

Real-time 3D wind field prediction onboard UAVs for safe flight in complex terrain

Philipp Oettershagen, Benjamin Müller, Florian Achermann and Roland Siegwart
Autonomous Systems Lab
Swiss Federal Institute of Technology Zurich (ETH Zurich)
Leonhardstrasse 21
8092 Zurich, Switzerland
+41 44 632 7395
philipp.oettershagen@mavt.ethz.ch

Abstract—Due to a lack of environment awareness, today’s low-altitude fixed-wing Unmanned Aerial Vehicles (UAVs) are limited to primitively follow user-defined waypoints. All high-level decision making is still performed by a human user. Fully-autonomous remote missions in complex environments however require true environment awareness both with respect to terrain and wind. While terrain-aware navigation is covered in existing literature, the real-time estimation and consideration of complex wind patterns onboard UAVs is not. This paper therefore presents the literature’s first-ever local 3D wind field prediction method which can run in real time onboard a UAV. The selected method is a simple downscaling approach which retrieves low-resolution data from global weather models and then adjusts the wind field via potential flow theory such that terrain boundaries, mass conservation, and the atmospheric stratification are observed. Typical 3D wind fields of 1 km^3 volume are calculated in below 10 seconds. Synthetic test cases such as the flow around a semi-cylinder, through a valley and over a ramp yield good results. A comparison with 3D LIDAR wind data collected over 10 days in the Swiss Alps shows an overall wind error reduction of 23% with respect to the zero-wind assumption that is mostly used for UAV path planning today. Overall, our initial research demonstrates the feasibility of real-time 3D wind field prediction onboard a UAV. However, the focus on low computation time means that the vertical wind prediction lacks accuracy. The paper therefore ends with a research outlook into real-time 3D wind field prediction through the fusion of machine learning techniques with Computational Fluid Dynamics (CFD) methods.

1. INTRODUCTION

The aerial scanning capabilities provided by Unmanned Aerial Vehicles (UAVs) are of significant benefit in search-and-rescue support, agricultural sensing, industrial inspection or border patrol [1]. However, today’s low-altitude fixed-wing UAVs are largely limited to a primitive set of waypoint-following tasks and have little awareness of the environment in which they fly. As a result, today, low-altitude UAV missions are mostly performed on a small scale. To penetrate into applications that can be of pivotal societal and commercial use, UAVs need to be capable of fully autonomous large-scale operations including beyond-visual-line-of-sight (BVLOS) conditions. There, cluttered terrain poses a significant risk not only because of a potential collision but also because of local meteorological effects: While rain, thunderstorms and global winds are large-scale effects that can be predicted and considered by Numerical Weather Prediction (NWP)-based planning systems [2], local turbulence including updrafts and downdrafts is usually caused by terrain features that are not resolved by the low-resolution NWP system. However, unexpected turbulence or downdrafts behind ridges can either

damage the UAV directly or exceed its maximum horizontal or climb speed and thereby cause collision with terrain. Especially slow, fragile and low propulsion-power-to-weight vehicles such as solar UAVs (Figure 1) are susceptible to such weather effects.



Figure 1: Cluttered terrain and strong winds, as e.g. observed in the pictured operation of *AtlantikSolar* [2], [3] above the Arctic, pose a significant risk to a UAV. This paper therefore presents a downscaling method to predict the local 3D wind field in real time such that path planners can calculate optimal trajectories throughout complex terrain and wind.

Safe operations in such complex environments require three technologies that do not exist together on today’s small-scale UAVs: First, a Simultaneous Localization And Mapping (SLAM) system to create a map of the environment, second, a method to predict the local 3D wind field around the UAV, and third, a path planner to calculate an optimal collision-free path through the terrain and 3D wind field. The focus of this paper is on wind field prediction. Such a system needs to run in real time to react to new terrain information. In addition, to guarantee flight safety at all times, including conditions with a degraded communication link, the system needs to run onboard and be independent from local infrastructure.

Contributions

This paper presents the literature’s first *real-time*² 3D wind field prediction method that can run directly onboard the limited computational resources of today’s small-scale UAV on-

²Given the quasi-static nature of most weather phenomena, the definition of *real time* used in this paper is more relaxed than the one used e.g. for flight control. It is as follows: Assume the UAV is constantly mapping new terrain at distance d in front of it. The wind prediction then needs to be fast enough to allow the UAV to react to potential dangers (winds) at distance d . For example, if $d = 300 \text{ m}$ and the ground speed is 10 m/s , then calculation times $t \ll 30 \text{ s}$ are considered real time. Real time capability is easier to reach for high altitude flight because d is also larger.

board computers. The selected downscaling method receives low-resolution NWP data as an input via low-bandwidth data links (e.g. satellite communication) and then leverages potential flow theory to generate a high-resolution 3D wind field close to the UAV. The method is assessed in simulations and with LIDAR wind profiles from Swiss alpine regions.

Given that this paper is the first in the literature to implement *real-time* 3D wind field predictions *from onboard a UAV*, we consider the paper initial work on the path towards full onboard weather-awareness of UAVs. It is not expected that the simple downscaling method captures all the different flow patterns over complex terrain. From a scientific viewpoint, the paper instead seeks to assess how accurately even this simple method can predict the wind field when running onboard a UAV in real time and whether this is already enough to allow significant trajectory planning improvements.

Organization

The remainder of this paper is organized as follows. Section 2 presents the current state of the art. Section 3 defines the method requirements and presents the design of the selected method. Section 4 presents both synthetic and experimental results. Section 5 concludes the paper.

2. STATE OF THE ART

UAV-based *predictions* of wind fields have not been discussed in the current literature yet. Instead, only *in-situ measurement* methods are described: Filtering approaches that fuse the information of an Inertial Measurement Unit (IMU), Global Positioning System (GPS) and airspeed sensor are mostly used [4], [5]. Approaches that estimate the wind vector without an airspeed sensor also exist [6]. In addition, formations of UAVs have been used to cooperatively measure wind fields [7]. To find suitable methods for the *online prediction* of 3D wind fields from UAVs, *downscaling* methods that historically come from the fields of meteorology and flow simulation are investigated. In general, these downscaling methods are of three different types:

- *Statistical methods* use previously measured wind data in a certain region and thereof derive wind expectations for the same region using statistical methods. Given that we aim to predict the wind over unvisited terrain, these methods are not suitable for our application.
- *Prognostic models* calculate the approximate solution of differential equations describing the Atmospheric Boundary Layer (ABL). They can reproduce small-scale weather effects, but are computationally expensive. ARPS, the Advanced Regional Prediction System [8], [9], has been applied to complex terrain [10], [11], [12], [13] and is freely available³. GRAMM, the Graz Mesoscale Model [14], considers mass, momentum, potential temperature and humidity conservation and is often used together with the dispersion model GRAL (Graz Lagrangian Model), e.g. in [15]. It is free⁴ and provides a GUI.
- *Diagnostic methods* do not have a proper physical background but return a physically consistent wind field with respect to mass or momentum conservation. They are less computationally demanding than prognostic models but cannot predict local effects such as slope winds [16]

or the time-dependency of the flow. MATHEW [17], [18] is a diagnostic mass consistent model which adds a potential field to an initial wind field to make it divergence free. The source code of a modified model is available [19]. CALMET [20] additionally models kinematic terrain effects, slope flows and blocking effects and can incorporate observational data at the cost of increased computational cost. The usage of CALMET coupled to NWP data showed a significant improvement with respect to NWP outputs [21], [22]. Software licences are available for free⁵. WindNinja [23] provides two different models: A mass-conserving model which resembles MATHEW, and a mass- and momentum-conserving model that uses commercial CFD software. Near surface winds are improved especially for high-wind events [24]. The software with GUI is available for free⁶.

