Quadrotor Flight Simulation in a CFD-generated Urban Wind Field

Nicholas Kakavitsas[1], Andrew Willis[2], Ryan Jacobik[3], Mesbah Uddin[4], and Artur Wolek[4]

[1]Department of Mechanical Engineering, University of North Carolina at Charlotte Charlotte
[2]Department of Electrical and Computer Engineering, University of North
[3]Department of Mechanical, Aerospace, and Biomedical Engineering, University of Tennessee Knoxville Knoxville
[4]Department of Mechanical Engineering, University of North

March 19, 2024

Abstract

This paper presents a software pipeline that enables simulating a quadrotor's flight in realistic urban wind fields where complex wind phenomena are common and have significant impact on vehicle dynamics. The pipeline integrates the OpenStreetMap database for obtaining real-world building geometry, the OpenFOAM computational fluid dynamics (CFD) solver for computing a three-dimensional, steady, wind field, the Gazebo robotics simulation environment, and the PX4 software-in-the-loop autopilot. A 3D wind plugin is developed to interpolate a pre-computed CFD wind field at runtime during the simulation. The approach is demonstrated by comparing the flight performance of a quadrotor flying over a university campus environment with and without the wind field.

Quadrotor Flight Simulation in a CFD-generated Urban Wind Field

Nicholas Kakavitsas
Department of Mechanical Engineering
University of North Carolina at Charlotte
Charlotte, NC, 28263
nkakavit@charlotte.edu

Andrew Willis
Department of Electrical and Computer Engineering
University of North Carolina at Charlotte
Charlotte, NC, 28263
arwillis@charlotte.edu

Ryan Jacobik
Department of Mechanical, Aerospace, and Biomedical Engineering
University of Tennessee Knoxville
Knoxville, TN, 37996
rjacobik@vols.utk.edu

Mesbah Uddin
Department of Mechanical Engineering
University of North Carolina at Charlotte
Charlotte, NC, 28263
muddin@charlotte.edu

Artur Wolek
Department of Mechanical Engineering
University of North Carolina at Charlotte
Charlotte, NC, 28263
awolek@charlotte.edu

Abstract—**This paper presents a software pipeline that enables simulating a quadrotor's flight in realistic urban wind fields where complex wind phenomena are common and have significant impact on vehicle dynamics. The pipeline integrates the OpenStreetMap database for obtaining real-world building geometry, the OpenFOAM computational fluid dynamics (CFD) solver for computing a three-dimensional, steady, wind field, the Gazebo robotics simulation environment, and the PX4 software-in-the-loop autopilot. A 3D wind plugin is developed to interpolate a pre-computed CFD wind field at runtime during the simulation. The approach is demonstrated by comparing the flight performance of a quadrotor flying over a university campus environment with and without the wind field.**

TABLE OF CONTENTS

1. INTRODUCTION

Uncrewed aerial systems (UAS) are increasingly being deployed in urban environments for applications such as commercial and medical package delivery, public safety, and infrastructure inspection. Flight simulation of UAS plays an important role in supporting these applications by providing a means to evaluate safety, performance, and reliability in engineering design, pilot training, and mission planning. The urban environment is a challenging operating space for UAS, especially at lower, building-level altitudes near structures where complex, three-dimensional, urban wind patterns can reduce UAS performance, increase power consumption, and impact stability. Flight simulation in urban wind fields can be used to evaluate wind-aware feedback control strategies that compensate for disturbances and path planning algorithms that are designed to avoid or exploit wind conditions.

