2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

End-to-End Thermal Updraft Detection and Estimation for Autonomous Soaring Using Temporal Convolutional Networks

Christian Gall, Walter Fichter, and Aamir Ahmad

***Abstract*— Exploiting thermal updrafts to gain altitude can significantly extend the endurance of fixed-wing aircraft, as has been demonstrated by human glider pilots for decades. In this work, we present a novel end-to-end deep learning approach for the simultaneous detection of multiple thermal updrafts and the estimation of their properties — a key capability to let autonomous unmanned aerial vehicles soar as well. In contrast to previous works, our approach does not require separate algorithms for the detection of individual updrafts. Instead, a sequence of sensor measurements from a time window of interest can be directly fed into our temporal convolutional network, which estimates the position, strength, and spread of the encountered updrafts. We demonstrated in simulations that our approach can reliably detect updrafts solely based on measurements of the aircraft's position and the local vertical wind velocity. Nevertheless, our method can additionally make use of measurements of the roll moment induced by updrafts, which improves the precision further. Compared with a particle-filter-based method, we can determine the correct number of encountered updrafts with an accuracy of $99.99\,\%$ instead of $79.50\,\%$, significantly improve the precision of strength as well as spread estimates, and reduce the computational demand.**

I. INTRODUCTION

Autonomous electric unmanned aerial vehicles (UAVs) are today used for a variety of applications, such as wildlife monitoring or search and rescue purposes. However, the endurance of electric UAVs is limited. In the case of fixed-wing aircraft, the endurance can be significantly improved by exploiting vertical winds to gain altitude, also known as static soaring [1], [2], [3]. Consequently, the detection of thermal updrafts and the estimation of their properties, such as position, spread, and maximum vertical wind velocity as a measure of their strength, has been an active research area [4], [5], [6], [7], [8].

However, these existing works cannot estimate multiple updrafts simultaneously. A particle-filter-based approach [9] can provide this ability, but at the cost of a high computational demand, the necessity to additionally execute clustering algorithms, and the risk of unintentionally losing awareness of detected updrafts due to particle resampling issues — undesirable characteristics for autonomous soaring missions that require UAVs with limited computational resources to reliably keep track of several updrafts. Encouraged by recent successful applications of deep learning approaches to the fields of autonomous soaring and thermal updraft estimation [8], [10], we propose a novel end-to-end updraft estimator based on temporal convolutional networks (TCNs) [11] that addresses the above shortcomings.

All authors are with the Institute of Flight Mechanics and Controls, University of Stuttgart, 70569 Stuttgart, Germany {christian.gall, fichter, aamir.ahmad}@ifr.uni-stuttgart.de

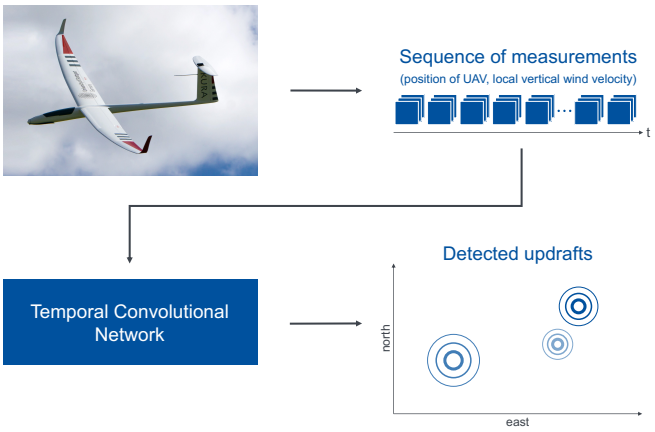

Fig. 1. Pictorial representation of the addressed task: A sequence of local vertical wind velocity and UAV position measurements is fed into a TCN that simultaneously detects and estimates multiple thermal updrafts.

The remainder of this paper is structured as follows: In Section II, we present related work. A formal definition of the problem depicted in Fig. 1 is presented in Section III, followed by an in-depth description of our proposed updraft estimation approach. In Section V, the results of conducted simulation experiments are presented and discussed. A brief summary and remarks on future work conclude this paper.

II. RELATED WORK

In this section, we present related work in the field of updraft estimation. As our approach is based on TCNs, we also cover seminal works about this network architecture.

A. Conventional Updraft Estimation

Existing works on updraft estimation can be categorized into model-free methods [12], [13], [14] and model-based methods [4], [5], [6], [7]. The former can map arbitrarily shaped updrafts, which may be beneficial for capturing their true complexity. The latter employ a simplified updraft observation model to enable a fast localization, assuming that the shape of the updraft velocity distribution resembles a bell curve. For the estimation, Kalman filters are a widely applied method [6], [7], [12], [13].

Our method does not rely on an updraft observation model, yet it is trained with data generated using such a model. Nevertheless, it could also be trained with real-world data.