3. METHOD DESIGN

Requirements

The following specific requirements are derived for the UAV-based wind prediction framework: It shall

- be usable on UAVs operating remotely or in cluttered terrain where the communication to the ground station is limited or even completely lost. An onboard-implementation is therefore required.
- provide results in near real-time, i.e. shall calculate a sufficiently accurate 3D wind field for an area of 1 km³ (with the UAV at its center) and 25 m grid resolution in less than 30 s calculation time on the UAV's onboard computer⁷. We assume this corresponds to 10 s on a standard laptop computer.
- have a robust and fully automated execution that does not require any human intervention.
- be modular and easily modifiable.

Selection

Five different models were traded off against the requirements presented above. First, both ARPS and GRAMM/GRAL were discarded because they violate the computation time constraint, i.e. usually require far more time to guarantee that a numerically stable solution is reached. The mass- and momentum-conserving version of WindNinja is discarded due to computation time disadvantages, too. Of the remaining models, CALMET is considered slower and has a less modular code base. The two remaining models are MATHEW and the mass-conserving WindNinja, which essentially solve the same mathematical problem. Finally, MATHEW is selected because it has a slim and modular source code and can, as shown by Walt [19], solve similar meteorological downscaling problems within the calculation time requirements. In addition, it solves the internal finite element problem using FEniCS, an open-source library that already provides many different solvers and preconditioners, has both Python and C++ bindings, and supports various grid types.

⁵http://www.src.com/calpuff/calpuff_eula.htm

⁶<https://www.firelab.org/document/windninja-software>

⁷Such as an INTEL UP Board, <http://www.up-board.org/up/>

³<http://www.caps.ou.edu/ARPS/arpsdown.html>

⁴<http://lampx.tugraz.at/~gral/index.php/download>

Fundamentals

MATHEW calculates the divergence-free wind field \vec{u} from an initial wind field \vec{u}^l (retrieved from a NWP system), a Digital Elevation Model (DEM) and the stability matrix \mathbf{S} . The problem domain Ω is an open set in 3-dimensional space and its boundaries are the Dirichlet boundary Γ_D , which covers the open-flow boundary, and the Neumann boundary Γ_N defining the terrain (Figure 2). The central *hard* constraint enforced by MATHEW is that the resulting \vec{u} needs to be divergence free over the domain Ω , i.e.

$$\nabla \cdot \vec{u} = 0. \quad (1)$$

This essentially means that the flow is incompressible. The main *soft* constraint is that the adjustments between the initial wind field \vec{u}^l and the final wind field \vec{u} shall be minimal. This expresses the assumption that the initial wind field observation \vec{u}^l contains valuable information collected through sensing or a prior model — and is thus *to some degree* correct (see Section 4 for a violation of that assumption). Mathematically, MATHEW reduces the accumulated error between \vec{u} and \vec{u}^l over the domain in a least-squares sense by minimizing the functional

$$L(\vec{u}; \lambda) = \int_{\Omega} \left(\frac{1}{2} \mathbf{S} (\vec{u} - \vec{u}^l) \cdot (\vec{u} - \vec{u}^l) + \lambda \nabla \cdot \vec{u} \right) dV, \quad (2)$$

with velocity potential λ and the atmospheric stability matrix

$$\mathbf{S} = \begin{pmatrix} \alpha_h^2 & 0 & 0 \\ 0 & \alpha_h^2 & 0 \\ 0 & 0 & \alpha_v^2 \end{pmatrix}. \quad (3)$$

Here, α_h and α_v are weights for the horizontal and vertical component. In contrast to the work by Sherman [17], they are herein defined as $\alpha_i^2 = 1/\sigma_i^2$, where σ_i are the observation errors or deviations of the observed and desired wind field. The weights α_h and α_v are used to define the stability parameter α as

$$\alpha = \frac{\alpha_h^2}{\alpha_v^2} = \left(\frac{w}{u} \right)^2, \quad (4)$$

where u and w are the magnitudes of the expected horizontal and vertical wind speeds, respectively. Inserting α into the

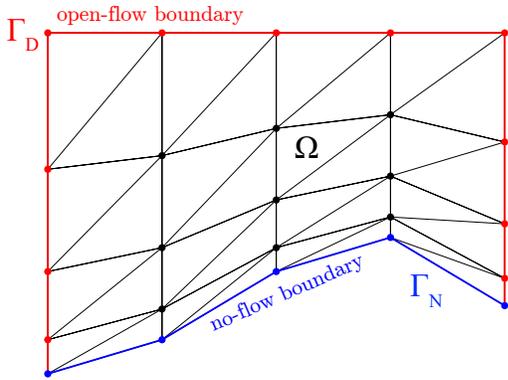


Figure 2: Computational domain and boundaries of MATHEW on a 2D grid: The red lines and dots represent the open-flow boundary whereas blue is used for the terrain. The black dots are the grid points within the domain Ω and the black lines illustrate the cells.

definition of the stability matrix, its inverse becomes

$$\mathbf{S}^{-1} = c \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{pmatrix}. \quad (5)$$

Note that the constant factor c is neglected ($c = 1$) in [19], [25] because it, first, does not have an effect on the final wind field \vec{u} as during the solution of Eq. (11) λ would automatically be scaled by $1/c$, and second, it cannot be used to adjust the atmospheric stability. This is however possible using the stability parameter α . Sherman [17] assesses multiple test cases and determines a typical value of $\alpha \approx 10^{-4}$. A larger α implies that the wind field is mainly corrected in vertical direction, a smaller one that the correction is principally applied to the horizontal direction.

To minimize Eq. (2), we apply the Euler-Lagrange equations for vector functions

$$\frac{\partial L}{\partial f_i} - \frac{d}{dx} \left(\frac{\partial L}{\partial f'_i} \right) = 0. \quad (6)$$

The problem definition now has the form

$$\vec{u} = \vec{u}^l + \mathbf{S}^{-1} \nabla \lambda. \quad (7)$$

Note that while the added potential flow field $\mathbf{S}^{-1} \nabla \lambda$ in Eq. (7) is vorticity-free, the final wind field \vec{u} can still contain vorticity (for example recirculation behind a ridge) if it is already present in \vec{u}^l . Two boundary conditions are now applied: The no-flow through terrain assumption, which translates to $\vec{u} \cdot \vec{n} = 0$ where \vec{n} is the normal vector at the terrain, and the assumption of no adjustment of the tangential velocities at the open-flow boundaries, which translates to $\lambda = 0$. The resulting system of equations is

$$\begin{aligned} \vec{u} &= \vec{u}^l + \mathbf{S}^{-1} \nabla \lambda && \text{in } \Omega \\ \nabla \cdot \vec{u} &= 0 && \text{in } \Omega \\ \lambda &= 0 && \text{on } \Gamma_D \\ \vec{u} \cdot \vec{n} &= 0 && \text{on } \Gamma_N. \end{aligned} \quad (8)$$

These equations can be expressed as a Partial Differential Equation (PDE) for the velocity potential λ by taking the scalar product with the gradient operator on both sides of Eq. (8):

$$\begin{aligned} -\nabla \cdot (\mathbf{S}^{-1} \nabla \lambda) &= \nabla \cdot \vec{u}^l && \text{in } \Omega \\ \lambda &= 0 && \text{on } \Gamma_D \\ (-\mathbf{S}^{-1} \nabla \lambda) \cdot \vec{n} &= \vec{u}^l \cdot \vec{n} && \text{on } \Gamma_N. \end{aligned} \quad (9)$$

