x, a_y, a_z\} \in a_b$ and $\{p, q, r\} \in \omega$ can be defined as acceleration and angular velocity respectively, ϕ and θ are defined as roll and pitch angles. g is the acceleration due to gravity, assumed to be a constant of 9.80665 m/s^2. ΔV_T, $\Delta\alpha$, and $\Delta\beta$ represent the components of gravitational acceleration projected onto the body-axis directions that influence the airspeed, angle of attack, and sideslip angle, respectively. These projections are determined based on the aircraft's attitude angles (pitch, roll, and yaw), and can be defined as follows:

$$\Delta V_T = -\cos\alpha\cos\beta\sin\theta + \sin\beta\sin\phi\cos\theta + \sin\alpha\cos\beta\cos\phi\cos\theta \tag{4}$$

$$\Delta\alpha = \sin\alpha\sin\theta + \cos\alpha\cos\phi\cos\theta \tag{5}$$

$$\Delta\beta = \cos\alpha\sin\beta\sin\theta + \cos\beta\cos\theta\sin\phi - \sin\alpha\sin\beta\cos\phi\cos\theta \tag{6}$$

In addition to these methods, assuming steady or quasi-steady flight conditions allows the omission of dynamic terms in the equations of motion, enabling the direct estimation of AOA and SSA based on aerodynamic coefficients and load factors derived from force equilibrium. Under the small-angle assumption, where AOA (α) and SSA (β) are sufficiently small, the nonlinear relationships between aerodynamic forces and motion can be linearized. By combining these linearized force expressions with an aerodynamic model, the following static estimation formulas can be obtained [16]:

$$\hat{\alpha} = \frac{-\left(C_{L_0} + C_{L_{\bar{q}}}\frac{qc}{2V} + C_{L_{\delta_E}}\delta_E\right)\left(\frac{\bar{q}S}{W}\right) - n_z}{C_{L_\alpha}\left(\frac{\bar{q}S}{W}\right) - n_x} \tag{7}$$

$$\hat{\beta} = \frac{n_y\left(\frac{W}{\bar{q}S}\right) - C_{Y_{\hat{p}}}\frac{pb}{2V_T} - C_{Y_{\hat{r}}}\frac{rb}{2V_T} - C_{Y_{\delta_A}}\delta_A - C_{Y_{\delta_R}}\delta_R}{C_{Y_\beta}} \tag{8}$$

The static estimation equations utilize various aerodynamic coefficients and variables, each with significance in aircraft dynamics. In the longitudinal axis, C_{L_0} denotes the zero-angle lift coefficient, while C_{L_α} represents the lift curve slope relative to AOA (α). Additionally, C_{L_q} indicates how lift changes with pitch rate (q), and $C_{L_{\delta_E}}$ quantifies lift variation due to elevator deflection (δ_E).

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

For lateral-directional dynamics, several key coefficients come into play. The side force derivative (C_{Y_β}) captures the relationship between SSA (β) and resulting side force. Similarly, C_{Y_p} and C_{Y_r} describe how roll rate (p) and yaw rate (r) affect side forces respectively. Control surface effects appear in $C_{Y_{\delta_A}}$ for aileron input (δ_A) and $C_{Y_{\delta_R}}$ for rudder deflection (δ_R).

Wind tunnel testing and computational fluid dynamics (CFD) simulations typically determine these aerodynamic coefficients. Since they depend primarily on aircraft geometry and configuration, these parameters remain constant during normal flight operations. The dynamic pressure follows the formula $\overline{q} = \frac{1}{2}\rho V^2$, while geometric parameters include the wing area (S), mean aerodynamic chord (c), and wingspan (b). The aircraft weight (W) is treated separately as a physical property.

Control inputs correspond to elevator (δ_E), aileron (δ_A), and rudder (δ_R) deflections, while onboard gyroscopes measure angular rates (p, q, and r). Additionally, accelerometer readings yield load factors along body axes (n_x, n_y, and n_z) through the relationship $n_i = a_i/g$, where g represents gravitational acceleration.

This approach offers a computationally efficient method for real-time estimation of AOA and SSA, when direct sensor measurements are unavailable. However, the reliability of the estimation is built upon key factors, including the validity of steady-flight and small-angle assumptions, and the fidelity of the aerodynamic model parameters.

In summary, while the dynamic and static formulations presented above describe the theoretical relationships between rigid-body dynamics and air data states, relying solely on these models for estimation is challenging due to the uncertainty of aerodynamic coefficients and sensor noise.

Therefore, in this paper, these physics-based equations are primarily utilized to validate the physical consistency of the selected input features for the deep learning model. By grounding the feature selection in these equations, we ensure that the inputs physically govern the target states.

Crucially, the proposed deep learning framework does not require explicit knowledge of aerodynamic coefficients during the inference phase; instead, it learns the nonlinear mapping directly from flight data, thereby bypassing the limitations of parameter uncertainty and simplifying assumptions inherent in classical model-based approaches.

III. Deep Learning-Based Synthetic Air Data Estimation

A. Model Architecture

The proposed TCNDLinear model is a hybrid deep learning architecture for synthetic air data estimation that combines two complementary time-series encoders: a TCN and a DLinear. The TCN captures local temporal features using dilated convolutions, while the DLinear module separately extracts and extrapolates trend and seasonal components to model long-term behavior. Both outputs are projected into a common latent space and fused using a learnable gating mechanism that adaptively balances their contributions. The fused representation is then passed through variable-specific multi-head self-attention modules, with each module dedicated to predicting one of the target variables. This design enables the model to capture short-term dynamics, long-term trends, and cross-variable interactions, supporting robust estimation even under sensor faults and flight anomalies (see Fig. 1).

