IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Deep Learning-aided Synthetic Airspeed Estimation of UAVs for Analytical Redundancy with a Temporal Convolutional Network

Hyungtae Lim[1], Hanseok Ryu[2], Matthew B. Rhudy[3], Dongjin Lee[4], Dongjin Jang[4], Changho Lee[2], Youngmin Park[2], Wonkeun Youn[5*], and Hyun Myung[1*], *Senior Member, IEEE*

***Abstract*—A synthetic air data system (SADS) is an analytical redundancy technique that is crucial for unmanned aerial vehicles (UAVs) and is used as a backup system during air data sensor failures. Unfortunately, the existing state-of-the-art approaches for SADS require GPS signals or high-fidelity dynamic UAV models. To address this problem, a novel synthetic airspeed estimation method that leverages deep learning and an unscented Kalman filter (UKF) for analytical redundancy is proposed. Our novel fusion-based method only requires an inertial measurement unit (IMU), elevator control input, and airflow angles while GPS, lift/drag coefficients, and complex aircraft dynamic models are not required. Additionally, we demonstrate that our proposed temporal convolutional network (TCN) is a more efficient model for airspeed estimation than the renowned models, such as ResNet or bidirectional long short-term memory (LSTM). Our deep learning-aided UKF was experimentally verified on long-duration real flight data and has promising performance compared with the state-of-the-art methods. In particular, it is confirmed that our proposed method robustly estimates the airspeed even under dynamic flight conditions where the performance of conventional methods is degraded.**

***Index Terms*—Aerial Systems: Applications; Sensor Fusion; Field Robots; Temporal Convolutional Network; Deep Learning; Unmanned Aerial Vehicles; Analytical redundancy**

Manuscript received: April 26, 2021; Revised July 29, 2021; Accepted September 12, 2021.

This paper was recommended for publication by Editor P. Pounds upon evaluation of the Associate Editor and Reviewers' comments. This research was supported by a grant (21CAUV-B151724-03) from 'Development of Certification Technology for Small Unmanned Aerial Vehicle Systems' Program Through the Korea Agency for Infrastructure Technology Advancement (KAIA), funded by the Ministry of Land, Infrastructure and Transport (MOLIT) and also supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1093445). The student was supported by the BK21 FOUR from the Ministry of Education (Republic of Korea).

[1]Hyungtae Lim and Hyun Myung are with the School of Electrical Engineering, KI-AI, KI-R at KAIST (Korea Advanced Institute of Science and Technology), Daejeon, 34141, Republic of Korea {shapelim, hmyung}@kaist.ac.kr. [2]Hanseok Ryu, Changho Lee, and Youngmin Park are with the Korea Aerospace Research Institute, Yuseong-gu, Daejeon, 34133, Republic of Korea {ryu1, leech, ympark}@kari.re.kr.[3]Matthew Rhudy is with the Division of Engineering, Business, and Computing at Pennsylvania State University, Reading, PA 19610, USA mbr5002@psu.edu. [4]Dongjin Lee and Dongjin Jang are with the Department of Unmanned Aircraft Systems at Hanseo University, Gomseom-ro, Taean-gun, Chungcheongnam-do, 32158, Republic of Korea djlee@hanseo.ac.kr; loveehdwls_7@naver.com. [5]Wonkeun Youn is with the Department of Autonomous Vehicle System Engineering, Chungnam National University, Yuseong-gu, Daejeon, 34134, Republic of Korea wkyoun@cnu.ac.kr. *The corresponding authors are Wonkeun Youn and Hyun Myung.

Digital Object Identifier (DOI): see top of this page.

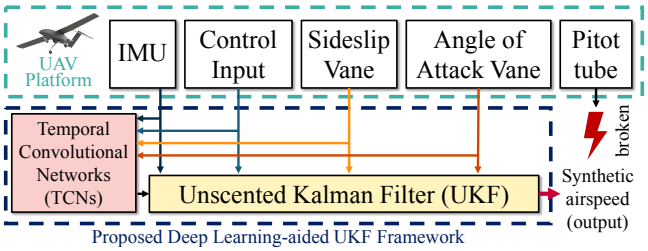

Fig. 1. Schematics of our deep learning-aided unscented Kalman filter (UKF) framework for analytical redundancy to estimate airspeed.

I. Introduction

In recent years, there has been an increased demand for a robust and certifiable synthetic air data system (SADS) to protect against the breakdown of air data systems (ADSs). Basically, ADS is a vital component of the existing sensor suite for both manned and unmanned aerial vehicles (UAVs) and directly provides some measurements, including the angle of attack (AOA), side slip angle (SSA), and airspeed. These parameters contain essential information about an aircraft's performance and safety in both normal and abnormal conditions [1], [2].

Unfortunately, ADSs are usually deployed outside an aircraft fuselage; thus, ADSs are likely to be exposed to conditions which can cause malfunctions. That is, moisture, icing, or water accumulation can cause blockage faults, which impede the ADSs from measuring precise air data [3] and thus can lead to catastrophic accidents. In fact, since 2003, commercial aircraft have had more than 35 recorded incidents due to ADS faults [4].

To tackle this issue, many solutions have been proposed in terms of hardware and software. First, *hardware redundancy* refers to a mechanical solution to prepare for the failure of one sensor by attaching multiple sensors to a commercial manned aircraft [5]. Accordingly, the method is usually based on a voting system, which can detect and discard false measurements [6]. This method is reasonable for large aircraft, yet it is not always appropriate for small UAVs because hardware redundancy does not usually meet the size, weight, and power (SWAP) constraints of UAVs.

For these reasons, *analytical redundancy*, which is also referred to as SADS for this application, has been actively explored as an alternative to hardware redundancy. Analytical redundancy refers to a method for estimating the relevant quantities (i.e., the airspeed, altitude, pressure, air temperature,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Mach number, AOA, and SSA) without requiring a direct measurement [7], which satisfies the SWAP constraints. For instance, analytical redundancy usually utilizes an inertial measurement unit (IMU) combined with a dynamic or kinematic UAV model and other measurement systems, such as GPS, to predict the airspeed, AOA, and SSA [4], [7], [8]. For this reason, SADS is used to monitor anomalies, such as sensor or system failures, in primary ADS; thus, it can serve as a backup system by providing the air data quantities through software [9].