B. Simultaneous Estimation of Multiple Updrafts

In contrast to the aforementioned model-based techniques, the particle filter presented by Notter et al. [9] shows the

unique ability to simultaneously estimate the properties of multiple updrafts. Assuming a bell-curve-like updraft velocity distribution as well, each particle consists of a hypothesis on the position, strength, and spread of the updraft. After resampling the particles by incorporating measurements of the local wind velocity, they are clustered with respect to their position hypotheses by applying the DBSCAN [15] algorithm to determine which particles belong to the same distinct updraft. In order to mitigate the risk of losing awareness of detected updrafts, Notter et al. suggested reducing the probability of resampling particles that have already been clustered.

Our method also allows estimating several updrafts at once. Since it has access to all measurements from the time window of interest at once, unintentionally losing awareness of updrafts can be ruled out.

C. Machine-Learning-Based Updraft Estimation

Machine learning methods, especially reinforcement learning, have been repeatedly examined for the autonomous soaring task [10], [16], [17], [18]. However, to the best of our knowledge, the artificial neural network presented by Zhang et al. in [8] is the only deep-learning-based approach to the explicit estimation of updraft properties. They applied a low-pass filter and a thresholding algorithm to sequences of simulated vertical wind velocity measurements in order to detect the presence of an updraft. Only then, they fed the sensor measurements from the subsequent 120 time steps into a network based on a Long Short-Term Memory (LSTM) unit to estimate a single updraft.

In contrast to [8], our approach does not require any separate low-pass filtering or thresholding algorithms to detect updrafts. Instead, we present an end-to-end learning approach. Moreover, we show that we can process significantly longer sequences with a length of $7\,000$ time steps.

D. Temporal Convolutional Networks

Convolutional neural networks have recently been applied to process time series data [11], [19], [20]. Bai et al. stated in [11] that such temporal convolutional networks are based on two principles: First, each layer outputs a sequence that shows the same length as the input sequence. Second, when calculating the values of the output sequence at time step t, the layers must only use data occurring at time step t or earlier in the input sequence.

Since the use case regarded in our work is not time series generation but the encoding of features from an input time series into a shorter output vector, we refrain from obeying these principles. Instead, our TCN is conceptually closer to the acausal encoder from [20].

III. PROBLEM FORMULATION

In this work, we seek a method that takes as input a sequence of measurements obtained by a fixed-wing UAV within a time window of interest and estimates the updrafts that were encountered within that window. This section provides a formal definition of this problem along with an overview of the assumed glider and updraft characteristics.

A. Glider Model in the Presence of Arbitrary Wind

For the simulation of a fixed-wing UAV, the three-degrees-of-freedom equations of motions that describe the dynamics of a glider in an arbitrary wind field presented in [21] are used. The flight-mechanical state $\boldsymbol{x} \in \mathbb{R}^6$ comprises the glider's position $\boldsymbol{p}_\mathrm{g} \in \mathbb{R}^3$ and its flight path velocity $\boldsymbol{v}_\mathrm{K} \in \mathbb{R}^3$, both with respect to the local Earth-fixed, north-east-down coordinate system. The control vector $\boldsymbol{u} \in \mathbb{R}^2$ comprises the angle of attack α and the air-relative bank angle μ_A.

Using basic equations of motions on the assumption of a flat, non-rotating Earth, the glider dynamics are described by

$$\dot{\boldsymbol{p}}_\mathrm{g} = \boldsymbol{v}_\mathrm{K} \tag{1}$$

$$\dot{\boldsymbol{v}}_\mathrm{K} = \frac{1}{m}\left(\boldsymbol{f}_\mathrm{G} + \boldsymbol{f}_\mathrm{A}\right) \tag{2}$$

with the glider mass m, the gravitational force $\boldsymbol{f}_\mathrm{G}\left(m, g\right)$, and the aerodynamic force $\boldsymbol{f}_\mathrm{A}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}_\mathrm{W}\right)$, where g denotes the gravitational acceleration and $\boldsymbol{v}_\mathrm{W}$ an arbitrary wind velocity vector. The flight path azimuth χ of the UAV is obtained with $\chi = \arctan_2\left(v_\mathrm{K,e}, v_\mathrm{K,n}\right)$, using the east and north component of the flight path velocity, respectively. For an in-depth description of this glider model, the reader is referred to [21].