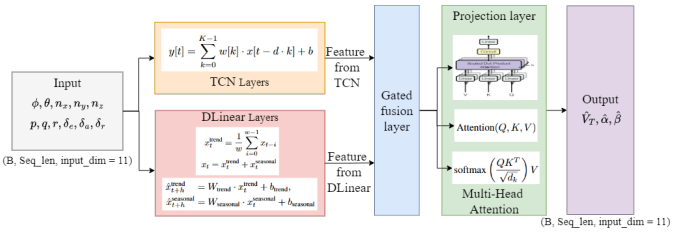


Fig. 1. Architecture of the proposed TCNDLinear model, which combines TCN and DLinear feature extractors with a self-attention fusion module

The input variables for the deep learning network were selected based on (1), (7), and (8), focusing on variables excluding constant aerodynamic coefficients and geometric factors from each equation. Accordingly, the input variables corresponding to the output variables airspeed, AOA, and SSA were defined as $[\phi, \theta, n_x, n_y, n_z, p, q, r, \delta_e, \delta_a, \delta_r] \in \mathbb{R}^{11}$.

B. Data Preprocessing

Time-series deep learning models are sensitive to scale differences among input features, making data normalization a critical preprocessing step. Aircraft sensor data include various physical quantities that differ in range and unit. Feeding these raw values directly into the model may degrade performance due to the disproportionate influence of high-magnitude variables. To mitigate this, a standard score normalization (Z-score) method is applied, defined as:

$$z = \frac{x - \mu}{\sigma} \tag{9}$$

where μ and σ represent the mean and standard deviation of each feature, respectively. This transformation standardizes inputs to have zero mean and unit variance, improving training efficiency and convergence. Although Z-score normalization is not fully robust to outliers, it provides better balance compared to raw input features and aligns well with models like DLinear, which benefit from standardized inputs. Given the presence of sensor noise and environmental disturbances in aircraft data, such normalization is essential for reliable learning.

C. Feature Extractors: TCN and DLinear

Deep learning neural networks offer a powerful means of modeling the complex and nonlinear relationships between air data and associated input variables. TCNs are particularly effective at capturing short-range temporal dependencies and complex nonlinear patterns in time series data through the use of dilated convolution operations [17]. The mathematical formulation of the dilated convolution is as follows [18]:

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

$$y[t] = \sum_{k=0}^{K-1} w[k] \cdot x[t - dk] + b \tag{10}$$

where $w[k]$ denotes the weights of the convolution kernel, d is the dilation factor, and K represents the kernel size. By utilizing dilated convolutions, TCNs can effectively capture long-range temporal dependencies by expanding the receptive field without increasing network depth.

In contrast to complex nonlinear models, DLinear adopts a decomposition-based approach that separates the input time series into trend and seasonal components [19]. The decomposition is typically performed using a moving average over a sliding window of size (w), which captures the underlying trend. The seasonal component is then obtained as the residual between the original signal and the trend:

$$\begin{aligned} x_t^{\text{trend}} &= \frac{1}{w} \sum_{i=0}^{w-1} x_{t-i} \\ x_t^{\text{seasonal}} &= x_t - x_t^{\text{trend}} \\ x_t &= x_t^{\text{trend}} + x_t^{\text{seasonal}} \end{aligned} \tag{11}$$

Each component is modeled independently using a linear transformation, and the final prediction is obtained by summing the two outputs:

$$\begin{aligned} \hat{x}_{t+h}^{\text{trend}} &= W_{\text{trend}} \cdot x_t^{\text{trend}} + b_{\text{trend}} \\ \hat{x}_{t+h}^{\text{seasonal}} &= W_{\text{seasonal}} \cdot x_t^{\text{seasonal}} + b_{\text{seasonal}} \end{aligned} \tag{12}$$

where $x_t^{\text{trend}}, x_t^{\text{seasonal}} \in \mathbb{R}^d$ are the trend and seasonal input vectors, and $W_{\text{trend}}, W_{\text{seasonal}} \in \mathbb{R}^{d \times d}$ are the corresponding weight matrices for linear projection. The bias terms $b_{\text{trend}}, b_{\text{seasonal}} \in \mathbb{R}^d$ are added after projection.

D. Feature Fusion: Gating and Self-Attention

To effectively integrate the local temporal features extracted by the TCN (H_{TCN}) and the global trend features captured by the DLinear (H_{DL}), we employ a learnable gating mechanism followed by a variable-specific multi-head attention projection.

First, both H_{TCN} and H_{DL} are projected into a shared latent feature space with dimension 11, corresponding to the dimensionality of the input variables. To dynamically balance the contribution of short-term dynamics and long-term trends, a gating vector $g_t \in \mathbb{R}^{11}$ is computed based on the concatenation of both feature sets. The mathematical formulation of this feature fusion at time step t is defined as:

$$\begin{aligned} g_t &= \sigma(W_g[H_{\text{TCN},t}; H_{\text{DL},t}] + b_g) \\ H_{\text{fused},t} &= g_t \odot H_{\text{TCN},t} + (1 - g_t) \odot H_{\text{DL},t} \end{aligned} \tag{13}$$

where $[\cdot;\cdot]$ denotes the concatenation operation, $W_g \in \mathbb{R}^{11 \times 22}$ is the learnable weight matrix, b_g is the bias vector, σ is the sigmoid activation function, and $\odot$ represents element-wise multiplication. This mechanism allows the model to selectively emphasize either local or global features for each feature dimension adaptively.