However, previous analytical redundancy methods have two main potential limitations. First, some methods use GPS measurements as a necessary input. Unfortunately, when GPS is not available due to some situations, such as jamming or spoofing [10], the synthetic airspeed estimate will not be available. Second, most of the previous studies utilized high-fidelity vehicle dynamic UAV models to estimate the air data [11]–[13]. However, obtaining an accurate high-fidelity UAV model is a very complex, costly, and time-consuming task. Furthermore, the estimated parameters of the aircraft model can change over time due to wear and tear on the airframe and propulsion system [7], [9] or in certain environments. In aviation, when an aircraft passes through air containing water droplets (moisture), icing can occur if the temperature at the droplet point is below $0^{\circ}C$. The icing on these aircraft causes structural changes in the UAV and changes in the lift or drag coefficients. In such cases, model-based SADS can yield inaccurate estimation results. Since the icing is a common source of ADS failure, SADS must be accurate under these circumstances.

Moreover, in the current era of deep learning [14], numerous attempts have been made to implement neural networks for various regression tasks, e.g., [15]–[18]. In particular, some researchers have adopted these machine learning-based approaches to aerospace applications. For example, Guo *et al.* [17] proposed a novel method for UAV sensor fault diagnoses using convolutional neural networks (CNNs). Allison *et al.* [19] proposed a machine learning approach for wind velocity estimation based on quadcopter state measurements without a wind sensor. Furthermore, in our previous study [20], we successfully showed that the long-short term memory (LSTM)-based AOA/SSA estimation method outperforms conventional model-aided methods for analytical redundancy.

In this paper, a novel deep learning-aided airspeed estimation framework is proposed for analytical redundancy and is presented in Fig. 1. To the best of our knowledge, this study is the first to attempt to introduce deep learning to estimate airspeed without direct GPS and airspeed measurements or relying on a complicated UAV model. The key contribution of this paper is fourfold:

1) A novel, fusion-based filter method that leverages a deep learning model and the unscented Kalman filter (UKF) is proposed for estimating the airspeed without direct GPS measurements or a high-fidelity UAV dynamic model.
2) The proposed deep learning model for synthetic airspeed estimation is shown to outperform conventional methods with small mean error and variance values. In addition, the temporal convolutional network (TCN) is shown to provide the most promising results compared with other deep learning models (i.e., ResNet [14] and bidirectional LSTM [15]) for synthetic airspeed estimation.
3) The performance of all estimation techniques is evaluated on long-duration real flight data from a fixed-wing UAV. Our deep learning-aided method shows robust and promising performance over the state-of-the-art methods, which implies that our proposed method overcomes the aforementioned limitations of previous approaches.

II. Preliminary

From the wind triangle relationship, the airspeed vector, $\mathbf{v}_a$, the AOA α, and the SSA β are defined in the body-fixed frame as follows [21], [22]:

$$\mathbf{v}_a^B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}^B = \begin{bmatrix} V_a \cos\alpha \cos\beta \\ V_a \sin\beta \\ V_a \sin\alpha \cos\beta \end{bmatrix} \tag{1}$$

$$\alpha = \tan^{-1}(\frac{w}{u}), \quad \beta = \sin^{-1}(\frac{v}{V_a}) \tag{2}$$

where $V_a = \sqrt{u^2 + v^2 + w^2}$ is the magnitude of the airspeed [21] and the superscript ^{B} indicates the body axis coordinate frame.

A. Synthetic Airspeed Estimation

According to [23], the acceleration in the z-direction of the body axes, a_z, can be defined as $a_z = \frac{\rho V_a^2}{2m} S C_z$, where m and S denote the mass and area of the wing, respectively; ρ is the local air density and C_z is the lift coefficient in the z-direction. Substituting $C_z = -C_L \cos\alpha - C_D \sin\alpha$, a_z is expanded as follows:

$$a_z = \frac{\rho V_a^2}{2m} S \Big(-C_L \cos\alpha - C_D \sin\alpha \Big) \tag{3}$$

where C_L and C_D refer to the lift and drag coefficients, respectively, which depend primarily on the AOA α.

First, in (3), C_L can be represented by the summation of four major lift coefficients as follows:

$$C_L = C_{L_0} + \alpha\, C_{L_\alpha} + \delta_e\, C_{L_{\delta_e}} + \frac{c}{2V_a} q\, C_{L_q} \tag{4}$$

where C_{L_0} and C_{L_α} denote the constant and linear lift coefficients, respectively. c denotes the mean aerodynamic chord of the wing. $C_{L_{\delta_e}}$ and C_{L_q} are lift coefficients with respect to elevator deflection δ_e and pitch rate q, respectively. c denotes the mean aerodynamic chord of the wing.

In (4), because most fixed-wing UAVs are designed to fly with nonaggressive maneuvers, the effect of pitch rate, C_{L_q}, can thus be neglected, yielding

$$C_L \approx C_{L_0} + \alpha\, C_{L_\alpha} + \delta_e\, C_{L_{\delta_e}}. \tag{5}$$

Next, the drag coefficient C_D in (3) can be represented in terms of α with the following nonlinear relationship [23]:

$$C_D = C_{D_p} + \frac{\left(C_{L_0} + \alpha\, C_{L_\alpha}\right)^2}{\pi e \mathrm{AR}} \tag{6}$$

where C_{D_p} is the constant drag coefficient from the parasitic drag effect, $\mathrm{AR} = b^2/S$ is the wing aspect ratio, b is the wingspan, and e is the Oswald efficiency factor. The last term in (6) represents the induced drag effect.

The small α assumption is generally reasonable because the flight trajectory of a fixed-wing UAV usually does not include aggressive flights in which α is large [24]. Thus, assuming a small angle approximation for α in (6), the following linear relationship is obtained for the drag coefficient:

$$C_D \approx C_{D_0} + \alpha C_{D_\alpha} \tag{7}$$

where C_{D_0} and C_{D_α} denote the constant and linear drag coefficients, respectively.

With a small angle approximation for α (i.e., $\sin\alpha = \alpha$ and $\cos\alpha = 1-\alpha^2/2$) and substituting (5) and (7) into (3) yields:

$$a_z \approx -\frac{\rho V_a^2}{2m} S\Big(C_{L_0} + \alpha(C_{L_\alpha} + C_{D_0}) + \delta_e C_{L_{\delta_e}} + \alpha^2(C_{D_\alpha} - (C_{L_0} + C_{L_{\delta_e}} + \alpha C_{L_\alpha})/2)\Big). \tag{8}$$

Additionally, it is reasonable to assume nonaggressive flight maneuvers (i.e., $|\alpha| \leq 5°$) so that the α^2 term is negligible and that the C_{L_0} coefficient is much smaller than C_{L_α}, allowing for the term $\alpha^2 \cdot C_{L_0}/2$ to be neglected in (8). With these reasonable approximations, the z-axis acceleration can be simplified with the following equation:

$$a_z \approx -\frac{\rho V_a^2}{2m} S\left(C_{L_0} + \alpha C_{L_\alpha} + \delta_e C_{L_{\delta_e}}\right). \tag{9}$$

Rearranging (9), the synthetic airspeed measurement, $\tilde{V}_{a,\text{syn}}$, can be expressed as:

$$\tilde{V}_{a,\text{syn}} \approx \sqrt{-\frac{2ma_z}{\rho S\left(C_{L_0} + \alpha C_{L_\alpha} + \delta_e C_{L_{\delta_e}}\right)}}. \tag{10}$$

where the symbol $\tilde{\cdot}$ denotes a synthetic measurement that differs from the actual measurement. From (10), it can be deduced that $\tilde{V}_{a,\text{syn}}$ is affected by the acceleration on the z-axis and the lift force. More specifically, assuming that the lift coefficients (i.e., C_{L_0}, C_{L_α}, and $C_{L_{\delta_e}}$) are available, the synthetic airspeed $\tilde{V}_{a,\text{syn}}$ can be computed accordingly.