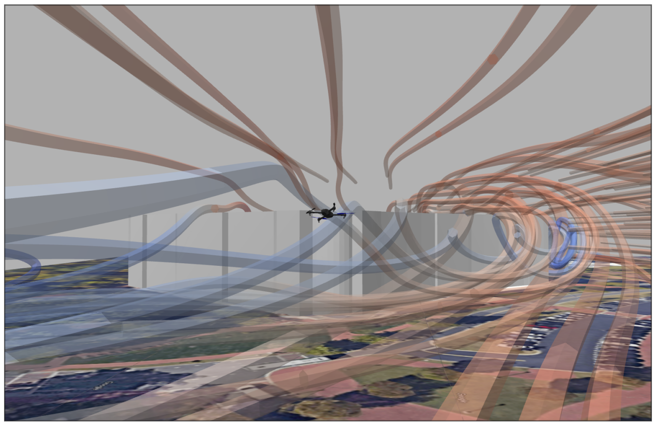

Figure 1: Screenshot of the customized Gazebo simulation environment developed in this work. The building geometry is imported from OpenStreetMap for a location on UNC Charlotte's campus. Translucent streamtubes generated using OpenFOAM indicate wind direction and magnitude (red corresponds to higher magnitudes and blue corresponds to lower magnitudes).

This paper presents a software framework that enables the simulation of UAS flight through a realistic urban environment with a steady wind field (see Fig. 1). The framework consists of four components: (1) a geometric environment model database (OpenStreetMap [1]), (2) a computational fluid dynamics (CFD) solver (OpenFOAM [2]), (3) a 3D UAS simulation environment (Gazebo [3] with the PX4 autopilot [4]), and (4) a custom simulation plugin that integrates a pre-computed CFD wind field with the simulator's vehicle dynamics. The OpenFOAM CFD solver captures complex flow field artifacts such as wind shear at roof top levels and street canyons with channelized flow. The OpenStreetMap geometric database allows geometric models to be generated automatically for arbitrary real-world locations (including 3D building geometries of urban areas). The open-source Gazebo simulator and PX4 software-in-the-loop (SITL) autopilot provide physically representative simulations of a UAS and its software and communication architecture. A custom Gazebo plugin uses the CFD solution to perturb the simulated UAS with injected wind disturbances. Autonomous flight tests are simulated in a urban wind field environment, with the capability for the pilot to see both the scene geometry and a visualization of the wind-flow field, through first-person-view goggles. The simulation framework also allows flight path planning using a standard mission planner (QGroundControl [5]).

Related Work

Computational studies of urban wind fields have demonstrated that turbulent wind can significantly impact the safety of drone operations [6], especially near buildings where gusts and wakes dominate [7]. To mitigate wind effects, prior work has investigated using CFD–based wind simulations for path planning [8–10] as well as for assessing control design methodologies (e.g., evaluating station-keeping performance near buildings [11]). Recently, micro weather and wind data providers have begun to provide mission planning and support services for drones and future urban air mobility vehicles operating in urban environments [12, 13]; however, such data services are not yet widely adopted. Machine learning predictions from CFD simulations have also been proposed to predict urban wind fields for small UAS flights [14]. Others have developed pre-computed CFD databases that can be generalized to different building morphologies [15].

A number of simulators have been created to integrate complex wind patterns with realistic flight dynamics and vehicle hardware/software system models (i.e., simulations of actuators, sensors, battery systems, and command and control architecture). Arguably the most realistic simulations include a mesh of the UAS within the CFD simulation and compute the flow as the vehicle moves through the environment [16, 17]. However, computational complexity limits these simulations to simplified (and usually open-loop) motions that are a few seconds in duration. To facilitate simulating longer flights and feedback-controlled UAS motion, a common approach is to simulate the urban wind field first (without the UAS) and then store the pre-computed wind field in memory to be queried at runtime as a UAS vehicle moves through the simulated environment. In the latter approach, local changes in wind velocity affect aerodynamic and thrust forces according to first-principles models [18–20]. CFD-based wind vehicle simulators can also capture the effect of wind velocity gradients over the span of the vehicle by using the velocity-point method [21]. Simulators have also been developed for other platforms in a similar manner by incorporating CFD-based wind simulations (e.g., for unmanned marine surface vessels [22]).

Visualizations of wind field data can also been used to aid human pilots that operate manned or remotely piloted aircraft. In [23], a pilot-in-the-loop study showed that adding wind indicators of direction/magnitude (measured point-wise at the current aircraft location) improved pilot confidence. Visualization can be in the form of two-dimensional or three-dimensional wind vector fields, streamtubes, or other volumetric indicators. For example, in [24] hazardous regions of turbulent air were rendered as translucent volumes in a display to assist helicopter pilots during ship deck landing.