Equation (9) is an elliptic PDE. For the transformation into a Finite Element Method (FEM) problem, the first line in Eq. (9) needs to be post-multiplied with a test function F satisfying its boundary condition ($F = 0$ on Γ_D) and integrated over the problem domain Ω :

$$-\int_{\Omega} \nabla \cdot (\mathbf{S}^{-1} \nabla \lambda) F dV = \int_{\Omega} (\nabla \cdot \vec{u}^l) F dV. \quad (10)$$

Using Green's Theorem on Eq. (10) and rearranging the terms yields the so called weak formulation of the problem

$$\underbrace{\int_{\Omega} (\mathbf{S}^{-1} \nabla \lambda) \cdot \nabla F dV}_{a(\lambda, F)} = \underbrace{\int_{\Omega} (\nabla \cdot \vec{u}^l) F dV - \int_{\Gamma_N} (\vec{u}^l \cdot \vec{n}) F dS}_{l(F)} \quad (11)$$

Here, $a(\lambda, F)$ is the bilinear form and $l(F)$ is the linear form. The procedure is to now solve for the velocity potential λ using the Ritz-Galerkin discretization (for details see Walt [19]), and to finally re-insert λ into Eq. (7) to retrieve the final 3D wind field \vec{u} .

Implementation

Figure 3 presents the data sources and steps required by our implementation of MATHEW. The framework is implemented in Python with Robot Operating System (ROS) [26] support. The input data are the simulation parameters, a DEM and coarse NWP data (wind profiles around and within the region of interest). The input data is monitored at 0.5 Hz and the model is initialized when the inputs have changed. The grid is generated and the initial wind field \vec{u}^I is calculated by linearly interpolating the coarse NWP data to the vertical layer heights of the high-resolution grid. Afterwards bilinear interpolation is used within each vertical layer to calculate the wind components at each grid point. After setting up the weak problem formulation and solving for the velocity potential λ , the final wind field \vec{u} is then calculated using Eq. (7) and forwarded to a flight planner as described in [27].

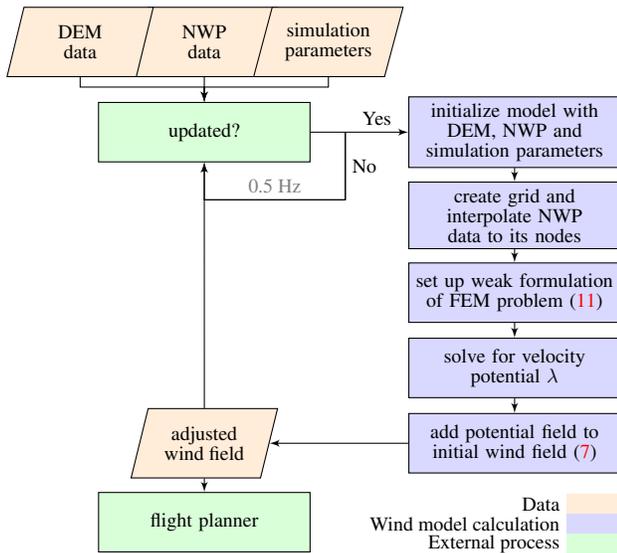


Figure 3: Wind downscaling architecture: Input data changes trigger the wind field downscaling calculation, which then publishes its outputs via ROS to the flight planner.

The decision to implement the architecture of Figure 3 onboard a UAV instead of a ground-based computer is motivated by the requirement to have wind field forecasts available at all times, i.e. even when flying remotely with reduced communication bandwidth to the ground station. The transmitted data volume therefore needs to be minimal. However, even the 1 km^3 high-resolution wind field defined in Section 3 contains 1681 horizontal grid points with multiple vertical layers. Calculating this wind field on a ground-based computer and transmitting the results via low-speed telemetry links (for example degraded line-of-sight telemetry or satellite communication) is not possible in real-time. Therefore, an onboard implementation is chosen. The NWP data is provided by the European Consortium for Small-Scale Modeling (COSMO) model [28] at 1–7 km horizontal resolution and up to 80 vertical layers. A ground-based client computer only extracts the NWP data at the specific time and region of interest. Assuming COSMO-1 NWP data, the 1 km^3 region

of interest then consists of only 9 horizontal grid points with different vertical levels that need to be transmitted. Even low-bandwidth links such as the satellite communication link on *AtlantikSolar* [3] can manage such a bandwidth. The DEM is either preloaded or generated directly onboard the UAV by a vision-based SLAM node [29].

Optimization

The framework’s accuracy and computation time depend on three critical parameters: The vertical grid point spacing, and the preconditioner and solver used by FEniCS. These are optimized by comparing the model to the analytical results for flow around a hemisphere with radius 0.25 m (Figure 4). It is placed at the origin of the domain $[-1, -1, 0] \times [1, 1, 1]$ (in m) with 1 m/s inflow in positive x -direction. $N_x = N_y = 41$ horizontal grid points are used. The stability parameter is $\alpha = 1$. While the horizontal spacing is constant, the vertical spacing is not: The vertical position z of a grid point at vertical index n and position x, y is defined as

$$z(x, y, n) = h(x, y) + (t(x, y) - h(x, y)) \cdot \gamma(n), \quad (12)$$

where $h(x, y)$ is the terrain height and $t(x, y)$ is the top vertical domain boundary at the x, y position. The discrete function $\gamma(n)$ needs to be strictly monotonically increasing and fulfill $\gamma(0) = 0$ and $\gamma(N_z) = 1$, where N_z is the number of vertical grid points. Equidistant, square-root, linear and squared spacing functions $\gamma(n)$ are tested. The error at a location is defined as the length of the difference vector between the model output and the analytic solution. The weighed total error sums up the product of each local error with the ratio between the height of the respective cell and the average cell height of the entire domain. The median weighted error for all vertical spacings is nearly the same, i.e. about 0.005 m/s . The linear vertical spacing is selected because its maximum weighted error is only 0.14 m/s and its weighted errors show a low standard deviation.

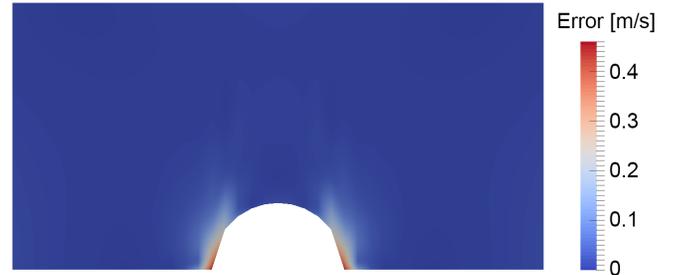


Figure 4: Non-weighted local errors, i.e. the magnitude of the difference between the wind field vectors predicted by the model and the analytical solution, for the hemisphere. The linear vertical spacing, ILU preconditioner and CG solver are used. The largest errors occur in a small region with abrupt terrain change at the bottom of the hemisphere.

The elliptic PDE of Eq. (9) was previously solved using a direct solver [19]. To fulfill the stringent computation time requirements, different FEniCS preconditioners and iterative solvers were assessed using the aforementioned hemisphere test case. The best combination is found to be the FEniCS Incomplete LU-decomposition (ILU) preconditioner with the Conjugate Gradient (CG) solver. Averaged over 10 runs, this combination reduces the computation time for λ from 81.8 s for the direct solver to only 6.3 s. The $< 10 \text{ s}$ calculation time requirement is thus fulfilled. The additional errors introduced into the wind field have a norm of less than 10^{-4} m/s and

are deemed acceptable given that a time reduction of 92% is achieved. Figure 4 shows the overall non-weighted errors of the framework. More details on the implementation and parameter optimization are presented by Müller [25].