B. Thermal Updraft Model

For the simulation of a wind field with thermal updrafts and its effect on the UAV, the model presented by Oettershagen et al. in [7] is employed. They modeled the local vertical wind velocity w, the down component of the wind velocity vector $\boldsymbol{v}_\mathrm{W}$, caused by a thermal updraft as a bell curve:

$$w\left(r_\mathrm{n}, r_\mathrm{e}\right) = -w_\mathrm{u} \cdot \exp\left(-\frac{r_\mathrm{n}^2 + r_\mathrm{e}^2}{r_\mathrm{u}^2}\right) \tag{3}$$

with the maximum vertical wind velocity at the updraft center w_u, the spread of the updraft r_u, and its positions

$$r_\mathrm{n} = p_\mathrm{g,n} - p_\mathrm{u,n}, \quad r_\mathrm{e} = p_\mathrm{g,e} - p_\mathrm{u,e} \tag{4}$$

relative to the glider in the north and east direction, respectively. The absolute updraft positions are analogously denoted by $p_\mathrm{u,n}$ and $p_\mathrm{u,e}$. Consistent with [8], we neglect horizontal winds.

Updrafts induce the moment L_u around the roll axis of the UAV since the wing that is closer to the updraft is lifted more strongly. Oettershagen et al. described this effect using the equation

$$L_\mathrm{u} = -\frac{1}{12} \cdot \frac{\mathrm{d}c_\mathrm{L}}{\mathrm{d}\alpha} \cdot \rho \cdot v_\mathrm{A} \cdot c \cdot b^3 \cdot \frac{w_\mathrm{u}}{r_\mathrm{u}^2} \cdot \exp\left(-\frac{r_\mathrm{n}^2 + r_\mathrm{e}^2}{r_\mathrm{u}^2}\right) \cdot \cos\left(\phi\right) \cdot \left[\cos\left(\psi\right) \cdot r_\mathrm{e} - \sin\left(\psi\right) \cdot r_\mathrm{n}\right] \tag{5}$$

with the wingspan b, the wing's chord length c, the air density ρ, the lift-curve slope $\frac{\mathrm{d}c_\mathrm{L}}{\mathrm{d}\alpha}$, and the absolute value of the air-relative velocity v_A. The roll and yaw angle are denoted by ϕ and ψ, respectively. In the course of this paper, we use the flight-mechanical approximations $\phi \approx \mu_\mathrm{A}$ and $\psi \approx \chi$. For a thorough derivation of (5), the reader is referred to [7].

Since we examine wind fields exhibiting multiple thermal updrafts, we calculate the resulting total vertical wind velocity and induced roll moment by accumulating the contributions of individual thermal updrafts:

$$\bar{w} = \sum_{i=0}^{n_\mathrm{u}} w_i, \quad \bar{L}_\mathrm{u} = \sum_{i=0}^{n_\mathrm{u}} L_{\mathrm{u},i}, \tag{6}$$

where n_u is the number of thermal updrafts in the wind field; w_i and $L_{\mathrm{u},i}$ are the vertical wind velocity and induced roll moment caused by the i-th updraft, as modeled by (3) and (5).

C. Updraft Detection and Estimation

Notter et al. derive in [9] that a sequence of measurements is required in order to achieve observability for the task of updraft localization from scalar measurements, such as the local vertical wind velocity. Consequently, we regard a sequence

$$\boldsymbol{X} = (\boldsymbol{x}_1\,,\, \boldsymbol{x}_2\,,\, \boldsymbol{x}_3\,,\, \dots\,,\, \boldsymbol{x}_L) \tag{7}$$

of length L, where each element $\boldsymbol{x}_t$ is a vector of measurements

$$\boldsymbol{x}_t = \left(p_{\mathrm{g,n},t},\ p_{\mathrm{g,e},t},\ \bar{w}_t\right)^\intercal \in \mathbb{R}^3 \tag{8}$$

consisting of the UAV's north position $p_{\mathrm{g,n},t}$, east position $p_{\mathrm{g,e},t}$, and the vertical wind velocity $\bar{w}_t$ it experiences at time step t. The roll moment induced by updrafts $\bar{L}_{\mathrm{u},t}$ may be considered as an additional measurement, similar to [8]. In this case, we obtain the measurement vector

$$\tilde{\boldsymbol{x}}_t = \left(p_{\mathrm{g,n},t},\ p_{\mathrm{g,e},t},\ \bar{w}_t,\ \bar{L}_{\mathrm{u},t}\right)^\intercal \in \mathbb{R}^4. \tag{9}$$

The problem then demands to estimate the north-east position $\boldsymbol{p}_\mathrm{u} \in \mathbb{R}^2$ of up to n_u updrafts, the maximum vertical wind velocity that can be experienced at their center w_u, and the spread of the updraft specified by r_u:

$$\hat{\boldsymbol{Y}} = \left(\hat{\boldsymbol{y}}_1\,,\, \hat{\boldsymbol{y}}_2\,,\, \hat{\boldsymbol{y}}_3\,,\, \dots\,,\, \hat{\boldsymbol{y}}_{n_\mathrm{u}}\right) \quad \text{with } \hat{\boldsymbol{y}}_i = \begin{pmatrix} \hat{\boldsymbol{p}}_\mathrm{u} \\ \hat{w}_\mathrm{u} \\ \hat{r}_\mathrm{u} \end{pmatrix}^\intercal \in \mathbb{R}^4 \tag{10}$$