Subsequently, to decode the fused features into the final target variables (airspeed V_T, angle of attack α, and sideslip angle β), we adopt a variable-specific attention structure. Unlike standard transformer architectures that typically share a single attention block for all outputs, our model instantiates 3 independent multi-head attention modules, where each module is dedicated to predicting a single target variable.

For the i-th target variable (where $i \in \{1, \ldots, N_{out}\}$), the fused representation H_{fused} is projected into query (Q_i), key (K_i), and value (V_i) spaces. The output is computed via scaled dot-product attention:

$$\text{Attention}_i(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{14}$$

where d_k is the dimension of the key vectors. To capture diverse temporal dependencies, each dedicated module utilizes 32 parallel attention heads. The outputs of these heads are concatenated and linearly projected to generate the final scalar prediction for the i-th variable. This separation ensures that the attention mechanism learns unique temporal dependencies specific to the physics of each aerodynamic state independently.

E. Network Training and Ablation Analysis

The model training and ablation analysis were conducted using independent flight logs collected from separate experiments. Detailed specifications of these datasets, including sample counts and trajectory characteristics, are described in Section IV-A.

The proposed deep learning model was implemented and trained using the PyTorch framework. Training was conducted on a system equipped with an 11th Gen Intel Core i9-11900K (3.50 GHz), 32 GB of RAM, and an NVIDIA RTX A5000 GPU with 24 GB of VRAM.

The training was capped at a maximum of 50 epochs, with early stopping triggered if training loss did not improve for 15 consecutive epochs. Optimization was performed using the AdamW algorithm, with an initial learning rate of 0.001 and a weight decay factor of 0.0001. To adjust the learning rate dynamically over the training period, a cosine annealing learning rate scheduler was employed [20].

The loss function used was L1 loss, and gaussian error linear unit (GELU) was applied as the activation function to introduce nonlinearity in the model [21]. The batch size during training was set to 256. Ablation experiments were conducted to quantitatively assess the impact of different model structures and hyperparameters on prediction performance.

The model's sensitivity to input and prediction sequence lengths was evaluated using root mean square error (RMSE). Input lengths from 32 to 256 and prediction lengths from 8 to 32 were tested. As illustrated in Fig. 2, the combination of input length 64 and prediction length 16 yielded the lowest average RMSE across all target variables, indicating a robust and well-balanced configuration.

Next, structural variations in the TCN component were explored by adjusting the number of layers, channel configurations, and kernel sizes. Specifically, channel configurations of $(32, 32)$, $(32, 64)$, and $(32, 64, 128)$ were evaluated with

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted,
but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

kernel sizes of 3, 5, and 7. As shown in Table I, the shallowest configuration with two layers, channels $(32, 32)$, and a kernel size of 3 achieved the best performance.

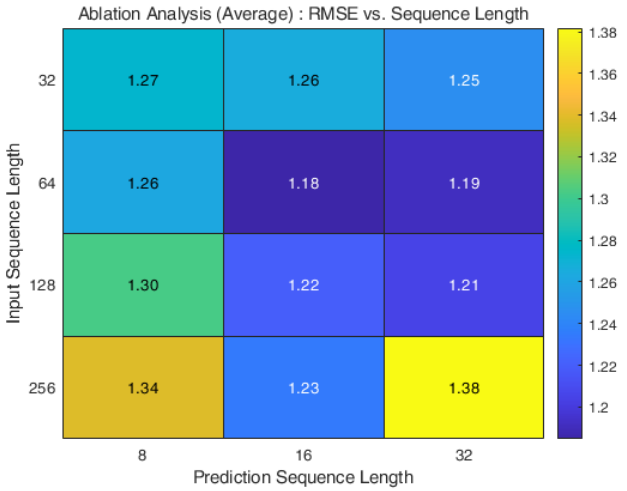


Fig. 2. Heatmap of model performance for varying sequence lengths, showing the average across all target variables

TABLE I

PERFORMANCE VARIATION OF THE PROPOSED MODEL WITH DIFFERENT TCN PARAMETER SETTINGS

Depth and Channels	Kernel size	Airspeed	RMSR AOA	SSA	Avg. RMSE
$(32, 32)$	3	1.2086	0.7758	1.4427	**1.1424**
	5	1.1646	0.7968	1.5639	1.1751
	7	1.3395	0.7688	1.4863	1.1982
$(32, 64)$	3	1.2387	0.7507	1.6421	1.2105
	5	1.3465	0.9213	1.5161	1.2613
	7	1.3082	0.8035	1.7425	1.2847
$(32, 64, 128)$	3	1.4097	0.8429	1.7364	1.3297
	5	1.5012	0.8367	2.0082	1.4487
	7	1.5512	0.8287	1.4406	1.2735

For the DLinear model, the ablation study included testing grouped (shared parameters across channels) versus individual (independent models per variable) settings, as well as different kernel sizes for time series decomposition. According to the results in Table II, the best performance was obtained when both grouped and individual strategies were combined, using a kernel size of 25.