B. Second Order CF for Airspeed Estimation

The dynamic equation for the airspeed state V_a can be defined in terms of measured accelerations by [25]:

$$\dot{V}_a = a_x \cos\alpha\cos\beta + a_y \sin\beta + a_z \sin\alpha\cos\beta + \mathbf{g}\Big(-\cos\alpha\cos\beta\sin\theta + \sin\beta\sin\phi\cos\theta + \sin\alpha\cos\beta\cos\phi\cos\theta\Big) \tag{11}$$

where $\mathbf{g}$ represents the acceleration due to gravity (i.e., $9.80665\,\text{m/s}^2$); a_x and a_y denote the acceleration in the x- and y-directions, respectively; and ϕ and θ are the roll and pitch of a UAV, respectively.

The complementary filter (CF), which consists of second-order high-pass and low-pass filters, is another attractive method to reconstruct airspeed with low computational complexity. In summary, given $[a_x \;\; a_y \;\; a_z]^\intercal$, $[\phi \;\; \theta]^\intercal$, $[\alpha \;\; \beta]^\intercal$, and δ_e, the proposed second-order CF, the airspeed $V_{a,\text{cf}}$, can be reconstructed by integrating the high-frequency part of $\dot{V}_a$ and the low-frequency synthetic measurements $\tilde{V}_{a,\text{syn}}$ as follows [1], [26]:

$$V_{a,\text{cf}} = \frac{s}{s^2 + 2\zeta_{\text{cf}} w_{\text{cf}} s + w_{\text{cf}}^2}\dot{V}_a + \frac{2\zeta_{\text{cf}} w_{\text{cf}} s + w_{\text{cf}}^2}{s^2 + 2\zeta_{\text{cf}} w_{\text{cf}} s + w_{\text{cf}}^2}\tilde{V}_{a,\text{syn}} \tag{12}$$

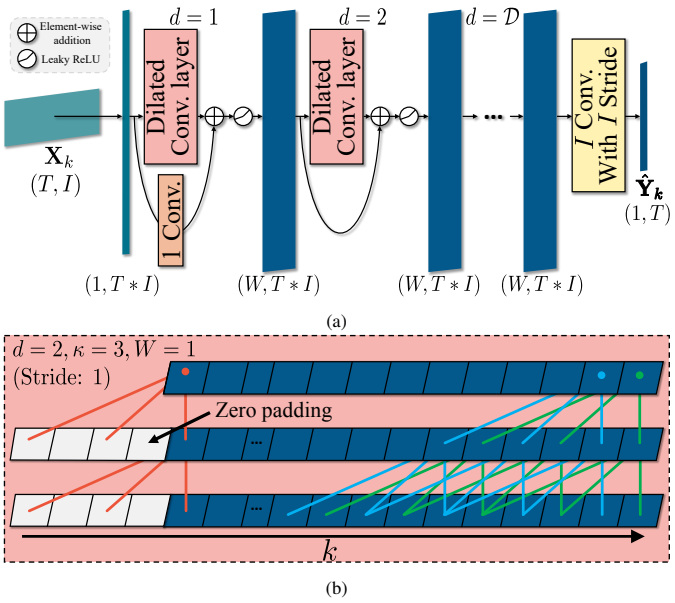

Fig. 2. (a) Overview of our temporal convolutional network (TCN) that consists of dilated convolutional layers and skip connections, each of which is followed by elementwise addition and a leaky rectified linear unit (ReLU). (b) Visual description of the operation of each dilated convolutional layer. The same colored lines represent the pipeline between the corresponding input and output layers (best viewed in color).

where ζ_{cf} and w_{cf} denote the damping ratio and break frequency of CF, respectively. In (12), $\tilde{V}_{a,\text{syn}}$ and $\dot{V}_a$ are computed by utilizing (10) and (11), respectively. The tuning of CF involves only two parameters, ζ_{cf} and w_{cf}, which determine the bandwidth and frequency responses of the filter, respectively.

III. End-to-End Synthetic Airspeed Estimation

A. Limitations of Conventional Methods

The aforementioned synthetic airspeed estimations work reasonably well in general, yet there are four limitations. First, the methods are based on the premise that UAVs operate with a small pitch rate so that the C_{L_q} term is negligible; however, this assumption sometimes fails in regard to soaring or dive flying. Second, in a drastic dive flying situation, a_z could become positive, so the term in the root in (10) becomes negative, which leads to catastrophic SADS failures. Third, the small angle approximation could induce nonnegligible errors when UAVs meet severe turbulence, which sometimes leads to large α values. Finally, the lift coefficient parameters (i.e., C_{L_0}, C_{L_α}, and $C_{L_{\delta_e}}$) are assumed to be already known, which are usually measured by wind tunnel experiments. Unfortunately, these experiments take considerable time, are expensive, and require field expertise. Furthermore, the parameters acquired always contain uncertainty and even change as time goes on for the reasons mentioned above (i.e., wear and tear, icing, or moisture on the wings or the sensor suite of UAVs) [3], [4], [27].

B. Temporal Convolutional Networks

To resolve these problems, we propose a learning-based method with TCN for robust synthetic airspeed estimation to

overcome the aforementioned limitations of the conventional approaches. TCN [18] has strong advantages over the typical variants of recurrent neural networks (RNNs), such as LSTM and the gated recurrent unit (GRU) [15], in that TCN avoids the exploding/vanishing gradient problem, which is a major issue of RNNs. In addition, TCN needs a small amount of memory when training [28]. Furthermore, empirically, TCN outperforms variants of RNNs in various sequence modeling tasks [18], and it shows promising results compared with other deep learning models with fewer floating point operations (FLOPS) on our synthetic airspeed estimation task (see Section VI.*C*).