Thus, we seek a function with parameters $\boldsymbol{\theta}$ that takes the sequence of sensor measurements $\boldsymbol{X}$ as input and produces the mapping

$$\hat{\boldsymbol{Y}} = f\left(\boldsymbol{X}, \boldsymbol{\theta}\right). \tag{11}$$

IV. Temporal Convolutional Network for Updraft Detection and Estimation

We propose to approximate $f\left(\boldsymbol{X}, \boldsymbol{\theta}\right)$ with a TCN and to find its parameters $\boldsymbol{\theta}$ using supervised learning. To be able to tailor the network architecture to a concrete use case, we assume a time window of interest for the updraft detection covering the last 700 seconds of flight, in which we expect to encounter up to three updrafts. At a sensor sampling rate of ten Hertz, this requires the TCN to process sequences of sensor measurements with $L = 7\,000$ time steps. In order to reduce these sequences to a single vector containing the estimated updraft properties, we designed the artificial neural network depicted in Fig. 2. In this section, we explain its architecture in detail and outline training specifics.

A. Architecture

The core of the proposed network forms a TCN consisting of six one-dimensional convolutional layers. First, a kernel of length one is utilized to transform the sensor measurements to a 32-dimensional space. Next, the transformed sequence is fed into a layer with a kernel length of ten, where the kernel is moved across the sequence by a stride value of one time step at a time. Although this barely reduces the sequence length, it may allow the network to detect fine-grained patterns that indicate updrafts at full temporal resolution. Only now, a series of three layers with increasing stride value incrementally condenses the sequence from a length of $6\,991$ to 108 time steps, followed by a sixth convolutional layer with a stride value of one to process the data of the condensed sequence in detail. Finally, the data is flattened and fed into a series of three fully connected layers, which eventually output a twelve-dimensional vector containing the estimated properties of up to three updrafts.

Each layer, except for the last one, is followed by a layer normalization [22] and ReLU activation layer. To reduce the number of network parameters in favor of an efficient computation, we apply the layer normalization without the per-element affine transformations proposed in [22].

B. Training Specifics

We trained the TCN for 100 epochs by performing gradient descent on mini-batches of size 64 using Adam [23] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon =$ 1e-8. The input data was normalized by scaling it to the range [0, 1]. In order to reduce overfitting, the loss function was formed by adding an L2 penalty of 1e-5 to the mean square error between the output vectors and vectors containing the normalized true updraft properties. As listed along with the other hyperparameters in Table I, we set the initial learning rate to 1e-3. We then applied a learning rate decay after 50 and 95 epochs, when we reduced the learning rate to a tenth each. All weights were initialized by sampling from a Gaussian distribution as proposed by He et al. in [24]; bias parameters were initially set to zero. We further empirically found that clipping the gradient at a norm of one helped stabilize the training.

Since the TCN outputs a fixed-length vector with space for estimates of three updrafts although the number of encountered updrafts can vary, we task the network to perform output padding. Precisely, the output vector is padded with minus ones if less than three updrafts are detected. This also enables determining the number of detected updrafts by counting the number of minus ones. In order to facilitate

TABLE I
Training Hyperparameters

Hyperparameter	Value
Epochs	100
Mini-batch size	64
Initial learning rate	1e-3
L2 penalty	1e-5
Gradient norm clipping value	1

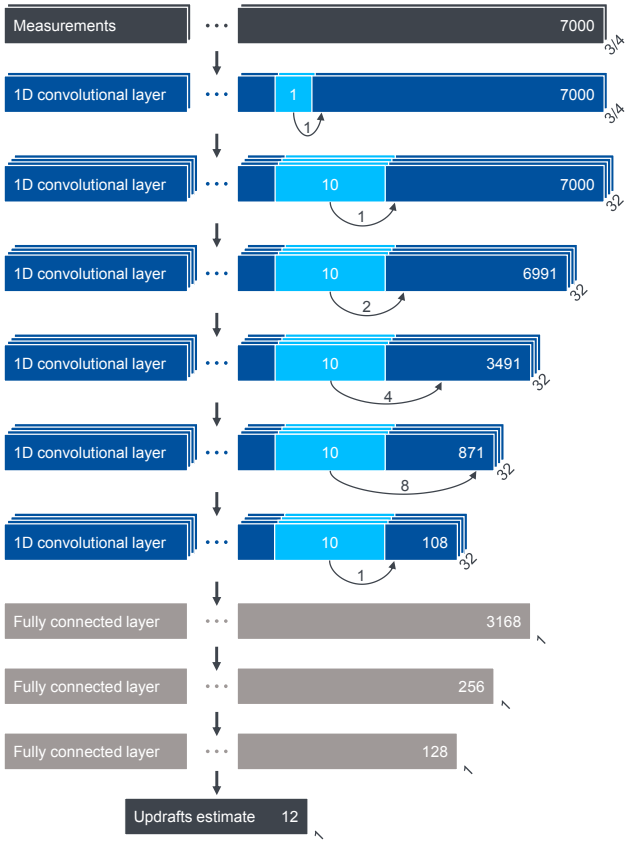

Fig. 2. Architecture of the proposed TCN. The horizontal numbers on the right represent the sequence length, which is not depicted true to scale; the diagonal numbers indicate the number of dimensions.

training, we ensure that there is only one correct output vector per training sequence. This is achieved by tasking the network to output the updraft estimates sorted by the updraft strength, placing the properties of the strongest updraft in the output vector first.