To compare fusion strategies, the multi-head self-attention mechanism was replaced with MLP and GRU layers. The attention-based configuration yielded the lowest RMSE across all target variables: airspeed RMSE was 1.10 (compared to 1.19 with MLP and 1.74 with GRU), AOA RMSE was 0.74 (vs. 0.77 and 0.81), and SSA RMSE was 1.60 (vs. 1.72 and 1.87). These results indicate the superiority of attention-based feature integration over simpler fusion methods.

F. Post-Estimation Filtering with UKF

To enhance the reliability of deep learning-based air data predictions under uncertain and nonlinear flight conditions,

TABLE II

PERFORMANCE VARIATION OF THE PROPOSED MODEL WITH DIFFERENT DLINEAR PARAMETER SETTINGS

Grouped	Individual	Kernel size	Airspeed	RMSR AOA	SSA	Avg. RMSE
False	False	15	1.2232	0.7051	1.6274	1.1852
		25	1.4257	0.9090	1.5929	1.3092
		35	1.6738	0.8008	1.6167	1.3638
False	True	15	1.2272	0.8731	1.5477	1.2160
		25	1.5346	0.8324	1.5414	1.3028
		35	1.5498	0.7603	1.6788	1.3296
True	False	15	1.5610	0.8352	1.6190	1.3384
		25	1.2983	0.8140	1.7249	1.2791
		35	1.4388	0.7702	1.7527	1.3206
True	True	15	1.3126	0.6564	1.5955	1.1882
		25	1.3022	0.5765	1.3646	**1.0811**
		35	1.3246	0.6427	1.8147	1.2607

the UKF is adopted as a post-processing module. While deep learning models can effectively learn complex nonlinear relationships from time-series sensor data, their predictions may become unstable when faced with input distributions or temporal dynamics not encountered during training. By integrating the UKF with model outputs, the system compensates for such limitations through physics-informed filtering, yielding improved accuracy and temporal consistency.

The UKF estimates the hidden states of a nonlinear dynamic system by propagating a set of sigma points through the system dynamics. Unlike the extended Kalman filter (EKF), which linearizes nonlinear functions using a first-order Taylor expansion, the UKF employs the unscented transform to preserve the mean and covariance of the state distribution under nonlinear transformations more accurately.

The state (x), control input (u), and measurement (z) vectors are defined as follows:

$$
\begin{aligned}
x &= \begin{bmatrix} V_T & \alpha & \beta \end{bmatrix}^\top \\
u &= \begin{bmatrix} a_x & a_y & a_z & \phi & \theta \end{bmatrix}^\top \\
z &= \begin{bmatrix} \hat{V}_T & \hat{\alpha} & \hat{\beta} \end{bmatrix}^\top
\end{aligned} \tag{15}
$$

where the control input u consists of the acceleration and attitude measurements required to compute the state derivatives in (1)–(3). The measurement vector z comprises the predicted values generated by the deep learning model. Since these outputs are synthesized by the neural network rather than obtained from physical sensors, they are explicitly treated as pseudo-measurements within the UKF framework. Consequently, the measurement noise acts as a representation of the estimation uncertainty of the deep learning model rather than hardware sensor noise.

UKF parameters are set following the standard formulation [22], [23]:

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

$$
\begin{aligned}
\lambda_{\text{ukf}} &= \alpha_{\text{ukf}}^2(L_s + \kappa_{\text{ukf}}) - L_s \\
\Psi_0^m &= \frac{\lambda_{\text{ukf}}}{L_s + \lambda_{\text{ukf}}} \\
\Psi_0^c &= \frac{\lambda_{\text{ukf}}}{L_s + \lambda_{\text{ukf}}} + 1 - \alpha_{\text{ukf}}^2 + \beta_{\text{ukf}} \\
\Psi_i^m &= \Psi_i^c = \frac{1}{2(L_s + \lambda_{\text{ukf}})}, \quad i = 1, \ldots, 2L_s
\end{aligned} \tag{16}
$$

where L_{s} denotes the dimension of the augmented state vector used in the UKF, which includes the original state variables (e.g., V_T, α, β, etc.) and additional process noise terms associated with system dynamics. In our implementation, this results in $L_{\text{s}} = 3$. The scaling coefficient α_{ukf} controls the spread of sigma points, with typical values ranging from 10^{-4} to 1. Smaller values of α_{ukf} result in tightly clustered sigma points, whereas larger values produce a wider distribution.

The parameter β_{ukf} incorporates prior distribution knowledge, and is optimally set to 2 for Gaussian priors. The parameter κ_{ukf} is generally set to 0. The weights Ψ_i^m and Ψ_i^c correspond to the i-th sigma point's contribution to the predicted mean and covariance, respectively.

To account for non-additive process noise, the UKF augments the state vector with additional noise terms, as expressed below:

$$
x_k^a = \begin{bmatrix} x_k \\ w_k \end{bmatrix} \tag{17}
$$

where the superscript a indicates the augmented form of the variable. The noise states are initialized to zero at each time step, in accordance with the zero-mean gaussian noise assumption. Augmentation is necessary because the process noise affects the system through the input vector $\mathbf{u}$, and cannot be directly added to the state vector as in the case of additive noise.