As shown in Fig. 2(a), our TCN model consists of dilated convolutional layers and skip connections, each of which is followed by elementwise addition and leaky ReLU. Each dilated convolutional layer is comprised of two 1D *causal convolutions* with kernel size κ and dilation size 2^{d-1}, where d denotes the d-th depth level of the layer whose total depth is denoted as $\mathcal{D}$, as shown in Fig. 2(b). The causal convolution operates in a biased way to prevent future information from leaking into the past. That is, when convolved, the output at time step k is solely based on the elements within k to the earlier step, i.e., $k-(\kappa-1)\cdot 2^{d-1}$, in the previous layer. Note that the size of each hidden layer of the causal convolution is the same as that of the input layer, so zero-padding with a size of $(\kappa-1)\cdot 2^{d-1}$ is employed for the d-th layer.

C. Input/Output of TCN

It is sufficient to estimate V_a solely based on minimal input parameters, which consist of a_z, α, and δ_e in (10); however, in our previous study, it was verified that taking all available parameters as input rather than taking the minimal parameters resulted in better performance [20]. This is because a_z, α, and δ_e are not the only parameters affecting the estimation of V_a, so applying only these terms may lead to potential information loss. For this reason, the k-th input parameter, $\mathbf{m}_k$, whose total number of parameters is I (in our case, $I=11$), is defined as follows:

$$\mathbf{m}_k = \begin{bmatrix} a_{x,k} & a_{y,k} & a_{z,k} & p_k & q_k & r_k & \delta_{e,k} & \delta_{a,k} & \delta_{r,k} & \alpha_k & \beta_k \end{bmatrix} \in \mathbb{R}^{11} \tag{13}$$

where p_k, q_k, and r_k denote the angular rates of an IMU in the x, y, and z directions of the body frame at time k, i.e., roll rate, pitch rate, and yaw rate, respectively; $\delta_{a,k}$ and $\delta_{r,k}$ denote the aileron and rudder deflections, respectively. Accordingly, the input of the network, $\mathbf{X}_k$, is defined as follows:

$$\mathbf{X}_k = \mathcal{F}(\{\mathbf{m}_{k-T+1}, \mathbf{m}_{k-T+2}, \ldots, \mathbf{m}_k\}) \tag{14}$$

where T and $\mathcal{F}(\cdot)$ denote the k-th input parameters, which are the sequence length, and a normalization function, respectively. Note that $\mathcal{F}(\cdot)$ is an elementwise min-max scaler because the scale of each parameter is different. For example, $a_{x,k}$ ranges in value from -6.67 to 6.06 m/s^2, whereas $\delta_{a,k}$ ranges in value from -0.14 to 0.17 rad in our training data set.

When training the network, $\mathbf{X}_k \in \mathbb{R}^{B\times T\times I}$ where B is the batch size; however, $\mathbf{X}_k$ is flattened because TCN consists of 1D causal convolutions, as illustrated in Fig. 2(a). After that, TCN outputs $\hat{\mathbf{Y}}_k$, which is expressed as:

$$\hat{\mathbf{Y}}_k = \{\hat{V}^{\text{norm}}_{a,k-T+1}, \hat{V}^{\text{norm}}_{a,k-T+2}, \ldots, \hat{V}^{\text{norm}}_{a,k}\}. \tag{15}$$

In fact, our target (i.e., the k-th estimation, $\hat{V}^{\text{norm}}_{a,k}$) is a normalized term within the range of 0 to 1. Therefore, our final output $\hat{V}_{a,k}$ can be estimated after denormalization with $\mathcal{F}^{-1}(\cdot)$. That is, $\hat{V}_{a,\text{TCN}} = \mathcal{F}^{-1}(\hat{V}^{\text{norm}}_{a,k})$.

D. Training Loss

Since our goal is to achieve regression with TCN, by letting Θ and Θ^* be all parameters of our network model and the optimal parameters, the loss term is defined as follows:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \frac{1}{N-T+1} \sum_{k=T-1}^{N} \parallel \mathbf{Y}_k - \hat{\mathbf{Y}}_k \parallel_1 \tag{16}$$

where $\|\cdot\|_1$ denotes the L_1 loss term and N denotes the total number of sequences in the training data set. $\mathbf{Y}_k$ denotes the k-th normalized ground truth measurement set.

IV. Deep Learning-Aided Synthetic Airspeed Estimation Framework

A. UKF for Airspeed Estimation

The state $\mathbf{x}$, input $\mathbf{u}$, and measurement $\mathbf{z}$ for the proposed filter are defined as:

$$\begin{aligned} \mathbf{x} &= [V_a] \\ \mathbf{u} &= [a_x \;\; a_y \;\; a_z \;\; \phi \;\; \theta \;\; \alpha \;\; \beta \;\; \delta_e]^\intercal \\ \mathbf{z} &= \left[\hat{V}_a\right] \end{aligned} \tag{17}$$

and note that $\hat{V}_a$ can be replaced with $\tilde{V}_{a,\text{syn}}$ or $\hat{V}_{a,\text{TCN}}$ (see Section VI.*C*). The dynamic equation $\dot{V}_a$ for the airspeed state V_a can be obtained from (11).

In this paper, the nonlinear state-space equation in the discrete-time domain is considered as follows:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \tag{18}$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{19}$$

where k is the time index. $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^l$, and $\mathbf{z}_k \in \mathbb{R}^m$ are the state, system input, and measurement, respectively, and $\mathbf{f}_k(\cdot)$ and $\mathbf{h}_k(\cdot)$ are the state transition and measurement functions, respectively. The process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$ are assumed to be uncorrelated white Gaussian noise (WGN) as follows:

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{Q}_k) \\ \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{R}_k) \\ \mathrm{E}\left[\mathbf{w}_k \mathbf{v}_k^\intercal\right] &= 0 \end{aligned} \tag{20}$$

where $\mathcal{N}(\boldsymbol{\mu}_k, \sum_k)$ denotes a Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\sum_k$.

In (11), the state equation is given in a continuous-time nonlinear state space format as follows:

$$\dot{\mathbf{x}}_c = \mathbf{f}_c(\mathbf{x}_c, \mathbf{u}, \mathbf{w}) \tag{21}$$

where $\mathbf{x}_c$ denotes the state in a continuous domain and $\mathbf{f}_c$ is the nonlinear continuous-time state transition function. To

implement this equation in a discrete time UKF framework, a first-order discretization is used as follows:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{x}_c(kT_s)\\ \mathbf{x}_k &= \mathbf{x}_{k-1} + T_s\mathbf{f}_c(\mathbf{x}_{k-1},\,\mathbf{u}_{k-1},\,\mathbf{w}_{k-1})\\ &\equiv \mathbf{f}_k(\mathbf{x}_{k-1},\,\mathbf{u}_{k-1},\,\mathbf{w}_{k-1})\end{aligned} \tag{22}$$

where T_s is the sampling period.