V. EXPERIMENTS

Labeled data containing the true position, strength, and spread of real-world updrafts is unavailable. We believe that such a dataset could, for instance, be created by manually labeling recorded flight data. However, this lies beyond the scope of this work, and we have previously shown in flight experiments that a sim-to-real transfer can be successfully performed in the regarded domain [10]. For these reasons, we trained and tested our TCN-based estimator in simulations.

After providing an outline of the experiment setup, we first examine if incorporating induced roll moment measurements proves beneficial. Subsequently, we present an in-depth examination of the proposed method. We conclude this section with a comparison of our TCN-based estimator with the particle filter from Notter et al. [9] since it can also simultaneously estimate multiple updrafts.

A. Experiment Setup

For the simulation, we considered a $1\,000 \times 1\,000$ meters wide field in which we arbitrarily distributed up to three random updrafts, which were modeled as described in Section III-B. The maximum vertical wind velocity and the spread of the updrafts were randomly sampled from continuous uniform distributions with the intervals specified along with the other experiment parameters in Table II, whereas the number of updrafts n_u was drawn from a discrete distribution with

$$P(n_\mathrm{u}=1)=0.2,\ P(n_\mathrm{u}=2)=0.3,\ P(n_\mathrm{u}=3)=0.5. \tag{12}$$

The only constraint for the random updraft positions was a minimum distance of 200 meters between the individual updrafts. The measurements of the vertical wind velocity $\bar{w}$ and the roll moment induced by updrafts $\bar{L}_\mathrm{u}$ were simulated by adding zero-mean Gaussian noise with the standard deviations (SD) σ_w and $\sigma_{\mathrm{L_u}}$ to the values obtained with (3) and (5), respectively.

Motivated by the observability analysis for the updraft detection task in [9], we guided the UAV along a zigzag lawnmower pattern using a proportional waypoint tracking controller to control the bank angle μ_A. In order to introduce variance, this path was randomly flipped and rotated by 90 degrees, as depicted in Fig. 3. The UAV, which was modeled according to the description in Section III-A, was spawned at the first waypoint of the path at an altitude of 350 meters, facing the path's second waypoint with an initial air-relative velocity of ten meters per second. The angle of attack was fixed to a value of 6.3 degrees, causing the UAV to reach the last waypoint within approximately 700 seconds, even if no updrafts are present.

Using the above simulation setup, we generated a dataset of 240 000 sequences of length 7 000 at a sampling rate of ten Hertz, using new random parameters for every new sequence. We split the dataset into training, validation, and test sets

TABLE II
EXPERIMENT PARAMETERS

Parameter	Symbol	Unit	Value
Glider model			
Mass	m	kg	4.5
Wingspan	b	m	4.3
Chord length	c	m	0.18
Lift-curve slope	$\mathrm{d}c_\mathrm{L}/\mathrm{d}\alpha$	$1/\mathrm{deg}$	0.1
Oswald efficiency number	e_0	-	0.95
Zero-lift drag coefficient	c_{D0}	-	0.015
Thermal updraft model			
Number of updrafts interval	n_u	-	$[1, 3]$
Max. vertical wind velocity interval	w_u	m/s	$[2, 10]$
Spread interval	r_u	m	$[50, 100]$
Environment			
Gravitational acceleration	g	$\mathrm{m/s^2}$	9.81
Air density	ρ	$\mathrm{kg/m^3}$	1.225
Simulation			
Sampling frequency	-	Hz	10
SD for $\bar{w}_u$ measurement noise	σ_w	m/s	0.05
SD for $\bar{L}_u$ measurement noise	$\sigma_{\mathrm{L_u}}$	Nm	0.06