Subsequently, the augmented error covariance $P_{k-1|k-1}^a$ for the augmented state $\hat{x}_{k-1}^a$ is constructed as a block diagonal matrix as follows:

$$
P_{k-1|k-1}^a = \text{diag}(P_{k-1}, Q_{k-1}) \tag{18}
$$

Next, $2L_{\text{s}} + 1$ sigma points $\chi_{i,k-1}$ are generated based on the prior mean $\hat{x}_{k-1|k-1}$, the augmented covariance $P_{k-1|k-1}^a$, and the scaling parameter λ_{ukf}, as follows [24]. Here, the scaling factor is defined as $\gamma = \sqrt{L_{\text{s}} + \lambda_{\text{ukf}}}$.

$$
\chi_{i,k-1} = \hat{x}_{k-1|k-1}^a + \Delta\chi_i, \quad \text{where}
$$

$$
\Delta\chi_i = \begin{cases} 0 & \text{if } i = 0 \\ \gamma \left[\sqrt{P_{k-1|k-1}^a} \right]_i & \text{if } i = 1, \ldots, L_s \\ -\gamma \left[\sqrt{P_{k-1|k-1}^a} \right]_{i-L_s} & \text{if } i = L_s + 1, \ldots, 2L_s \end{cases} \tag{19}
$$

In the time update step, the UKF propagates the sigma points through the system dynamics to predict the next state and its covariance. The corresponding equations are given as follows [24]:

$$
\begin{aligned}
\hat{\chi}_{i,k|k-1} &= f_k(\chi_{i,k-1}, u_{k-1}, 0), \quad i = 0, 1, \ldots, 2L_{\text{s}} \\
\hat{x}_{k|k-1} &= \sum_{i=0}^{2L_{\text{s}}} \Psi_i^m \hat{\chi}_{i,k|k-1} \\
P_{k|k-1} &= \sum_{i=0}^{2L_{\text{s}}} \Psi_i^c (\hat{\chi}_{i,k|k-1} - \hat{x}_{k|k-1})(\hat{\chi}_{i,k|k-1} - \hat{x}_{k|k-1})^\top
\end{aligned} \tag{20}
$$

where $\hat{x}_{k|k-1}$ denotes the predicted state vector, estimated from the previous state using the system dynamics. The corresponding covariance matrix, $P_{k|k-1}$, represents the uncertainty of the prediction.

In the measurement update step, the correction is performed using synthetic air data predicted by the deep learning model, as formalized in the following update equations:

$$
\begin{aligned}
\hat{z}_{k|k-1} &= \sum_{i=0}^{2L_{\text{s}}} \Psi_i^m \hat{\varphi}_{i,k|k-1} \\
P_{zz,k} &= \sum_{i=0}^{2L_{\text{s}}} \Psi_i^c (\hat{\varphi}_{i,k|k-1} - \hat{z}_{k|k-1})(\hat{\varphi}_{i,k|k-1} - \hat{z}_{k|k-1})^\top \\
&\quad + R_{k-1} \\
P_{xy,k} &= \sum_{i=0}^{2L_{\text{s}}} \Psi_i^c (\hat{\chi}_{i,k|k-1} - \hat{x}_{k|k-1})(\hat{\varphi}_{i,k|k-1} - \hat{z}_{k|k-1})^\top \\
P_{k|k} &= P_{k|k-1} - K_k P_{zz,k} K_k^\top \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - \hat{z}_{k|k-1})
\end{aligned} \tag{21}
$$

where

$$
\begin{aligned}
K_k &= P_{xy,k} P_{zz,k}^{-1} \\
\hat{\varphi}_{i,k|k-1} &= h_k(\chi_{i,k|k-1}, u_k, 0), \quad i = 0, 1, \ldots, 2L_{\text{s}}
\end{aligned} \tag{22}
$$

where $h_k(\cdot)$ denotes the nonlinear measurement function.

In the measurement update step, a statistical gating mechanism is incorporated to evaluate the validity of the deep learning-based measurements in real time. The innovation ν_k, defined as the difference between the observed measurement z_k and the predicted measurement $\hat{z}_{k|k-1}$, serves as a key indicator for detecting anomalies:

$$
\nu_k = z_k - \hat{z}_{k|k-1} \tag{23}
$$

To quantify the reliability of the measurement, the Normalized Innovation Squared (NIS), denoted as ϵ_k, is computed using the innovation covariance $P_{zz,k}$:

$$
\epsilon_k = \nu_k^\top (P_{zz,k})^{-1} \nu_k \tag{24}
$$

Based on the χ^2 distribution hypothesis, if ϵ_k exceeds a predefined threshold γ_{χ^2}, the deep learning output is considered unreliable. In such cases, the measurement update is skipped,

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

and the system state relies solely on the time update (IMU-based prediction), thereby preventing false data injection from the learning model.

$$\hat{x}_{k|k} = \begin{cases} \hat{x}_{k|k-1} + K_k \nu_k & \text{if } \epsilon_k \leq \gamma_{\chi^2} \\ \hat{x}_{k|k-1} & \text{if } \epsilon_k > \gamma_{\chi^2} \end{cases} \tag{25}$$

This mechanism ensures that the SADS maintains robustness even when the deep learning model encounters out-of-distribution inputs or exhibits uncertainty.