In the UKF, the parameters are defined as follows [29]–[31]:

$$\begin{cases}\lambda_{\text{ukf}} &= \alpha_{\text{ukf}}^2(L_\text{s}+\kappa_{\text{ukf}}) - L_\text{s}\\ \Psi_0^m &= \lambda_{\text{ukf}}/(L_\text{s}+\lambda_{\text{ukf}})\\ \Psi_0^c &= \lambda_{\text{ukf}}/(L_\text{s}+\lambda_{\text{ukf}}) + 1 - \alpha_{\text{ukf}}^2 + \beta_{\text{ukf}}\\ \Psi_i^m &= \Psi_i^c = 1/[2\,(L_\text{s}+\lambda_{\text{ukf}})],\quad i = 1,\cdots,2L_\text{s}\end{cases} \tag{23}$$

where λ_{ukf} refers to an additional scaling parameter, L_s refers to the size of the augmented state vector, and Ψ_i^m and Ψ_i^c denote the weight of the mean and covariance corresponding to the i-th sigma point, respectively. α_{ukf}, β_{ukf}, and κ_{ukf} are the primary, secondary, and tertiary scaling parameters for the UKF. The spread of the sigma points is determined by the primary scaling factor: $10^{-4} \leq \alpha_{\text{ukf}} \leq 1$. Typical values for the secondary and tertiary parameters are $\beta_{\text{ukf}} = 2$ (assuming a Gaussian distribution) and $\kappa_{\text{ukf}} = 0$.

According to the nonadditive noise assumptions as in (18), the state vectors of the UKF can be extended with additional states to model nonadditive process noise as follows:

$$\mathbf{x}_k^a = \begin{bmatrix}\mathbf{x}_k\\ \mathbf{w}_k\end{bmatrix} \tag{24}$$

where $\mathbf{x}_k^a$ is the augmented state vector, which includes a process noise state $\mathbf{w}_k$ that retains a mean value of zero throughout the filter operation. However, this augmentation provides a means for introducing nonadditive process noise into the system through the spread in the sigma-points for these states with the block diagonal augmented error covariance $\mathbf{P}_k^a$, as follows:

$$\mathbf{P}_k^a = \begin{bmatrix}\mathbf{P}_k & 0\\ 0 & \mathbf{Q}_k\end{bmatrix}. \tag{25}$$

Then, a set of $2L_\text{s}+1$ sigma points $\boldsymbol{\chi}_{i,k-1}$ are generated according to the prior mean $\hat{\mathbf{x}}_{k-1|k-1}$, the augmented covariance $\mathbf{P}_{k-1|k-1}^a$ and the parameter λ_{ukf} as follows:

$$\begin{aligned}\boldsymbol{\chi}_{i,k-1} = \;& \hat{\mathbf{x}}_{k-1|k-1}^a\\ &+ \begin{cases}0 & \text{for } i = 0\\ \sqrt{(L_\text{s}+\lambda_{\text{ukf}})}\sqrt{\mathbf{P}_{k-1|k-1}^a} & \text{for } i = 1,\cdots,L_\text{s}\\ -\sqrt{(L_\text{s}+\lambda_{\text{ukf}})}\sqrt{\mathbf{P}_{k-1|k-1}^a} & \text{for } i = L_\text{s},\cdots,2L_\text{s}.\end{cases}\end{aligned} \tag{26}$$

Then, the UKF time update equations are given as [32]:

$$\begin{cases}\hat{\boldsymbol{\chi}}_{i,k|k-1} = f\left(\boldsymbol{\chi}_{i,k-1},\;\mathbf{u}_{k-1},\,0\right),\quad i = 0,\;1,\cdots,\;2L_\text{s}\\ \hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2L_\text{s}} \Psi_i^m\,\hat{\boldsymbol{\chi}}_{i,k|k-1}\\ \mathbf{P}_{k|k-1} = \sum_{i=0}^{2L_\text{s}} \Psi_i^c(\hat{\boldsymbol{\chi}}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\hat{\boldsymbol{\chi}}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})^\intercal\end{cases}$$

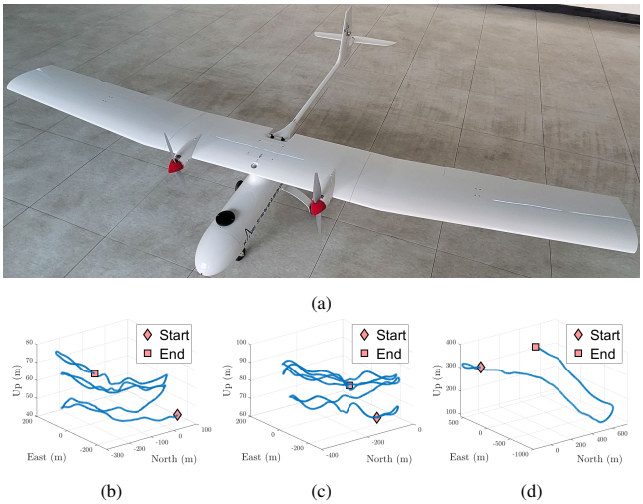

Fig. 3. (a) Our UAV system (b), (c), and (d) Trajectories of our test data set: sequence `1`, sequence `2`, and sequence `3`.

TABLE I
Geometric/inertial specifications for a UAV

Description	Symbol	UAV
Wingspan	b	2.6 m
Wing area	S	0.724 m^2
Mass	m	5.9 kg

where $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ are the predicted state and its covariance, respectively.

Then, the UKF measurement update equations are given as [32]:

$$\begin{cases}\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2L_\text{s}} \Psi_i^m\,\hat{\boldsymbol{\varphi}}_{i,k|k-1}\\ \mathbf{P}_{zz,k} = \sum_{i=0}^{2L_\text{s}} \Psi_i^c(\hat{\boldsymbol{\varphi}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})(\hat{\boldsymbol{\varphi}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})^\intercal\\ \qquad\qquad + \mathbf{R}_{k-1}\\ \mathbf{P}_{xy,k} = \sum_{i=0}^{2L_\text{s}} \Psi_i^c(\hat{\boldsymbol{\chi}}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\hat{\boldsymbol{\varphi}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})^\intercal\\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{zz,k}\mathbf{K}_k^\intercal\\ \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})\end{cases}$$

where

$$\begin{cases}\mathbf{K}_k = \mathbf{P}_{xy,k}\mathbf{P}_{zz,k}^{-1}\\ \hat{\boldsymbol{\varphi}}_{i,k|k-1} = h(\boldsymbol{\chi}_{i,k|k-1},\;\mathbf{u}_k,\;0),\quad i = 0,\;1,\cdots,\;2L_\text{s}.\end{cases}$$

V. Experiments

A. UAV Configuration

As shown in Fig. 3(a), our UAV, which was developed by the Korea Aerospace Research Institute (KARI), is equipped with an open-source Pixhawk autopilot system, an IMU (LSM303D integrated accelerometers/magnetometers and L3GD20 gyroscope), a barometer (MS5611, TE Connectivity), an airspeed sensor (MS4525DO, TE Connectivity), and a GPS receiver (NEO-M8N, u-blox) [33]. Note that the GPS data are not used in any of the presented estimation filters. The geometric/inertial specifications of the UAV are summarized in

Table I. The identified lift coefficient parameters of C_{L_0}, C_{L_α}, and $C_{L_{\delta_e}}$ are 0.0628, $0.2757\,\mathrm{rad}^{-1}$, and $2.9682\,\mathrm{rad}^{-1}$ by conducting a least squares analysis on flight data involving elevator flight maneuvers, respectively [34].