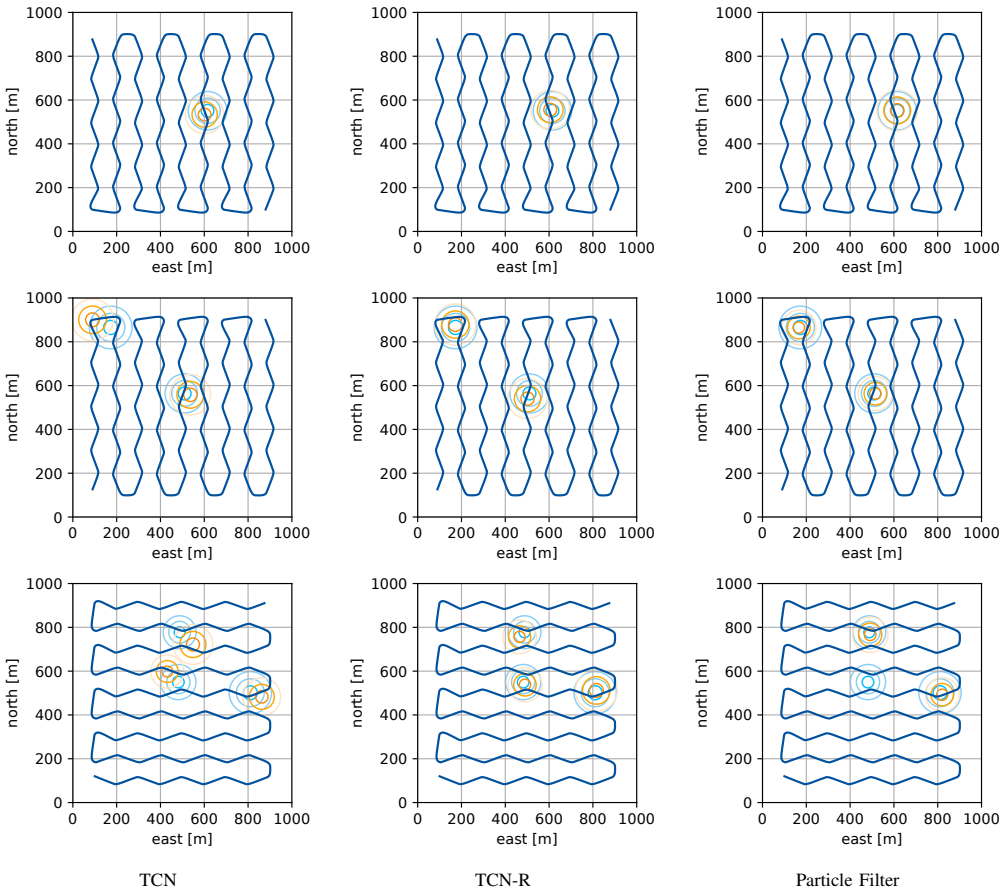

Fig. 3. Simulation results comparing the proposed *TCN*, *TCN-R*, and the particle filter from [9]. Orange circles mark the estimated position and spread of detected updrafts; light blue circles show their true position and spread. The flight path of the UAV is shown in dark blue.

with the ratio 4:1:1. The simulation was written in Python; PyTorch was used as the machine learning framework.

The particle filter [9] was implemented using the original code published by Notter et al. and the parameters listed in [9], except for the adjustment of the random sampling of the particles to match the expected distributions of the updrafts' position, strength, and spread in our simulation.

B. Effect of Incorporating Roll Moment Measurements

Some works on updraft estimation and autonomous soaring, such as [7], [18] or the LSTM-based network from Zhang et al. [8], incorporate a measurement of the roll moment induced by updrafts to improve observability. This motivated us to examine the effect of additionally feeding these measurements into our proposed TCN as well.

We distinguish between the *TCN*, which only takes measurements of $p_{\mathrm{g,n}}$, $p_{\mathrm{g,e}}$, and $\bar{w}$ as input, and *TCN-R*, which additionally uses $\bar{L}_{\mathrm{u}}$, as specified in (8) and (9), respectively. Both network architectures are identical, except for the first layer, which either transforms three- or four-dimensional measurement vectors to a 32-dimensional space.

As Table III shows, incorporating roll moment measurements reduced the mean absolute error (MAE) between the estimated and the true updraft properties from the test set by approximately $30\,\%$. Hence, we deem incorporating these measurements as desirable and consequently examine *TCN-R* in more detail below. Nevertheless, based on the map-like visualization of the estimates in Fig. 3, we assess the updraft estimation using *TCN* as a feasible alternative if roll moment measurements are unavailable and accuracy is not critical.

C. In-Depth Examination

In order to examine *TCN-R* in more detail, we evaluated the MAE in dependence on n_{u}. Table IV shows that the MAE increased with an increasing number of encountered updrafts. For instance, the MAE of the updraft position in the north direction $\hat{p}_{\mathrm{u,n}}$ was reduced to less than a fourth if the properties of only a single encountered updraft needed to be estimated compared to three encountered updrafts.