IV. Experimental Validation

A. Experimental Flight and Data Sets

The experimental platform used in this study is the Nando Groppo G70, an Italian ultralight fixed-wing aircraft, as illustrated in Fig. 3. The aircraft features a metal semi-monocoque structure, a fixed tricycle landing gear configuration, and a single-engine airframe. It is powered by a Rotax 912ULS 100HP engine and equipped with a Helix H50F fixed-pitch propeller. Key specifications include a wingspan of 8.9 m, a fuselage length of 6.22 m, a wing area of 10.56 m^2, and a maximum takeoff weight of 472.5 kg. The main wing and horizontal stabilizer utilize NACA 2415 and NACA 0012 airfoils, respectively. A dual control system allows operation from both cockpit seats. The onboard instrumentation suite includes a pitot-static system, accelerometers, gyroscopes, GNSS, and control surface input sensors, enabling real-time acquisition of flight log data during operation [25].

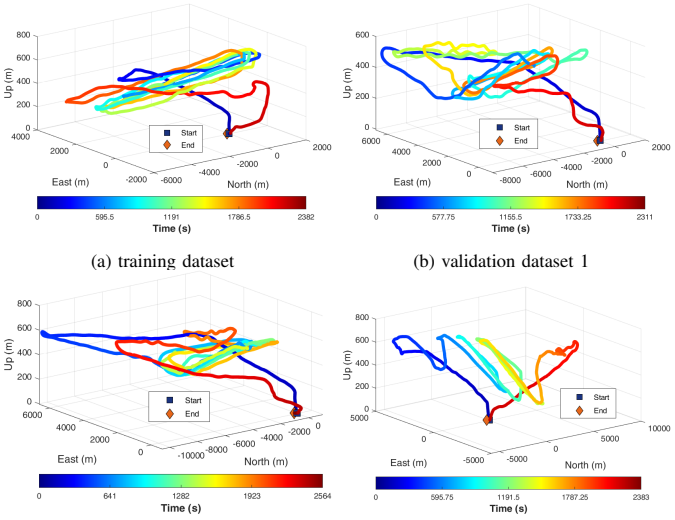
Flight experiments for training and validating the SADS were conducted in northern Italy, using two independent flight test instrumentation (FTI) systems onboard the same aircraft. The primary sensing systems comprise an ADS, AHRS, and GNSS, which measure dynamic/static pressure, acceleration, angular velocity, orientation, position, and velocity. All experiments were conducted with a sampling rate of 50 Hz. The resulting dataset was constructed from four flight test logs, each involving specific maneuvers designed to excite the aircraft's longitudinal and lateral-directional dynamic modes across the flight envelope [26].

The collected datasets capture diverse dynamic behaviors, and their corresponding 3D flight trajectories are visualized in Fig. 4(a)–(d). These datasets consist of 207,376, 197,296, 160,960, and 207,344 time-series samples, respectively. Each dataset contains a comprehensive set of sensor measurements, with input features selected according to the definitions provided in Section III.

For the ablation study, the model was trained using the dataset shown in Fig. 4(a) and validated with the dataset in Fig. 4(b). To further evaluate generalization performance across different mission profiles and maneuvers, the model trained on dataset (a) was tested against all three distinct validation trajectories shown in Fig. 4(b)–(d).

(a) reproduced from [25] (b) reproduced from [26]

Fig. 3. (a) the Nando Groppo G70 ultralight aircraft used for flight testing; (b) the pitot boom mounted on the aircraft for air data acquisition



(a) training dataset (b) validation dataset 1

(c) validation dataset 2 (d) validation dataset 3

Fig. 4. Flight trajectories from four data sequences, where (a)–(d) correspond to different dynamic maneuvers used in dataset construction

B. Comparison with Other Deep Learning Methods

A comparative evaluation of six time series prediction models (LSTM, GRU, TCN, Transformer, DLinear, and TCNDLinear) is presented in Table III and Fig. 5. For each validation dataset, the RMSE was computed for the three target variables. The final column of the table summarizes the overall performance using aggregated metrics (RMSE, standard deviation, and 99th percentile error) calculated across all test samples.

The proposed TCNDLinear model achieved the best overall performance (RMSE: **1.3926**, Stddev: **1.3849**, 99%: **3.7019**), achieved low errors across all variables. In comparison, LSTM and GRU resulted in higher prediction errors, indicating limitations in modeling complex temporal dependencies. DLinear and Transformer models showed notably poor performance in airspeed estimation under certain conditions, suggesting sensitivity to noise or limited generalization capability. Among the evaluated models, the proposed TCNDLinear exhibited robustness to extreme prediction errors, highlighting the benefit of combining local temporal feature extraction with trend modeling for stable air data estimation. The proposed method achieves RMSEs below 2 m/s for airspeed and below 2° for angle estimates, which are comparable to or better than results reported in related work [27].