B. Flight Data Sets

Four sets of flight experiments with different altitudes were performed to acquire data in various situations. Consequently, a total of 107,824 samples of data were acquired. Three partial flight sequences were chosen as the test data set, namely, sequence `1`, sequence `2`, and sequence `3`, which are shown in Figs. 3(b)–3(d), respectively. In short, the data set was split into 84,919 samples for training and 22,905 samples for testing.

C. Training the Network

For training, the Adam optimizer [35] was utilized with a learning rate of 0.003, momentum of 0.9, and weight decay of 0.0001. We also set the the learning rate decay factor to 0.9 and the decay step to 1 (i.e., the decay occurs every epoch). Training was conducted for 40 epochs with a batch size of 96 on a NVidia TITAN Xp GPU.

D. Parameter Settings of the Deep Learning-aided UKF

The tuning of the UKF for flight experiments is very complicated due to the multiple parameters within the filter. In this paper, the sampling interval T_s was $0.02\,\mathrm{s}$, and process noise covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ were empirically set to $\mathrm{diag}([0.05\cdot \mathrm{I}_{3,3}\,(\mathrm{m/s^2})^2\,, 0.0001\cdot \mathrm{I}_{4,4}\,(\mathrm{rad})^2])$ and $1.5\,(\mathrm{m/s})^2$, respectively.

The initial state of V_a is calculated from the body fixed velocity $[u\ v\ w]^\intercal$, which is provided by the flight control computer as: $V_a = \sqrt{u^2+v^2+w^2}$. Then, the proposed filter starts when the airspeed V_a is greater than $5\mathrm{m/s}$, which means that the UAV has completed takeoff and started flying. The initial error covariance $\mathbf{P}$ was set to $0.1\,(\mathrm{m/s})^2$.

VI. Results and Discussion

A. Ablation Study

Impact of sequence length and depth level As shown in Fig. 4, it is observed that a larger $\mathcal{D}$ leads to performance improvement when the sequence length is large enough. This is due to the increase in nonlinearity, as well as the expansion of the receptive fields. That is, a larger $\mathcal{D}$ helps TCN learn intertemporal and intermodal relationships better because the dilated size becomes larger, so the receptive fields of the kernel in TCN are expanded. Additionally, as T becomes larger, the performance generally increases because the large size of T can provide networks with more temporal information.

However, it is noted that a saturation of performance occurs when $\mathcal{D}$ is large, yet T is somewhat short. This is attributed to the fact that the number of zero-padding dilutes the inherent information of the feature. When T is short, the ratio of zero-padding to input data is relatively large compared to that when T is large, so the number of meaningless floats increases.

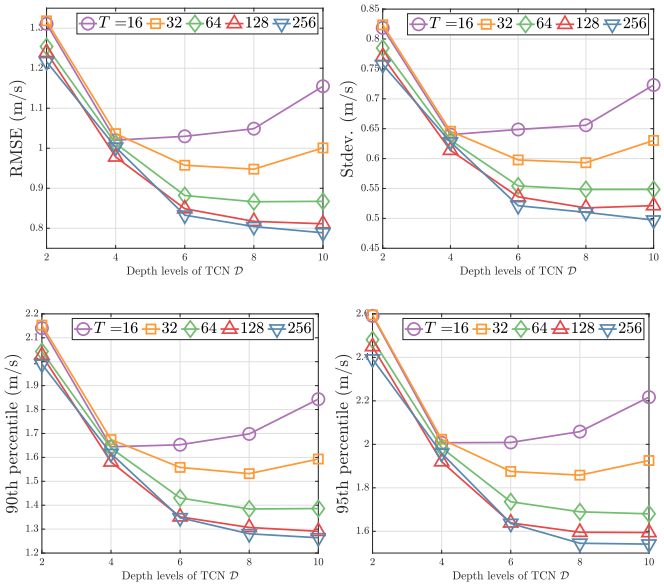

Fig. 4. Performance changes with varying sequence lengths for input T and depth levels for TCN $\mathcal{D}$ when the kernel size is set to $\kappa = 7$. (L-R, T-B): the RMSE, standard deviation, 90th percentile, and 95th percentile of the errors, respectively. For all metrics, *the lower the value is, the better the result is.*

Therefore, this implies that even though nonlinearity and receptive fields increase, an appropriate amount of temporal data is necessary for precise airspeed estimation. Consequently, T is set to 256 and $\mathcal{D}$ is set to 10.

Impact of kernel size After choosing T and $\mathcal{D}$, the effect of κ was analyzed. As shown in Table II, a larger κ gives rise to performance improvement because a larger κ increases the receptive field of TCN. Thus, it allows the network to learn better inter-temporal relationships. Therefore, κ is set to 7 for our framework.

B. Effectiveness of Fusion-based Framework

As shown in Table III, our deep learning-aided UKF improves the overall performance compared with when TCN is used alone. Therefore, the effectiveness of our fusion-based method is verified. While the UKF leads to only a minor improvement in performance, it plays an important role as a fail-safe strategy. In fact, the UAV system can collapse due to just one outlier measurement when only deep learning methods or synthetic airspeed methods are used. However, UKF is based on a Bayesian filter, so it makes the system more robust against any outliers. The significant decrease in the 99th percentile of errors supports this rationale.