In addition, for $n_{\mathrm{u}} \in \{2, 3\}$, we list the MAE for the strongest, second strongest, and least strong updraft sepa-

TABLE III
COMPARISON OF TCN WITH TCN-R

	Acc.	MAE			
Estimator	n_{u} [%]	$\hat{p}_{\mathrm{u,n}}$ [m]	$\hat{p}_{\mathrm{u,e}}$ [m]	$\hat{w}_{\mathrm{u}}$ [m/s]	$\hat{r}_{\mathrm{u}}$ [m]
TCN	99.985	38.397	38.191	0.247	3.005
TCN-R	**99.995**	**26.181**	**26.738**	**0.144**	**2.185**

TABLE IV
IN-DEPTH RESULTS OF TCN-R

Updraft(s)	MAE $\hat{p}_{\mathrm{u,n}}$ [m]	MAE $\hat{p}_{\mathrm{u,e}}$ [m]	MAE $\hat{w}_{\mathrm{u}}$ [m/s]	MAE $\hat{r}_{\mathrm{u}}$ [m]
$n_{\mathrm{u}} = 1$				
#1	**8.034**	**8.392**	**0.074**	**0.710**
$n_{\mathrm{u}} = 2$				
#1	**17.849**	**18.418**	**0.105**	**1.514**
#2	25.328	25.312	0.143	1.823
All	20.286	20.658	0.125	1.614
$n_{\mathrm{u}} = 3$				
#1	**26.851**	**27.394**	**0.136**	**2.234**
#2	56.372	56.111	0.186	3.935
#3	42.231	42.456	0.199	3.264
All	36.846	37.591	0.184	3.105

rately in Table IV. One can see that the strongest updraft, denoted by #1, was consistently estimated best. A likely reason for this is that every training sequence can be used to train estimating updraft #1. However, only a fraction of sequences contained information of a second or third updraft, as given in (12). Since we chose to task the TCN to output the detected updrafts sorted by the estimated maximum vertical wind velocity $\hat{w}_{\mathrm{u}}$ in descending order, we foster that the strongest updraft — and as a result the likely most relevant updraft for gaining altitude — is estimated best.

D. Comparison with a Particle Filter

Lastly, we compared the proposed TCNs with the particle filter (PF) from [9] using a subset of the test set containing $1\,000$ sequences. We let the particle filter process the sequences of measurements iteratively, one time step at a time. Once all $7\,000$ time steps have been processed, we compared its estimates with the TCNs' estimates. Since the particle filter does not output the detected updrafts in a defined order, we matched the detected updrafts with the true updrafts based on the minimal distance between their positions before the MAE calculation. To allow for an unbiased comparison, we applied the same procedure to the TCN estimates.

First, we consider a varying number of updrafts $n_{\mathrm{u}} \in [1, 3]$. As listed in Table V, the particle filter's MAE of the positions was approximately 16 meters smaller for the detected updrafts than *TCN-R*'s MAE. However, the particle filter did either not detect all encountered updrafts or lost awareness of previously detected updrafts, as can, for instance, be seen in the lower right map of Fig. 3. Whereas the TCNs detected the correct number of updrafts with an accuracy of $100.0\,\%$, the particle filter showed a lower accuracy of $79.5\,\%$. For the updraft strength estimate $\hat{w}_{\mathrm{u}}$ and the spread estimate $\hat{r}_{\mathrm{u}}$, the TCNs delivered superior results as well, reducing the MAE of $\hat{w}_{\mathrm{u}}$ by an order of magnitude and the MAE of $\hat{r}_{\mathrm{u}}$ by more than $63\,\%$.

Second, we regard the cases in which only one updraft was encountered. Here, we could observe position estimate

TABLE V
COMPARISON OF TCNS WITH PARTICLE FILTER

Estimator	Acc. n_{u} [%]	MAE $\hat{p}_{\mathrm{u,n}}$ [m]	MAE $\hat{p}_{\mathrm{u,e}}$ [m]	MAE $\hat{w}_{\mathrm{u}}$ [m/s]	MAE $\hat{r}_{\mathrm{u}}$ [m]
$n_{\mathrm{u}} = 1$					
PF	85.027	7.810	**8.890**	0.784	6.630
TCN	**100.000**	9.692	10.334	0.148	0.841
TCN-R	**100.000**	**7.697**	9.043	**0.071**	**0.668**
$n_{\mathrm{u}} \in [1, 3]$					
PF	79.500	**10.945**	**12.040**	1.194	8.508
TCN	**100.000**	39.424	39.002	0.250	3.135
TCN-R	**100.000**	27.640	27.631	**0.146**	**2.266**

MAEs for *TCN-R* that are on par with the particle filter's estimates. In this case, the strength and spread estimate MAE were both one order of magnitude smaller for the TCNs, as compared with the particle filter's MAE.