C. Results of Synthetic Air Data Estimation

To validate the robustness of the proposed UKF-based post-processing framework (detailed in Section III), fault scenarios representing realistic operational degradations were simulated.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

TABLE III

COMPARISON OF MODEL PERFORMANCE ON THREE DATASETS FOR AIRSPEED, AOA, AND SSA PREDICTION. REPORTED METRICS INCLUDE RMSE, STANDARD DEVIATION, AND 99TH PERCENTILE ERROR

Model	Data (b) RMSE			Data (c) RMSE			Data (d) RMSE			Combined Metrics		
	Airspeed	AOA	SSA	Airspeed	AOA	SSA	Airspeed	AOA	SSA	RMSE	Stdev	99%
LSTM	1.9732	1.2680	1.8436	1.6734	1.2308	1.5299	1.1324	1.8020	2.6015	1.7455	1.7414	3.8557
GRU	2.0683	1.2781	1.8113	1.7714	1.2253	1.6416	1.2185	1.6094	2.8180	1.7980	1.7949	4.1978
TCN	1.311	0.8166	1.9246	1.0788	0.7987	1.5705	1.0132	1.4109	2.4073	1.4798	1.4789	4.3979
Transformer	2.3091	1.0223	1.5702	1.9953	0.9379	1.2421	1.1575	1.0788	2.2710	1.6037	1.5963	3.7824
DLinear	2.4114	0.8073	1.6051	2.2596	0.7857	1.3619	1.5239	1.5169	2.0197	1.6834	1.6640	3.7603
TCNDLinear	1.4086	0.9151	1.4913	1.1342	0.8358	1.2291	1.6588	1.2276	2.0239	**1.3926**	**1.3849**	**3.7019**

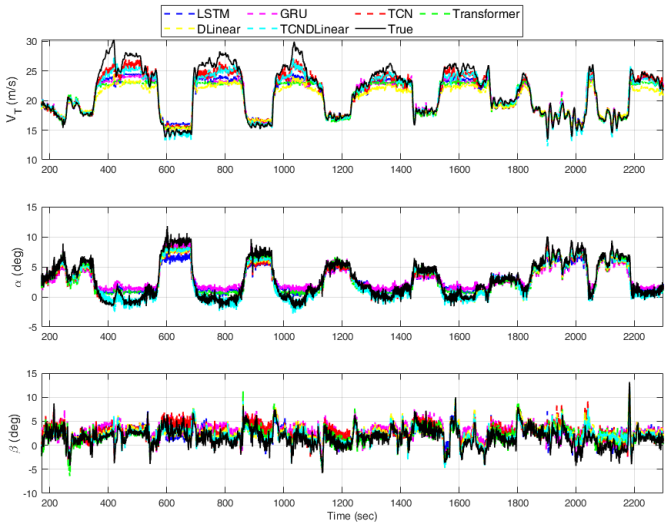


Fig. 5. Predicted versus ground truth values for airspeed (V_T), angle of attack (α), and sideslip angle (β) on the second validation dataset, comparing the outputs of six prediction models (colored lines) against the reference data (black line)

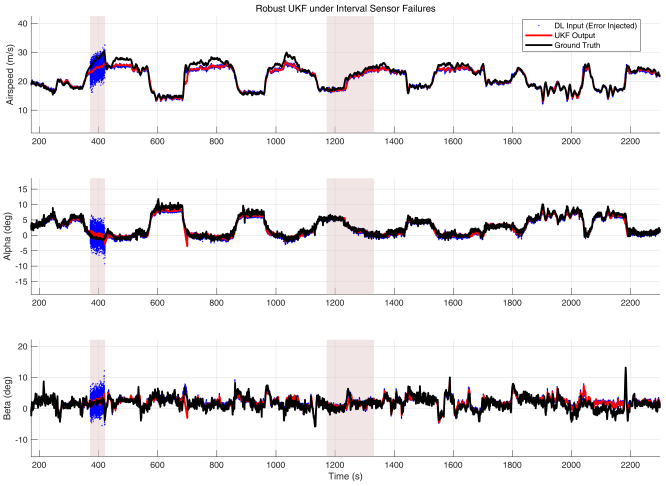


Fig. 6. Robustness comparison under fault injection, showing stable performance of the UKF during two fault-injected intervals

Specifically, two failure modes were introduced to evaluate the statistical gating mechanism: (1) a significant increase in prediction uncertainty due to degraded input data quality, and (2) measurement latency caused by system overload.

For the experimental evaluation, the filter parameters were empirically tuned and fixed to prevent overfitting. The process noise covariance matrix was set to $Q = \mathrm{diag}(1 \times 10^{-3},\ 1 \times 10^{-4},\ 1 \times 10^{-4})$, and the measurement noise covariance matrix was defined as $R = \mathrm{diag}(15,\ 0.1,\ 0.1)$, reflecting the uncertainty characteristics of the deep learning-based measurements. These parameters were consistently applied across all validation trials.

Synthetic outliers were generated by adding ± 2 magnitude noise to randomly selected measurement samples in the early time segments, while approximately 3 s delays were introduced at intermediate intervals to emulate realistic inference degradation and latency effects.

The simulation results show that, in spite of these injected defects, the suggested filter retains rather stable estimate performance, as shown in Fig. 6. The variation of the deep learning output values grew significantly during the [371.8 s – 421.8 s] interval, indicating decreased reliability. The UKF output, however, lessened this effect and reasonably adhered to the ground truth. Similarly, the filter identified delayed measurements and partially inhibited the spread of estimation errors in the [1171.8 s – 1331.8 s] period.

The performance of UKF is further supported by quantitative RMSE metrics. The UKF produced better RMSE values than the deep learning-only case: airspeed dropped from 1.2002 to 1.1391, angle of attack from 0.8953 to 0.8152, and sideslip angle from 1.3006 to 1.2695. These findings demonstrate that the proposed filter considerably lessens the impact of prediction uncertainties in the deep learning model and helps to preserve the general stability of SADS, even though it might not totally eradicate them.