4. RESULTS

Synthetic Test Cases

Steep and narrow valley—The steep and narrow valley (Figure 5) tests how mass conservation is achieved for different stability parameters. At the center of the valley ($x = 0$, $y = 0$) the width is one fourth of the inflow width. The initial wind field $\vec{u}^I = [5, 0, 0] \text{ m/s}$ is constant over the entire domain. While \vec{u}^I is divergence free over Ω , it violates the terrain boundary conditions. The model output \vec{u} is shown in Figures 5 and 6. The ratio of the average inflow speed to the average valley wind speed is roughly 1/4, thus mass conservation is fulfilled. The average wind speed in the domain is slightly reduced (Table 1). For $\alpha = 1.00$, the flow is also able to avoid the orifice represented by the valley by rising in front of the valley and sinking again afterwards. Given the open flow boundary above the valley this behavior is expected. Overall, the test case shows that \vec{u} is made divergence free, terrain following and mass consistent. However, it also shows a major difference to common CFD software: Equation (2) minimizes the least squares error between \vec{u} and \vec{u}^I over the whole domain Ω , and this error is smaller if \vec{u} is decreased in the inflow than if it is kept constant at the inflow but significantly increased inside the valley. This process does not respect inflow boundary conditions and significantly changes the mean valley inflow in Table 1. Therefore, even if the inflow boundary conditions are known with much higher accuracy than the conditions inside the domain (e.g. from highly accurate NWP data) they are not respected.

Table 1: Wind speeds for the valley test, the initial field \vec{u}^I and the adjusted field \vec{u} at two stability values α .

Wind field	\vec{u}^I	\vec{u}	\vec{u}
Stability parameter α		0.01	1.00
Mean valley inflow [m/s]	5.0	2.32	2.75
Maximal wind speed [m/s]	5.0	11.86	11.55
Average domain wind [m/s]	5.0	4.48	4.62

Ramp—The synthetic ramp test is also initialized with $\vec{u}^I = [5, 0, 0] \text{ m/s}$ in the entire domain. The result for a 600 m domain height is shown in Figure 7. The geometry again represents a flow orifice, and the overall wind magnitudes are therefore decreased at the inflow and increased at the outflow. The ratio between inflow and outflow velocities is however lower than in the valley because the geometry narrows less and the top domain boundary is an open flow boundary where some mass is expelled. The vertical flow field and the regions of maximum wind speed close to the top of the ramp — where soaring flight would be optimal — are represented well. The comparison with smaller domain heights shows a decrease in quality and emphasizes that MATHEW requires a sufficiently large domain height. For the remaining analysis the domain factor, i.e. the ratio of domain height to the height of the tallest obstacle, is therefore set to 3.5. Overall, the results agree very well with the intuitive solution.

Experimental Validation

Measurement and simulation setup—To compare the model output to actual wind data, an extensive literature review for

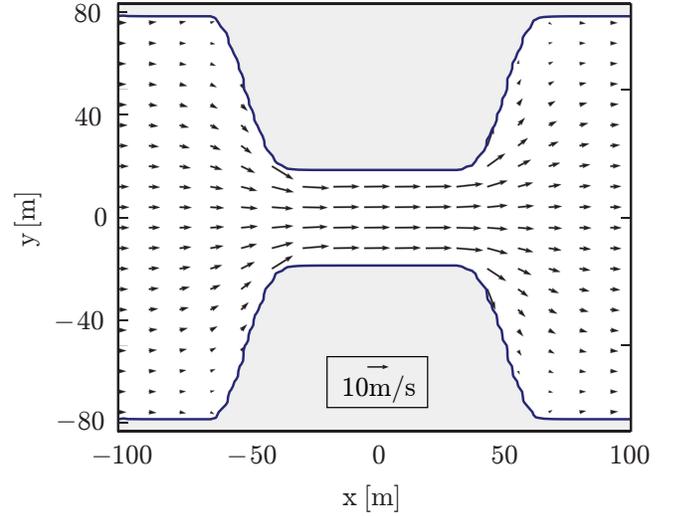


Figure 5: Synthetic valley. The terrain height is 0 m in the white area and 100 m in the grey areas. The adjusted flow (black arrows) is shown at $z = 40$ m and $\alpha = 1.00$. As expected, the wind speed in the valley is roughly four times as high as at the in and outflow.

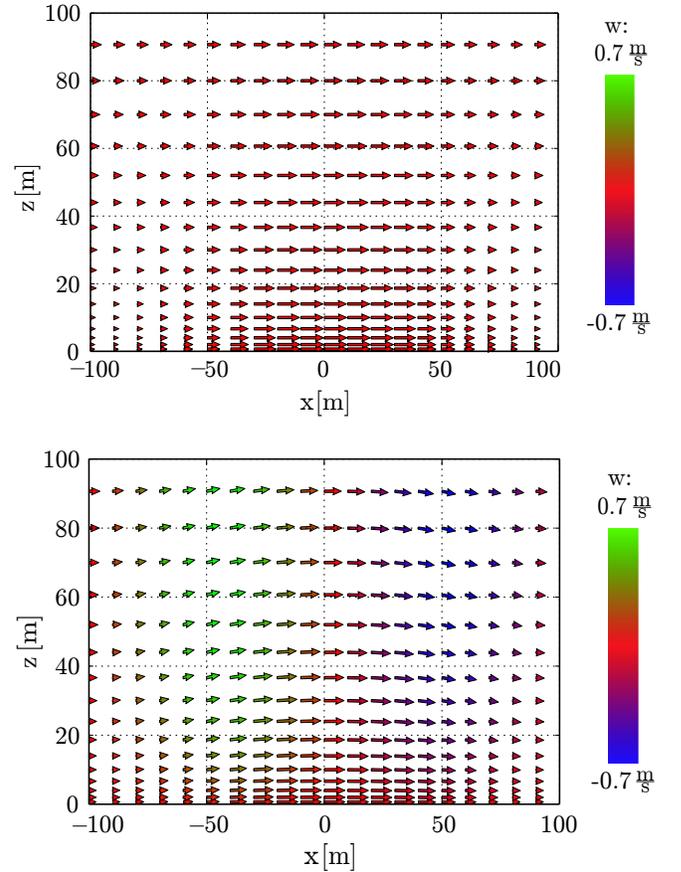


Figure 6: Adjusted wind fields \vec{u} for $\alpha = 0.01$ (top) and $\alpha = 1.00$ (bottom) for the synthetic valley test case in the x - z cross section at $y = 0$. The arrow color represents the vertical component w . As expected, the wind speed is higher inside the valley. The overall wind magnitude is decreased in both cases. For $\alpha = 1.00$, the flow is also adjusted vertically.