On a workstation with an Nvidia GeForce RTX 3080 graphics processing unit (GPU) and an AMD Ryzen threadripper 3960x 24-core central processing unit (CPU), processing a single time step with the particle filter took approximately 3e-2 seconds on the CPU without using parallel computing. Processing all $7\,000$ time steps at once with the TCNs took 4e-3 seconds on the CPU and 8e-4 seconds on the GPU — one and two orders of magnitude less than the particle filter, respectively. This conveys an initial insight into the low computational demand of the proposed approach.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a novel end-to-end deep learning approach to the simultaneous estimation of multiple thermal updrafts. Our proposed TCN takes a sequence of measurements from the past minutes of flight as input and estimates the location, strength, and width of updrafts that were encountered within this timespan. We demonstrated in simulations that measurements of the UAV's position and the local vertical wind velocity suffice to achieve this task, yet the accuracy can be improved by additionally feeding measurements of the roll moment induced by updrafts into the TCN. Our simulation results show that the presented TCN is a superior alternative to a particle-filter-based updraft estimation — especially if a reliable detection, an accurate estimate of the updrafts' strength as well as spread, and a low computational demand are key.

For future work, we intend to test the proposed TCN with real-world data and reuse parts of its network architecture as an updraft feature encoder for an end-to-end reinforcement learning approach to autonomous cross-country soaring.

ACKNOWLEDGMENT

The topic presented has been investigated within the project *Decision Making for Environmental Energy Exploitation with Small Aircraft*, funded by the Cyber Valley Research Fund (CyVy-RF-2021-21). The financial support is gratefully acknowledged.

REFERENCES

[1] J. A. Cobano, D. Alejo, S. Vera, G. Heredia, and A. Ollero, "Multiple gliding uav coordination for static soaring in real time applications," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 790–795.

[2] J. A. Cobano, D. Alejo, S. Sukkarieh, G. Heredia, and A. Ollero, "Thermal detection and generation of collision-free trajectories for cooperative soaring uavs," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2948–2954.

[3] J. J. Acevedo, N. R. Lawrance, B. C. Arrue, S. Sukkarieh, and A. Ollero, "Persistent monitoring with a team of autonomous gliders using static soaring," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4842–4848.

[4] M. Allen and V. Lin, "Guidance and control of an autonomous soaring vehicle with flight test results," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007, p. 867.

[5] D. J. Edwards and L. M. Silverberg, "Autonomous soaring: The montague cross-country challenge," *Journal of Aircraft*, vol. 47, no. 5, pp. 1763–1769, 2010.

[6] A. D. Kahn, "Atmospheric thermal location estimation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2363–2369, 2017.

[7] P. Oettershagen, T. Stastny, T. Hinzmann, K. Rudin, T. Mantel, A. Melzer, B. Wawrzacz, G. Hitz, and R. Siegwart, "Robotic technologies for solar-powered uavs: Fully autonomous updraft-aware aerial sensing for multiday search-and-rescue missions," *Journal of Field Robotics*, vol. 35, no. 4, pp. 612–640, 2018.

[8] Y. Zhang, K. Li, K. Li, and J. Liu, "Intelligent prediction method for updraft of uav that is based on lstm network," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 2, pp. 464–475, 2023.

[9] S. Notter, P. Groß, P. Schrapel, and W. Fichter, "Multiple thermal updraft estimation and observability analysis," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 3, pp. 490–503, 2020.

[10] S. Notter, C. Gall, G. Müller, A. Ahmad, and W. Fichter, "Deep reinforcement learning approach for integrated updraft mapping and exploitation," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 10, pp. 1997–2004, 2023.

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

[12] J. Bird and J. Langelaan, "Spline mapping to maximize energy exploitation of non-uniform thermals," *Technical Soaring*, vol. 37, no. 3, pp. 38–44, 2013.

[13] N. Depenbusch and J. Langelaan, "Coordinated mapping and exploration for autonomous soaring," in *Infotech@ Aerospace 2011*. American Institute of Aeronautics and Astronautics, 2011, p. 1436.

[14] N. R. J. Lawrance and S. Sukkarieh, "Path planning for autonomous soaring flight in dynamic wind fields," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2499–2505.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[16] C. Montella and J. R. Spletzer, "Reinforcement learning for autonomous dynamic soaring in shear winds," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3423–3428.

[17] G. Reddy, A. Celani, T. J. Sejnowski, and M. Vergassola, "Learning to soar in turbulent environments," *Proceedings of the National Academy of Sciences*, vol. 113, no. 33, pp. E4877–E4884, 2016.

[18] G. Reddy, J. Wong-Ng, A. Celani, T. J. Sejnowski, and M. Vergassola, "Glider soaring via reinforcement learning in the field," *Nature*, vol. 562, pp. 236–239, 2018.

[19] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016, arXiv:1609.03499.

[20] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1003–1012.

[21] S. Notter, F. Schimpf, G. Müller, and W. Fichter, "Hierarchical reinforcement learning approach for autonomous cross-country soaring," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 1, pp. 114–126, 2023.

[22] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, arXiv:1607.06450.

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

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.