V. CONCLUSION

A deep learning-based synthetic air data estimation framework was developed and validated using real flight test data from an ultralight aircraft. The proposed TCNDLinear model combines temporal convolutional networks for capturing local temporal dynamics with a DLinear module for modeling long-term trends, enabling accurate estimation of airspeed, angle of attack, and sideslip angle. Comparative evaluations against six baseline models demonstrated consistent improvements in prediction accuracy and robustness across multiple flight trajectories. To enhance reliability under abnormal conditions, an unscented Kalman filter was incorporated as a post-processing module. The filtering framework improved estima-

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2026.3658151

tion stability in the presence of synthetic outliers and delayed measurements, indicating its effectiveness as a complementary robustness enhancement rather than a replacement for the learning-based estimator.

While the proposed approach shows stable performance within the tested flight envelope, it has not yet been evaluated under more extreme conditions, such as high angles of attack or airspeeds beyond the training range (e.g., above 30 m/s). Future work will focus on extending the evaluation to such off-nominal scenarios using high-fidelity simulation environments and additional flight data, as well as investigating adaptive filtering strategies to further improve robustness under severe sensor degradation.

REFERENCES

[1] G. Ariante, S. Ponte, U. Papa, and G. Del Core, "Estimation of airspeed, angle of attack, and sideslip for small unmanned aerial vehicles (uavs) using a micro-pitot tube," *Electronics*, vol. 10, no. 19, p. 2325, 2021.

[2] C. T. Boesser, "The effects of angle-of-attack indication on aircraft control in the event of an airspeed indicator malfunction," M.S. thesis, Embry-Riddle Aeronautical University, 2013, available: https://commons.erau.edu/edt/24.

[3] K. Sun, C. D. Regan, and D. G. Egziabher, "Gnss/ins based estimation of air data and wind vector using flight maneuvers," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2018, pp. 838–849.

[4] F. A. P. Lie and D. Gebre-Egziabher, "Sensitivity analysis of model-based synthetic air data estimators," in *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0081.

[5] H. Ahn and S. Chung, "Deep learning-based anomaly detection for individual drone vehicles performing swarm missions," *Expert Systems with Applications*, vol. 244, p. 122869, 2024.

[6] F. A. P. Lie and D. Gebre-Egziabher, "Synthetic air data system," *Journal of Aircraft*, vol. 50, no. 4, pp. 1234–1249, 2013.

[7] W. Youn, H. Choi, A. Cho, S. Kim, and M. B. Rhudy, "Aerodynamic model-aided estimation of attitude, 3-d wind, airspeed, aoa, and ssa for high-altitude long-endurance uav," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4300–4314, 2020.

[8] K. Sun, "Reliable air data solutions for small unmanned aircraft systems," Ph.D. dissertation, University of Minnesota, 2020.

[9] E. Can, "Energy-aware adaptive altitude control of uavs via fuzzy–pso optimization within a port-hamiltonian framework under icing and sensor noise," *International Journal of Aeronautical and Space Sciences*, pp. 1–17, 2025.

[10] H. Lim, H. Ryu, M. B. Rhudy, D. Lee, D. Jang, C. Lee, Y. Park, W. Youn, and H. Myung, "Deep learning-aided synthetic airspeed estimation of uavs for analytical redundancy with a temporal convolutional network," *IEEE Robotics and Automation Letters*, 2021, early Access.

[11] W. Youn, H. Lim, H. S. Choi, M. B. Rhudy, H. Ryu, S. Kim, and H. Myung, "State estimation for hale uavs with deep-learning-aided virtual aoa/ssa sensors for analytical redundancy," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5276–5283, 2021.

[12] H. Karali, M. Uzun, B. Yuksek, and G. Inalhan, "Data-driven synthetic air data estimation system development for a fighter aircraft," in *AIAA AVIATION 2023 Forum*, 2023, p. 3439.

[13] E. Can, "Uav dc motor behaviors control using new hybrid pso-fuzzy logic with a dynamic error based optimization," *Aerospace Science and Technology*, p. 111156, 2025.

[14] E. A. Morelli and V. Klein, *Aircraft system identification: theory and practice*. Sunflyte Enterprises Williamsburg, VA, 2016, vol. 2.

[15] R. C. Nelson *et al.*, *Flight stability and automatic control*. WCB/McGraw Hill New York, 1998, vol. 2.

[16] J. Valasek, J. Harris, S. Pruchnicki, M. McCrink, J. Gregory, and D. G. Sizoo, "Derived angle of attack and sideslip angle characterization for general aviation," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 6, pp. 1039–1054, 2020.

[17] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[18] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[19] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[20] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[21] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[22] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[23] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.

[24] J. Kim, "Cooperative localization and unknown currents estimation using multiple autonomous underwater vehicles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2365–2371, 2020.

[25] A. Lerro, P. Gili, M. L. Fravolini, and M. Napolitano, "Experimental analysis of neural approaches for synthetic angle-of-attack estimation," *International Journal of Aerospace Engineering*, vol. 2021, no. 1, 2021, art. no. 9982722.

[26] F. Battaini, "Identification of the basic aerodynamic model of an ultra-light aircraft from flight test data," M.S. thesis, Politecnico di Milano, 2016.

[27] W. Youn, H. Ryu, D. Jang, C. Lee, Y. Park, D. Lee, and M. B. Rhudy, "Model-aided synthetic airspeed estimation of uavs for analytical redundancy," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5841–5848, 2021.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on February 04,2026 at 16:49:22 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.