Paper Contributions and Organization

The contributions of the paper are: (1) a software framework that uses the OpenFOAM computational fluid dynamics (CFD) solver to compute three-dimensional urban wind flow patterns for an arbitrary urban environment available from the OpenStreetMap database, and (2) integration of the urban wind velocity field, along with a stream-tube visualization, into a realistic quadrotor flight simulator (Gazebo) using software-in-the-loop (PX4) sensing, estimation, and control. The framework can be used to evaluate pre-programmed missions using a standard ground station planner (QGroundControl) or with first-person-view flight googles for pilot-in-the-loop flight simulations.

The remainder of the paper is organized as follows. Section 2 describes the simulation software architecture, including processing the building geometry, CFD wind flow modeling, and integrating the pre-computed wind with a Gazebo robotic simulator and the PX4 autopilot. Section 3 describes simulation results, and the paper is concluded in Sec. 4.

2. SOFTWARE ARCHITECTURE

This section presents the simulation software architecture that is summarized in Fig. 2.

Extracting Urban Environment Geometry

The software pipeline begins by obtaining a geometric model of the target environment by downloading a 3D geometric model of the region of interest via a query consisting of (latitude, longitude) (WGS84) coordinates and the desired radius (in meters). Automated tools access and extract a terrain model from the OpenStreetMap database [1] of urban environment geometries. Our `osm_mapgen` python script implementing this capability is available online[1]. The script produces a .STL file (a 3D geometry mesh) that can be visualized using software such as Blender [25], along with the bounding box dimensions (length, width, and height) of the geometry. An example of the geometry for a 250 meter radius around a location on UNC Charlotte's campus (latitude 35.310523°N and longitude 80.739100°W) is shown in Fig. 3 where satellite imagery is integrated into the model via Google Maps [26]. Note that the OpenStreetMap database has limited information concerning building heights and approximate values are used in the present study.

Computational Fluid Dynamics Simulations

To simulate the wind patterns in an urban environment, computational fluid dynamics (CFD) simulations were performed using OpenFOAM [2]. The urban geometry was imported as an .STL file generated by the process described above. The urban environment model is placed in a virtual wind tunnel for CFD analysis. As a rule of thumb, we select the simulation volume to a height of at least $2h$ where h is the height of the tallest building in the mesh. The building geometry can be rotated such that the wind blows in the positive x direction in the coordinate system of the solver.

[1]`https://github.com/uncc-visionlab/osm_mapgen`

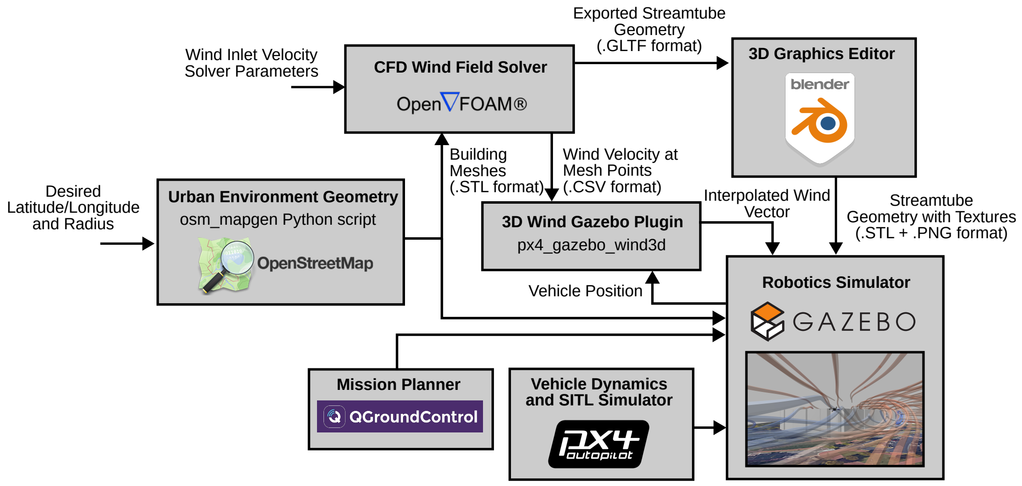
Exported Streamtube Geometry (.GLTF format)
Wind Inlet Velocity Solver Parameters
CFD Wind Field Solver
OpenFOAM®
3D Graphics Editor
blender
Building Meshes (.STL format)
Wind Velocity at Mesh Points (.CSV format)
Desired Latitude/Longitude and Radius
Urban Environment Geometry
osm_mapgen Python script
OpenStreetMap
3D Wind Gazebo Plugin
px4_gazebo_wind3d
Interpolated Wind Vector
Streamtube Geometry with Textures (.STL + .PNG format)
Robotics Simulator
GAZEBO
Vehicle Position
Mission Planner
QGroundControl
Vehicle Dynamics and SITL Simulator
PX4 autopilot