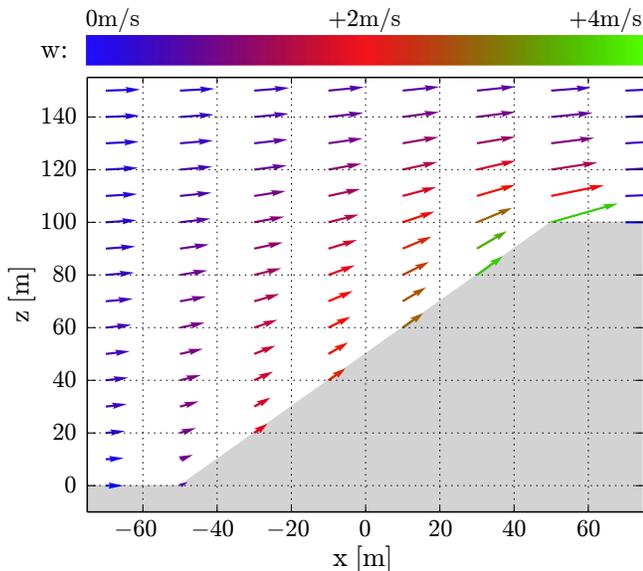


Figure 7: Synthetic ramp results: Both the observed flow acceleration and the vertical components w correspond well to an intuitive solution. The arrows scale with the magnitude and the color visualizes w . We retrieve 3.79 m/s average inflow, 5.79 m/s average outflow, 1.23 m/s minimal wind, and 9.69 m/s maximal wind.

measured multi-dimensional wind fields in complex terrain and at typical flying altitudes of a UAV was performed. Unfortunately, even after multiple weeks of searching no such measurements could be found. Therefore, 1-D LIDAR measurements of wind vectors at multiple altitude levels (Figure 8) are used. The LIDAR data was recorded by the Zurich University of Applied Sciences at three different locations and time periods (Table 2). It was filtered and averaged to exclude invalid raw data, missing measurements, or excessive changes within consecutive time steps. More details about the exact instrumentation and measurements can be found in [30] and [31] for San Bernardino/Bivio and Vorab, respectively.

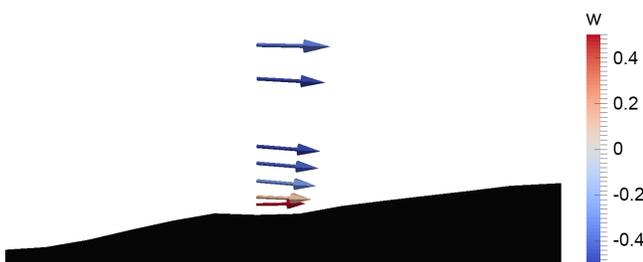


Figure 8: Exemplary LIDAR profile from Vorab with wind vectors at different heights. While one would expect $w > 0$ due to the left-to-right winds and the increasing terrain height in that direction, the LIDAR yields $w < 0$ for most altitudes. Such *unintuitive* LIDAR measurements occur regularly. They can be caused by terrain or effects that lie outside the computational domain (e.g. large mountain waves), which the downscaling model can of course not take into account.

Figure 9 visualizes the terrain, the computational domain and the LIDAR position, which is always the center of the computational domain with side length 2.2 km. A grid with 20 vertical layers and 50 m horizontal resolution is used. Note that for San Bernardino and Bivio only COSMO-2 weather

Table 2: LIDAR measurement sites in Switzerland, time periods and amount of valid data (where d =days and p = LIDAR profiles). The valid data days are less than the time period because data had to be filtered out e.g. due to weather.

	San Bern.	Bivio	Vorab
Lat. °N	46.46357	46.46251	46.87398
Lon. °E	9.18465	9.66864	9.18186
Year	2015	2015	2016
Period	Jun.2–Jun.17	Jun.17–Jul.21	Mar.22–Apr.4
Valid data	2d/30p	2d/36p	6d/70p

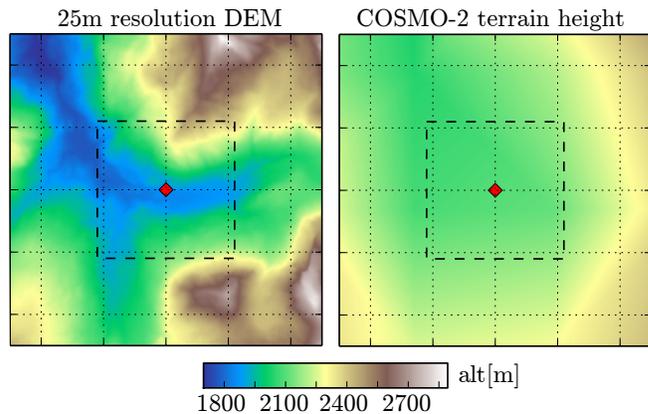


Figure 9: Terrain (5x5 km), downscaling domain (2.2x2.2 km, dashed line) and LIDAR location (red diamond) for the Bivio measurement site. North is up. The COSMO-2 terrain is not able to resolve the cluttered terrain. The initial wind field \vec{u}^l provided by COSMO-2 will thus not be a good approximation of \vec{u} . DEM data: © 2016 swisstopo (JD100042), free for educational purposes.

data is available, while the higher-resolution COSMO-1 data can be used for Vorab.

Measured and estimated wind fields—To show how the downscaling model can improve the wind field estimate, but to also show its limitations with respect to modelling small-scale thermally induced wind phenomena, this section exemplarily compares the model output and measured winds for Bivio. As visible in Figure 9 the LIDAR is located in a steep east-west valley but the corresponding COSMO-2 terrain is basically flat. As a result, on July 8th 9:00 UTC the interpolated COSMO-2 wind field indicates flow coming from south-east (Figure 10) and therefore violates the assumption of no flow through the terrain boundaries. The downscaling model is able to rotate \vec{u} in the correct direction such that the winds satisfy the terrain boundary conditions. The measured wind direction agrees well with the downscaled wind field. However, the wind speed is significantly underestimated both by the initial wind field and the adjusted wind field. The overall observed situation is a so called mountain breeze, i.e. a cold air mass that moves from the mountain into the valley during the night and early morning hours.

One hour later, at 10:00 UTC, the wind direction has changed and wind is flowing from the valley to the mountains. This *valley breeze* situation is shown in Figure 10. The interpolated wind direction again does not respect the terrain boundaries and also differs significantly from the measured wind direction. The downscaling model is again able to predict the correct wind direction. The wind magnitude is overestimated

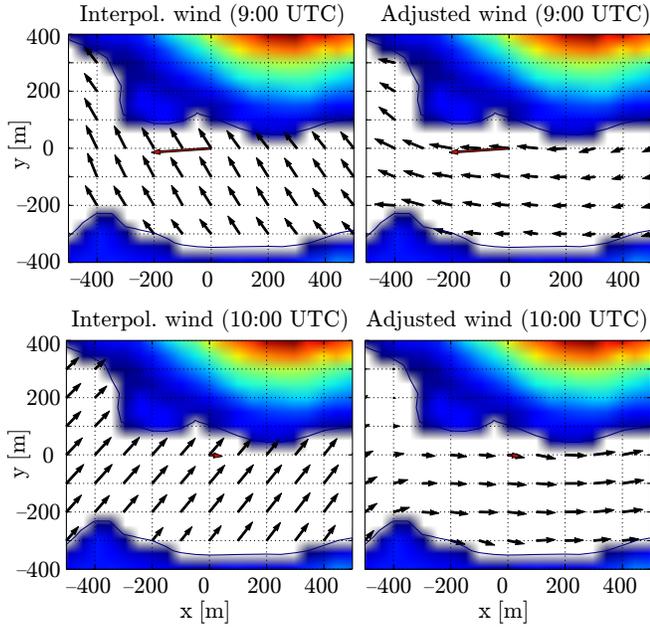


Figure 10: Initial and adjusted wind fields \vec{u} (black arrows) at Bivio for 09:00UTC (top) and 10:00UTC (bottom) on July 8th 2015. The wind is shown at 1900 m AMSL. The red arrow is the wind measured by the LIDAR.

by both approaches, but the error for the downscaling model is much smaller.

To analyze the reason for the underestimated wind speed at 09:00 UTC, Figure 11 plots the measured, interpolated and modeled wind profiles. The measured horizontal wind profile shows a distinct maximum 40 m above ground. This small-scale phenomenon is called *low-level jet* and is not represented in the interpolated wind profile. Given that the downscaling model has no deeper physical understanding of thermally-induced effects, it also cannot predict this behavior. At 10:00 UTC, the interpolated wind is again too high but the downscaling model can partially correct for this. In both cases, the downscaling model clearly improves the wind direction such that the overall agreement between the modeled and measured wind vectors is improved.