C. Comparison with the State-of-the-art Methods

Finally, our fusion-based method was quantitatively compared with both the conventional and deep learning-based

TABLE II
Performance with different kernel sizes, κ, on our UAV data set (unit: m/s)

Kernel size	RMSE	Median	Stdev	90%	95%	99%
$\kappa = 3$	0.794	0.500	0.501	1.286	1.547	2.169
$\kappa = 5$	0.788	0.511	**0.489**	1.282	1.556	2.115
$\kappa = 7$	**0.785**	**0.493**	0.497	**1.264**	**1.524**	**2.082**

TABLE III
Comparison with the state-of-the-art methods. Stdev and 99% denote the standard deviation and the 99th percentile of errors, respectively. For RMSE, Stdev, and 99%, the unit is m/s and *the lower the metrics are, the better the result is*

Type	Method	FLOPS	Sequence 1			Sequence 2			Sequence 3			Total RMSE
			RMSE	Stdev	99%	RMSE	Stdev	99%	RMSE	Stdev	99%	
Conventional	Synthetic	-	1.421	0.920	4.230	1.935	1.134	4.832	3.011	1.587	6.561	2.235
	CF	-	1.238	0.785	3.466	1.817	1.029	4.151	2.823	1.473	6.140	2.077
	UKF w/ Synthetic	-	1.129	0.697	3.051	1.778	1.003	3.933	2.247	1.188	4.886	1.781
Learning-based	ResNet–18 [14]	11M	0.819	0.499	2.138	0.863	0.530	2.185	1.028	0.641	3.015	0.912
	ResNet–50 [14]	24M	0.854	0.527	2.093	0.962	0.603	2.571	0.983	0.657	3.377	0.938
	Bi–LSTM [15]	243K	0.744	0.457	1.950	0.795	0.482	2.145	0.930	0.607	2.942	0.829
	TCN (Proposed)	**85K**	0.728	**0.442**	1.919	0.768	0.469	2.012	0.851	0.565	2.624	0.785
Fusion	UKF w/ TCN (Proposed)	**85K**	**0.725**	0.446	**1.905**	**0.766**	**0.467**	**1.999**	**0.848**	**0.563**	**2.542**	**0.782**

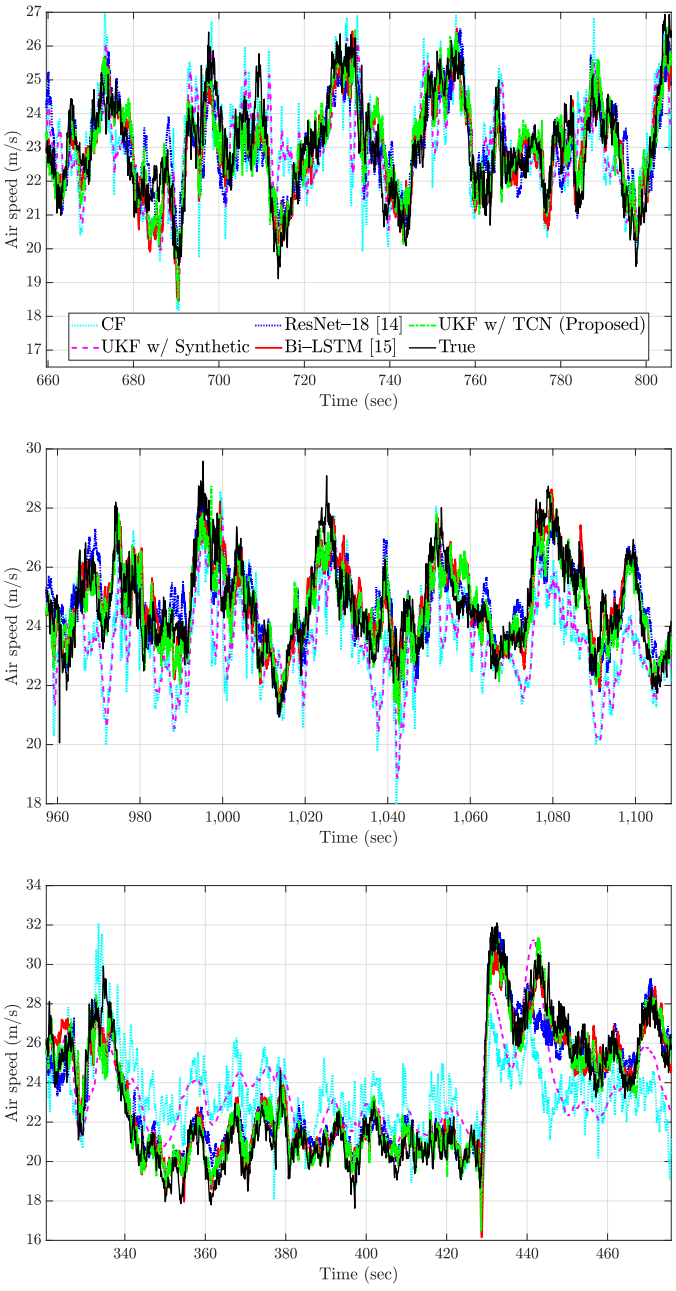

Fig. 5. (T-B): Synthetic airspeed estimation results on sequence 1, sequence 2, and sequence 3 (best viewed in color).

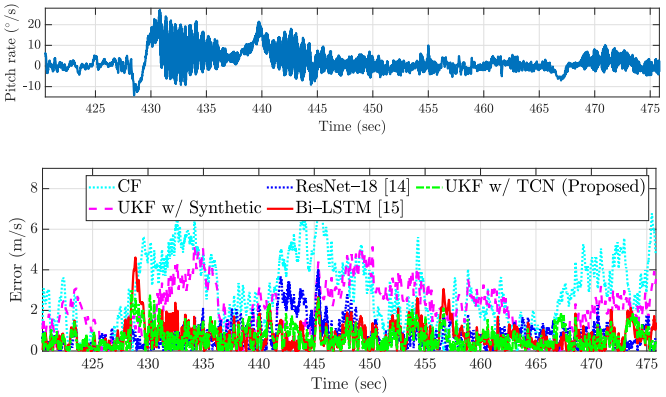

Fig. 6. (T-B): Pitch rate and error with respect to time in the harsh case. In soaring flight, it is not safe to assume that the lift coefficient term is negligible; thus, the errors of the conventional methods, i.e., CF and UKF w/Synthetic, become larger. In contrast, the deep learning-based methods show smaller errors due to their nonlinearity. In particular, our proposed method presents smaller errors than all the other methods.

methods, namely, *Synthetic* in (12), *CF* in (14), *UKF* that takes Synthetic as measurements, *ResNet* [14], and *Bi-LSTM* [15]. For learning-based methods, we set the shape of the first kernel in ResNet and Bi-LSTM to suit our task.

As shown in Table III and Fig. 5, other methods show reasonable airspeed estimations. However, the conventional methods struggle in some cases where their assumptions are no longer met, which is mainly represented in Fig. 6. In particular, the error approximately from 426 s to 440 s becomes larger because q grows and fluctuates; thus, the CF and UKF w/Synthetic show worse performance than the learning-based methods because the effect of pitch rate, C_{L_q}, was neglected in (5).