Figure 2: Diagram of overall software architecture.

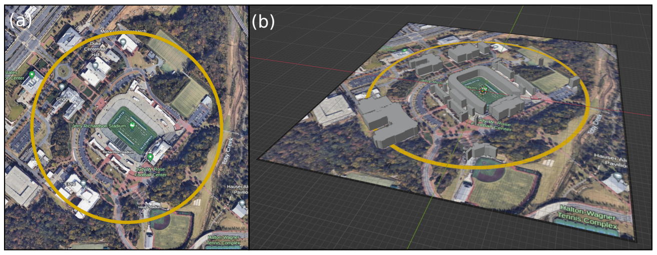

Figure 3: Example geometry imported from the OpenStreetMap database using the `osm_mapgen` package. Panel (a) shows the origin of the map located on UNC Charlotte's campus position and a 250 meters radius for capturing building geometry. The origin of the map (center) is located in the middle of the field in Jerry Richardson Stadium. Panel (b) shows a close up view of the 3D geometry imported with an underlying satellite image.

The upwind, downwind, and side faces of the volume are offset from the nearest vertical plane by at least $10h$. The bottom of the simulation volume is aligned with the ground plane. Since the geometry is imported with a radius R this gives a simulation volume with dimensions of approximate length/width of $2(R + h)$ and height $2h$ or more. The simulation volume used in this study is shown in Fig. 4 and had a length and width of 600 m and height of 50 m.

The following boundary conditions were applied: the left vertical plane (as viewed in Fig. 4) is a velocity inlet with a uniform freestream wind in the positive x direction. Nominally, this corresponds to an eastward wind (the geometry can be rotated to simulate other wind directions). In this work, we set the wind magnitude to be 10 m/s. The right vertical plane is the outlet, and the remaining four planes define the simulation volume and allow no flow across their boundaries. The simulation is similar to placing the urban geometry in a virtual wind tunnel. The geometry is first meshed using a hexahedral decomposition via the `BlockMesh` utility. In this simulation, we use 60 cells in the x and y directions, and 10 cells in the z direction for the coarse mesh. Next, `SnappyHexMesh` provides a more refined meshing in the vicinity of the building geometry (which we set as the volume of 500 m length/width and 50 m height centered on the origin). A more efficient mesh could be constructed by refining only around individual buildings. However, tuning the accuracy of the CFD simulation is outside the scope of this paper, and will be addressed in future work.

In this study, the incompressible CFD solver `simpleFoam`, coupled with the $k - \varepsilon$ turbulence model as proposed by Launder and his coworkers [27–29] (referred to as SKE), is employed to solve the Reynolds Averaged Navier-Stokes (RANS) mass and momentum transport equations. The simulation is conducted for a total of 400 iterations to allow the flow to statistically converge. One simulation takes approximately 1 hour of CPU time on a laptop computer (12th Gen Intel Core i7-12800H processor with 24M Cache, up to 4.80 GHz CPU, and 32 GB RAM).