Statistical evaluation— To quantitatively assess the wind field improvement, a comparison of the downscaling model against LIDAR measurements is performed over all test cases. Given that we only want to investigate modeling errors but not errors due to a badly selected stability parameter, we always first run the model for $\alpha = [10^{-6}, 10^{-5}, \dots, 10^1]$ and select the run with the *optimal* α , i.e. the one which yields the minimum Root Mean Square Error (RMSE) between the modeled and measured wind profile. This approach of course has a certain risk of overfitting such that the following results have to be considered best-case results. In the future, the wind field downscaling model will automatically select the best α by assessing the current atmospheric stratification using the temperature profiles in the NWP data.

Given that only 1D measurement data exists, all wind fields can only be compared locally at the LIDAR’s position. In addition to the downscaled wind field \vec{u} , we also compare the initial wind field \vec{u}^1 and a zero-wind field $\vec{u}^0 = 0$. The LIDAR measurements serve as ground-truth. The comparison metric

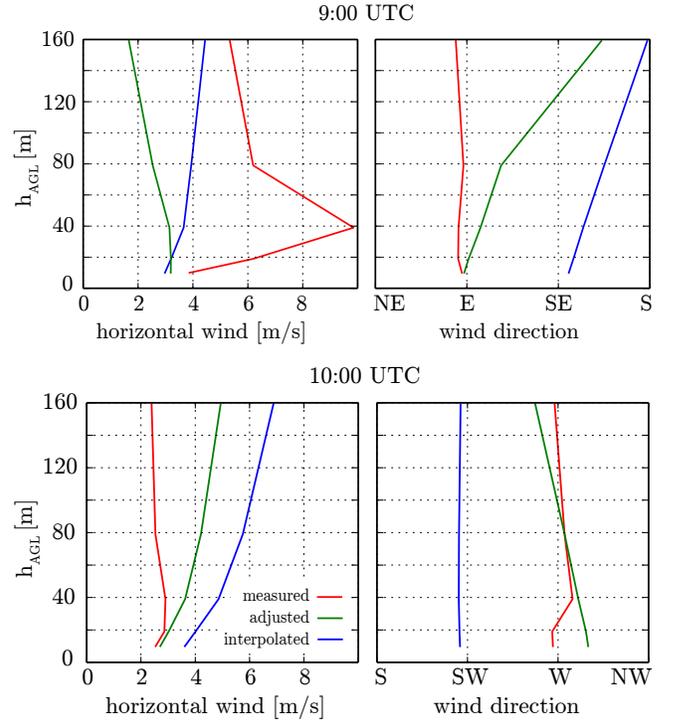


Figure 11: Wind profiles at Bivio on July 8th 2015 at 9:00 UTC (top) and 10:00 UTC (bottom). In the top plot, the LIDAR measures a so-called *low-level jet*. The downscaling model significantly improves the wind direction and tends to improve the wind speed.

is the root mean square error. It is calculated for two newly introduced variables: First, the weighted horizontal vector error⁸ is

$$e_{\text{hor}} = \frac{\sqrt{(u_p - u_m)^2 + (v_p - v_m)^2}}{v_{\text{air}}} \quad (13)$$

Here, the subscripts p and m stand for the predicted and measured components, and v_{air} is the nominal airspeed of 9 m/s for *AtlantikSolar* [3]. The variable e_{hor} therefore normalizes the horizontal wind estimate error with the aircraft speed. If $e_{\text{hor}} > 1$, the wind is too strong to overcome the wind and collision with terrain can occur. The second variable is the weighted vertical error

$$e_{\text{ver}} = \begin{cases} \frac{w_p - w_m}{w_{\text{sr}}} & \text{if } w_p - w_m \leq 0 \\ \frac{w_p - w_m}{w_{\text{cr}}} & \text{if } w_p - w_m > 0, \end{cases} \quad (14)$$

where w_{sr} is the maximum sink rate (positive) and w_{cr} the maximum climb rate. For *AtlantikSolar* these are 3 m/s and 1.5 m/s respectively. This error is especially important for fragile or low-power aircraft: For example, if the predicted vertical wind $w_p = 0$ m/s and $e_{\text{ver}} < -1$, then the aircraft sink rate is exceeded when altitude hold is required and the aircraft may sustain structural damage. Equivalently, when $w_p = 0$ m/s but $e_{\text{ver}} > 1$, then the actual downwards wind magnitude is higher than the aircraft climb rate and the

⁸It would make more sense to compare the horizontal wind speed and direction instead of combining them in one variable. But given that the zero wind field \vec{u}_0 has no proper wind direction, the above introduced variable seems to be the best solution.

aircraft is in risk of colliding with terrain. Overall, if any of the error magnitudes exceed unity, the aircraft is in danger.

Table 3 shows the final results. The root mean square horizontal errors, vertical errors, and — given that they were already properly normalized with the aircraft nominal speed, climb rate and sink rate — their sum are presented. In Bivio, the vertical error is reduced by neither the initial nor the downscaled wind field but the downscaling model reduces the horizontal error by 38% with respect to the zero wind assumption. In San Bernardino, the vertical error is slightly reduced by both \bar{u}^I and \bar{u} . The horizontal error is reduced by 27% by \bar{u}^I and by 31% by \bar{u} . In Vorab, the initial wind field yields a much larger vertical error than the zero-wind assumption and the downscaling model can only marginally compensate for that. The horizontal error is however reduced by 46% by \bar{u}^I and by 48% by \bar{u} . Averaged over all three test cases (with equal weight for each), it becomes clear that both COSMO-2 and the higher-resolution COSMO-1 model provide initial wind fields which have similar or higher vertical errors than the zero-wind assumption. The low-quality terrain representation (Figure 9) is certainly contributing to this issue. The downscaling model then does not have the leverage to significantly improve this: On average, \bar{u}^I increases the vertical error by 30%, and the error increase after downscaling is still 26%. However, the horizontal error is reduced by 31% by the initial wind field and by 41% for the downscaled wind field. The sum of both errors reduces by 15% with the initial wind field and by 23% with the presented downscaling method.

The error histograms are displayed in Figure 12. In a first step, only the horizontal error distribution is analysed: The no wind assumption shows a wide distribution with two peaks at approximately 4 m/s and 12 m/s. The interpolation of NWP data already performs better and the respective error distribution has only one peak at slightly below 4 m/s. The output for the model with optimal stability parameter performs even better, the respective peak is around 3 m/s. Looking at the vertical error distributions, the zero wind assumption leads to the smallest RMSE. The interpolated wind field shows a much higher vertical RMSE and thus more spread in the error distribution. The adjusted wind field shows a very similar distribution. As discussed before, this mainly comes from the measurements in Vorab ($e_{\text{ver}} = 0.71/0.27/0.14$ for Vorab/San Bernardino/Bivio respectively), which show vertical winds that cannot be described by simple phenomena (see for example Figure 8) and are therefore not covered in COSMO data. It should be noted that modeling local anomalies such as mountain waves or thermal flows is also very challenging for more complex (e.g. *prognostic*) downscaling models. In addition it is also possible that the LIDAR data is error-prone because of unfiltered clouds, fog or precipitation. All in all, the adjusted wind field however has a 23% and 8% smaller RMSE than the zero-wind and interpolated wind fields respectively.