On the other hand, all learning-based methods show better performance than the conventional methods, overcoming the limitations mentioned in Section III.*A*. Among them, our fusion-based method shows the most promising results. It is noted that sometimes ResNet and Bi-LSTM have large errors, which are manifested as sharper spikes with respect to the ground truth, than those of our method, as represented in Figs. 5 and 6. In particular, ResNet is likely to be less sensitive when encountering sudden airspeed changes, which is presented approximately from 440 s to 444 s in sequence 3. Therefore, our proposed method is confirmed to be a more ap-

propriate and robust method for synthetic airspeed estimation and can help UAVs protect themselves against sensor failures even during aggressive flight maneuvers. Besides, our method only takes 6.7 msec for inference in offline mode because it requires a few FLOPS compared with other deep learning models [14], [15].

VII. Conclusions

In this study, a robust deep learning-based synthetic airspeed estimation method has been proposed. Our proposed method was demonstrated to overcome the limitations of conventional methods via the nonlinear estimation of the deep learning model. In particular, our method provides a precise synthetic airspeed with smaller error and variance in performance, which enables UAVs to utilize our proposed method as a backup system via software without a direct airspeed measurement system. Therefore, this paper has successfully demonstrated the feasibility of the proposed algorithm for analytical redundancy. In future works, we plan to investigate the effect of our deep learning-aided method in more severe cases or to devise a more sophisticated end-to-end method that is more robust against the partial disconnection of input signals.

References

[1] P. Tian, H. Chao, H. P. Flanagan, S. G. Hagerott, and Y. Gu, "Design and evaluation of UAV flow angle estimation filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 1, pp. 371–383, 2018.

[2] A. Wenz and T. A. Johansen, "Moving horizon estimation of air data parameters for UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, 2019.

[3] P. Freeman, P. Seiler, and G. J. Balas, "Air data system fault modeling and detection," *Control Eng. Pract.*, vol. 21, no. 10, pp. 1290–1301, 2013.

[4] P. Lu, L. Van Eykeren, E. Van Kampen, C. De Visser, and Q. Chu, "Adaptive three-step Kalman filter for air data sensor fault detection and diagnosis," *J. Guid. Control Dyn.*, vol. 39, no. 3, pp. 590–604, 2016.

[5] W. Youn, H. Choi, A. Cho, S. Kim, and M. B. Rhudy, "Aerodynamic model-aided estimation of attitude, 3D wind, airspeed, AOA, and SSA for high-altitude long-endurance UAV," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 6, pp. 4300–4314, 2020.

[6] A. Zolghadri, "The challenge of advanced model-based FDIR for real-world flight-critical applications," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 249–259, 2018.

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

[8] 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.

[9] K. Sun, C. D. Regan, and D. Gebre-Egziabher, "Observability and performance analysis of a model-free synthetic air data estimator," *J. Aircraft*, vol. 56, no. 4, pp. 1–16, 2019.

[10] K. Sun, H. Chang, J. Lee, J. Seo, Y. Morton, and S. Pullen, "Performance benefit from dual-frequency GNSS-based aviation applications under Ionospheric Scintillation: A new approach to fading process modeling," in *Proc. Int. Technical Meeting of The Institute of Navigation*, 2020, pp. 889–899.

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

[12] S. Hansen and M. Blanke, "Diagnosis of airspeed measurement faults for unmanned aerial vehicles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 1, pp. 224–239, 2014.

[13] M. L. Fravolini, M. Rhudy, S. Gururajan, S. Cascianelli, and M. Napolitano, "Experimental evaluation of two pitot free analytical redundancy techniques for the estimation of the airspeed of an UAV," *SAE Int. J. Aerosp.*, vol. 7, no. 2014-01-2163, pp. 109–116, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[15] H. Lim and H. Myung, "Effective indoor robot localization by stacked bidirectional LSTM using beacon-based range measurements," in *Proc. Int. Conf. Robot Intelligence Technology and Applications (RiTA)*, 2018, pp. 144–151.

[16] H. Lim, C. Park, and H. Myung, "RONet: Real-time range-only indoor localization via stacked bidirectional LSTM with residual attention," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems (IROS)*, 2019, pp. 3241–3247.

[17] D. Guo, M. Zhong, H. Ji, Y. Liu, and R. Yang, "A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors," *Neurocomputing*, vol. 319, pp. 155–163, 2018.

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

[19] S. Allison, H. Bai, and B. Jayaraman, "Wind estimation using quadcopter motion: A machine learning approach," *Aerosp. Sci. Technol.*, vol. 98, p. 105699, 2020.

[20] 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 Robot. Autom. Lett.*, 2021, *In Press*.

[21] M. B. Rhudy, Y. Gu, J. N. Gross, and H. Chao, "Onboard wind velocity estimation comparison for unmanned aircraft systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 55–66, 2017.

[22] M. R. Cohen and J. R. Forbes, "Navigation and control of unconventional VTOL UAVs in forward-flight with explicit wind velocity estimation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1151–1158, 2020.

[23] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ: Princeton University Press, 2012.

[24] Y. Mitikiri and K. Mohseni, "Attitude control of micro/mini aerial vehicles and estimation of aerodynamic angles formulated as parametric uncertainties," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2063–2070, 2018.

[25] V. Klein and E. A. Morelli, *Aircraft System Identification: Theory and Practice*. Reston, VA: American Institute of Aeronautics and Astronautics, 2006.

[26] W. T. Higgins, "A comparison of complementary and kalman filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 11, no. 3, pp. 321–325, 1975.

[27] P. Lu and E.-J. Van Kampen, "Aircraft inertial measurement unit fault identification with application to real flight data," in *Proc. AIAA Guidance, Navigation, and Control Conf.*, 2015, p. 0859.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for Semantic Segmentation," in *Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[29] M. Rhudy and Y. Gu, "Understanding nonlinear Kalman Filters part 1: Selection of EKF or UKF," *Interactive Robot. Lett.*, pp. 1–9, 2013.

[30] M. Rhudy and Y. Gu, "Understanding nonlinear Kalman Filters, part 2: An implementation guide," *Interactive Robot. Lett.*, pp. 1–18, 2013.

[31] W. Youn and H. Myung, "Robust interacting multiple model with modeling uncertainties for maneuvering target tracking," *IEEE Access*, vol. 7, pp. 65 427–65 443, 2019.

[32] J. Kim, "Cooperative localization and unknown currents estimation using multiple autonomous underwater vehicles," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2365–2371, 2020.

[33] W. Youn, M. B. Rhudy, A. Cho, and H. Myung, "Fuzzy adaptive attitude estimation for a fixed-wing UAV with a virtual SSA sensor during a GPS outage," *IEEE Sens. J.*, pp. 1456–1472, 2019.

[34] H. Chao, H. P. Flanagan, P. Tian, and S. G. Hagerott, "Flight test investigation of stall/spin detection techniques for a flying wing UAS," in *Proc. AIAA Atmospheric Flight Mechanics Conf.*, 2017, p. 1631.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," preprint 2014, arXiv:1412.6980.