While investigating the efficacy of the turbulence modeling approach for the current flow configuration is beyond the scope of this paper, a brief discussion on the rationale behind using the SKE model in this study is included. The problem considered in this paper involves the interaction of the wake behind an array of buildings with a moving aerial vehicle, for which no experimental data is available to guide the choice of the turbulence model. In the absence of such data, canonical cases, such as the wall-mounted cylinder [30] and cube [31–34] may serve as alternative benchmarks. Although extensive literature exists on CFD correlation studies for these simple flow configurations, our firsthand experience with these cases, as detailed in our previous work on the wall-mounted cylinder [30] and cube [34], indicates that commonly employed turbulence models consistently fall short of achieving complete accuracy when compared against experimental studies [33] and Direct Numerical Studies (DNS) [31, 32]. However, these works suggest that among the various turbulence models, the Realizable $k - \varepsilon$ model, a variant of the SKE model proposed by Shish et al. [35], appears to be superior in predicting flow characteristics within the wake and stagnation regions. Considering the demonstrated robustness and numerical stability of the SKE model in existing literature, along with our firsthand positive experiences, we have chosen to proceed with this model for this work. This decision is driven by the primary objectives of our study, aiming to mitigate uncertainties and solution divergence (caused by the choice of turbulence model employed) of the finite volume numerical schemes utilized in CFD simulations. However, caution should be used in applying the SKE model to predict forces and moments on a vehicle (not considered here, but of interest in future work). Studies have shown [36] that although the SKE model's variants demonstrated reasonable accuracy in predicting the aerodynamic characteristics of vehicle (in their case a ground vehicle) at zero degrees yaw, only the SST $k - \omega$ model developed by Menter et al. [37–39] exhibited consistent prediction trends at non-zero yaw angles, aligning closely with experimentally observed data. Moreover, amongst all the models tested in [36] only the SST $k - \omega$ model predicted two critical phenomena: (1) a drag loss as the vehicle pitched, and (2) a relatively consistent prediction of front downforce, side forces, and consequently, moments.

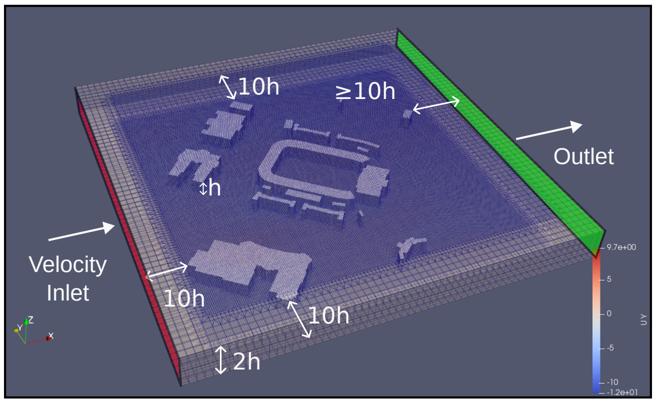

Figure 4: Example hexahedral mesh of the control volume used for CFD simulation. The volume mesh near urban buildings is finer than in regions of empty space on the boundary of the simulation volume.

Once the OpenFOAM CFD simulation is completed, the results are loaded into the open-source visualization utility Paraview. Paraview allows visualization of the streamtubes by the application of a streamlines filter. The streamlines filter uses a spherical pattern of starting locations and integrates the particle velocity both forward and backward to visualize the flow (see Fig. 5). The resulting streamlines

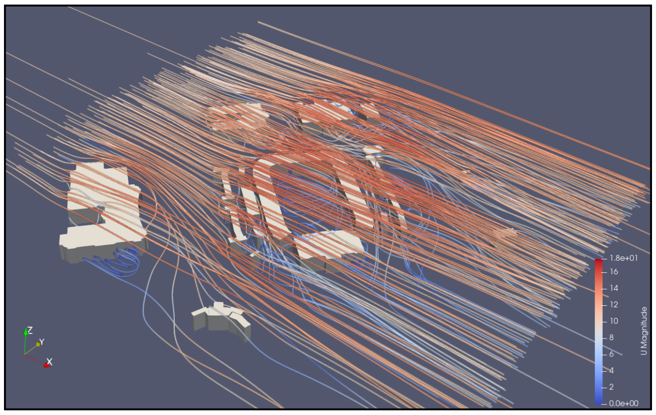

Figure 5: CFD simulation visualization in Paraview of wind flow in positive x direction at 10 m/s over the urban environment. Streamtube colors represent wind magnitude.