5. CONCLUSION

This paper has presented the literature’s first 3D wind field prediction system designed to run in real-time onboard a UAV. The method is based on downscaling low-resolution wind data from a global weather model via potential flow theory, which guarantees that the terrain boundaries in a high-resolution 2.5D height map, mass conservation and the atmospheric stratification are observed. Due to the model’s simplicity, typical flow fields (1 km³ at 25 m resolution) are

Table 3: Horizontal and vertical wind estimate root mean square errors for the zero-wind assumption \bar{u}^0 , the NWP-based interpolation \bar{u}^I and the adjusted wind field \bar{u} with optimal stability parameter. The LIDAR measurement serves as wind ground truth. The percentage values are the relative changes with respect to the zero-wind assumption.

Type	RMSE(e_{ver})	RMSE(e_{hor})	Sum
<i>Bivio</i>			
Zero wind	0.13	0.54	0.67
Interp. wind	0.16 (+22%)	0.52 (-4%)	0.68 (+1%)
Adjusted wind	0.14 (+9%)	0.34 (-38%)	0.48 (-28%)
<i>San Bernardino</i>			
Zero wind	0.30	0.71	1.01
Interp. wind	0.27 (-10%)	0.52 (-27%)	0.79 (-22%)
Adjusted wind	0.27 (-9%)	0.50 (-31%)	0.77 (-24%)
<i>Vorab</i>			
Zero wind	0.47	1.20	1.67
Interp. wind	0.74 (+58%)	0.64 (-46%)	1.38 (-17%)
Adjusted wind	0.71 (+53%)	0.63 (-48%)	1.34 (-20%)
<i>All test cases combined</i>			
Zero wind	0.30	0.82	1.12
Interp. wind	0.39 (+30%)	0.56 (-31%)	0.95 (-15%)
Adjusted wind	0.38 (+26%)	0.49 (-41%)	0.86 (-23%)

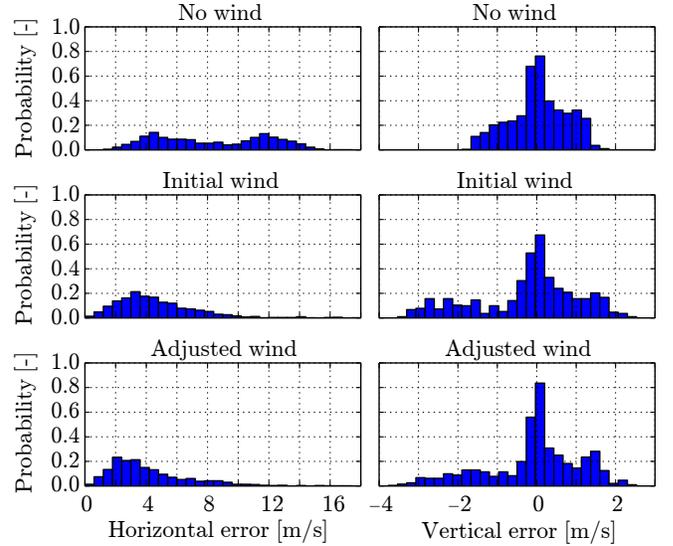


Figure 12: Error histograms: The horizontal error distribution is clearly improved by the downscaling model, but the vertical component is worse than the no wind assumption.

calculated in below 10 s on a standard laptop computer. The model yields good qualitative results in synthetic valley and ramp test cases. The comparison to 1D LIDAR profiles collected in the Swiss Alps shows an average horizontal wind error decrease of 41% over the *zero-wind assumption* that is mostly used in UAV path planning today. However, the vertical wind error is increased by 26% in contrast to the zero-wind assumption. All in all, as shown in Table 3, the method decreases the overall error with respect to a zero-wind assumption by 23%. It is thus more accurate than not considering any wind during the planning stage at all, and is also slightly better than the pure interpolation of a global weather model. As a result, path planners can generate safer aircraft trajectories e.g. by avoiding areas of strong horizontal

winds. However, due to the simplicity of the model the accuracy is not yet sufficient for the use on fragile or low-power UAVs. Follow-up research is thus required.

Improving the Present Method

The remaining uncertainty in this initial evaluation concerns the vertical wind error: It is mostly caused by the *Vorab* LIDAR measurements, which contain local anomalies such as mountain waves or thermal flows that are also hard to predict for more sophisticated models and even human intuition. The LIDAR data's accuracy might occasionally be compromised by clouds, fog or precipitation that were not filtered sufficiently. To allow a final evaluation of the method's accuracy, more extensive 3D wind field data — collected in a wind tunnel or via UAVs with accurate airspeed probes — thus needs to be collected. However, to reach the vertical wind accuracy required for fragile low-power UAVs, the method itself and its settings also require extensions. Future research is therefore suggested in the following areas:

- An increase of the computational domain. This models larger areas of terrain and allows to rudimentarily capture the effect of mountain waves which, as in the *Vorab* test case, may cause significant errors.
- Online fusion of the wind measurements taken by the UAV along its previous trajectory. This can significantly increase the initial wind field quality. For example, such a method could easily detect and resolve both the wind direction and wind speed errors in \vec{u}^1 in Figure 10.

Outlook: Leveraging Novel Methods

If, after implementing those extensions, the current method still does not prove accurate enough, then methods which do not only *adjust* the initial wind field \vec{u}^1 , but which completely *recalculate* it will be investigated. A novel approach to manage the increased computational complexity is to combine CFD simulation techniques with machine learning approaches. For example, the computer vision related literature has leveraged Convolutional Neural Networks (CNN's) to speed up fluid simulations by either completely replacing the fluid solver [32] or by replacing significant parts such as the pressure correction [33]. Other approaches use a hybrid LSTM-CNN network [34] or a regression forest [35]. The resulting networks speed up the generation of fluid flows up to three orders of magnitude compared to state-of-the-art fluid solvers. However, the methods developed in computer vision compute how the flow will develop over time and focus on the stability of such a solution instead of the accuracy in the predicted velocity field. In contrast, the discussed use case of UAV flight over complex terrain does not require time propagation because a global weather model provides time dependent boundary conditions, but it does require high wind field accuracy. In that context, Umetani and Bickel [36] present a machine learning approach for aerodynamic design that uses a Gaussian Progress Regressor to predict the drag coefficient of an object and the velocity field around it. The CNN proposed by Guo et al. [37] predicts the steady laminar flow around objects and achieves a speed up of up to four orders of magnitude compared to a Lattice Boltzmann Method solver [38]. All in all, the presented methods prove the ability of machine learning methods to replace and speed up current state-of-the-art fluid solvers. However, for accurately predicting the wind over complex terrain the methods still need to be adopted accordingly.

Acknowledgements

We'd like to thank Prof. Bruno Neining and David Braig from the Zurich University of Applied Sciences both for providing the LIDAR wind measurements and assistance in evaluating the data.

REFERENCES

- [1] N. Colella and G. Wenneker, "Pathfinder: Developing a solar rechargeable aircraft," *IEEE Potentials*, vol. 15, no. 1, pp. 18–23, 1996.
- [2] P. Oettershagen, J. Förster, L. Wirth, J. Ambühl, and R. Siegwart, "Meteorology-Aware Multi-Goal Path Planning for Large-Scale Inspection Missions with Solar-Powered Aircraft," *Journal of Aerospace Information Systems (submitted)*, 2018, <https://arxiv.org/abs/1711.10328>.
- [3] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, S. Leutenegger, K. Alexis, and R. Siegwart, "Design of small hand-launched solar-powered UAVs: From concept study to a multi-day world endurance record flight," *Journal of Field Robotics (JFR)*, vol. 34, no. 7, pp. 1352–1377, Oct. 2017.
- [4] J. Petrich and K. Subbarao, "On-board wind speed estimation for UAVs," in *AIAA Guidance Navigation and Control Conference*, 2011.
- [5] J. W. Langelaan, N. Alley, and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *Journal of Guidance Control and Dynamics*, vol. 34, no. 4, p. 1016, 2011.
- [6] H. J. Palanhandalam-Madapusi, A. Girard, and D. S. Bernstein, "Wind-field reconstruction using flight data," in *American Control Conference, 2008*. IEEE, 2008, pp. 1863–1868.
- [7] T. Larrabee, H. Chao, M. Rhudy, Y. Gu, and M. R. Napolitano, "Wind field estimation in UAV formation flight," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 5408–5413.
- [8] M. Xue, K. K. Droegemeier, and V. Wong, "The Advanced Regional Prediction System (ARPS)—A multi-scale nonhydrostatic atmospheric simulation and prediction model. Part I: Model dynamics and verification," *Meteorology and atmospheric physics*, vol. 75, no. 3-4, pp. 161–193, 2000.
- [9] M. Xue, K. K. Droegemeier, V. Wong, A. Shapiro, K. Brewster, F. Carr, D. Weber, Y. Liu, and D. Wang, "The Advanced Regional Prediction System (ARPS)—A multi-scale nonhydrostatic atmospheric simulation and prediction tool. Part II: Model physics and applications," *Meteorology and atmospheric physics*, vol. 76, no. 3-4, pp. 143–165, 2001.
- [10] R. Mott and M. Lehning, "Meteorological modeling of very high-resolution wind fields and snow deposition for mountains," *Journal of Hydrometeorology*, vol. 11, no. 4, pp. 934–949, 2010.
- [11] R. Mott, M. Schirmer, M. Bavay, T. Grünwald, and M. Lehning, "Understanding snow-transport processes shaping the mountain snow-cover," *The Cryosphere*, vol. 4, no. 4, pp. 545–559, 2010.
- [12] C. Hug, F. Faure, M. Lehning, S. Mobbs, N. Rader-

- schall, and J.-A. Hertig, “Small scale wind field simulations for the steep Gaudergrat ridge using CFX-4 and ARPS; influence of the boundary conditions and comparison with measurements,” *Hrvatski meteorološki časopis*, vol. 40, no. 40, pp. 46–49, 2005.
- [13] N. Raderschall, M. Lehning, and C. Schär, “Fine-scale modeling of the boundary layer wind field over steep topography,” *Water Resources Research*, vol. 44, no. 9, 2008.
- [14] “Documentation of the prognostic mesoscale model GRAMM (Graz Mesoscale Model) Vs. 16.1,” <https://lampx.tugraz.at/~gral/index.php/download>, Government of Styria, accessed: 2016-08-29.
- [15] D. Oettl, “A multiscale modelling methodology applicable for regulatory purposes taking into account effects of complex terrain and buildings on pollutant dispersion: a case study for an inner Alpine basin,” *Environmental Science and Pollution Research*, vol. 22, no. 22, pp. 17 860–17 875, 2015.
- [16] C. Ratto, R. Festa, C. Romeo, O. Frumento, and M. Galluzzi, “Mass-consistent models for wind fields over complex terrain: The state of the art,” *Environmental Software*, vol. 9, no. 4, pp. 247–268, 1994.
- [17] C. A. Sherman, “A mass-consistent model for wind fields over complex terrain,” *Journal of applied meteorology*, vol. 17, no. 3, pp. 312–319, 1978.
- [18] C. S. Sherman, “MATHEW: A mass-consistent wind field model,” Ph.D. dissertation, University of California/Livermore, 1978.
- [19] R. Walt, “Divergence-Free Wind Downscaling using Numerical Weather Prediction Data,” Master’s thesis, ETH Zurich, 2016.
- [20] “A User’s Guide for the CALMET Meteorological Model,” https://www.arpae.it/cms3/documenti/_cerca_doc/meteo/ambiente/manuale_calmet.pdf, accessed: 2016-08-28.
- [21] H. Truhetz, A. Gobiet, and G. Kirchengast, “Evaluation of a dynamic-diagnostic modelling approach to generate highly resolved wind fields in the Alpine region,” *Meteorologische Zeitschrift*, vol. 16, no. 2, pp. 191–201, 2007.
- [22] L. Morales, F. Lang, and C. Mattar, “Mesoscale wind speed simulation using CALMET model and reanalysis information: An application to wind potential,” *Renewable Energy*, vol. 48, pp. 57–71, 2012.
- [23] J. M. Forthofer, B. W. Butler, and N. S. Wagenbrenner, “A comparison of three approaches for simulating fine-scale surface winds in support of wildland fire management. Part I. Model formulation and comparison against measurements,” *International Journal of Wildland Fire*, vol. 23, no. 7, pp. 969–981, 2014.
- [24] N. S. Wagenbrenner, J. M. Forthofer, B. K. Lamb, K. S. Shannon, and B. W. Butler, “Downscaling surface wind predictions from numerical weather prediction models in complex terrain with WindNinja,” *Atmospheric Chemistry and Physics*, vol. 16, no. 8, pp. 5229–5241, 2016.
- [25] B. Müller, “Terrain-based 3d wind-field and meteorology modeling for environment-aware operation of unmanned aerial vehicles,” Master’s thesis, ETH Zurich, 2016.
- [26] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [27] P. Oettershagen, F. Achermann, B. Müller, D. Schneider, and R. Siegwart, “Towards Fully Environment-Aware UAVs: Real-Time Path Planning with Online Forecasting of Local 3D Wind Fields in Adverse Terrain,” *ARXIV*, 2017, <https://arxiv.org/abs/1712.03608>.
- [28] “MeteoSwiss: COSMO forecasting system,” <http://www.meteoswiss.admin.ch/home/measurement-and-forecasting-systems/warning-and-forecasting-systems/cosmo-forecasting-system.html>, accessed: 2017-04-08.
- [29] T. Hinzmann, T. Schneider, M. T. Dymczyk, A. Melzer, T. Mantel, R. Siegwart, and I. Gilitschenski, “Robust map generation for fixed-wing uavs with low-cost highly-oblique monocular cameras,” in *2016 IEEE International Conference on Intelligent Robots and Systems (IROS)*, October 2016.
- [30] A. Kuratle and G. Sauter, “Windprofilewindmessungen mit LIDAR im Gebirge,” Projektarbeit, ZHAW, 2015.
- [31] M. Hasler and J. Maron, “LIDAR-Anwendungen für die Aviaktik und die Windenergie,” Bachelorarbeit, ZHAW, 2016.
- [32] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler, “Deep Fluids: A Generative Network for Parameterized Fluid Simulations,” jun 2018. [Online]. Available: <http://arxiv.org/abs/1806.02071>
- [33] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, “Accelerating Eulerian Fluid Simulation With Convolutional Networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- [34] S. Wiewel, M. Becher, and N. Thuerey, “Latent-space physics: Towards learning the temporal evolution of fluid flow,” *CoRR*, vol. abs/1802.10123, 2018. [Online]. Available: <http://arxiv.org/abs/1802.10123>
- [35] L. Ladický, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross, “Data-driven fluid simulations using regression forests,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 199:1–199:9, Oct. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2816795.2818129>
- [36] N. Umetani and B. Bickel, “Learning three-dimensional flow for interactive aerodynamic design,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 89:1–89:10, Jul. 2018. [Online]. Available: <http://doi.acm.org/10.1145/3197517.3201325>
- [37] G. Xiaoxiao, L. Wei, and I. Francesco, “Convolutional neural networks for steady flow approximation,” *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016.
- [38] G. R. McNamara and G. Zanetti, “Use of the boltzmann equation to simulate lattice-gas automata,” *Phys. Rev. Lett.*, vol. 61, pp. 2332–2335, Nov 1988. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.61.2332>