are made more prominent by applying the streamtubes filter to increase their diameter. The streamtubes can be colorized to indicate various aspects of the flow, such as the total wind magnitude (Fig. 5) or only one component of the velocity (e.g., perpendicular to the inflow direction as shown in Fig. 6.The geometry is then exported as a .GLTF file (3D mesh) and a .CSV (comma-separated value) file records the three-dimensional velocity vector at each vertex in the mesh at the final simulation iteration.

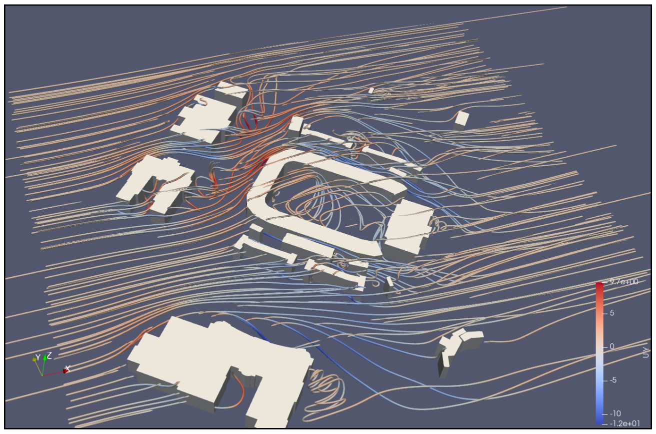

Figure 6: CFD simulation visualization in Paraview, similar to Fig. 5. Streamtube colors represent wind magnitude components in the y direction. For a vehicle traveling in the positive x direction the red regions correspond to high cross-winds. Streamtube above building level height are not shown to illustrate cross-flow within inter-building regions.

The current implementation considers only steady-state wind patterns that may ignore or average-out transient turbulent flow structures. Future work may consider including other turbulence modelling available in OpenFOAM, such as, the Unsteady Reynolds Averaged Simulation, and Detached Eddy Simulation (DES) [40], and using unsteady flow solutions for time-varying interpolation at runtime. The 1 hour time required to perform a CFD simulation prohibits the use of the framework for practical flight planning/optimization (since real-world wind patterns can change more quickly). One approach to address this challenge could be to precompute the solution for a known building geometry in a matrix of flow fields that can be interpolated. A future research direction of interest is to develop reduced order models (ROM), following the approaches of Mohrfeld and Uddin [41], or Misar et al. [42], that can provide fast, accurate, and reliable flow predictions utilizing computational resources of the onboard controllers on the moving aerial vehicle.

Gazebo Environment and Visualization

Gazebo [3] is an open source advanced robotic simulator that provides a plugin-based interface to physics and rendering engines, an extensive set of sensor and vehicle models, and an asynchronous messaged passing architecture. In this work, we utilize the default 3DR Iris quadrotor model available via the PX4 autopilot SITL packages that integrate with the Gazebo simulator. Blender is used to import the streamtube .GLTF files generated from the CFD simulation (Sec. 2). The geometry is then exported as an .STL along with separate texture files to enable correct rendering in Gazebo of the streamtube colors and transparency. Once loaded into Gazebo, the simulation environment appears as shown in Fig. 1. An optional feature, shown in Fig. 1, is to include the static Google maps plugin [43] that creates satellite imagery textures which are overlayed on the ground plane and can be used for image-aiding navigation simulation [44]. The simulation environment can also be used to control the quadrotor with a pilot-in-the-loop. FPV goggles were used to connected to create an immersive flight experience as shown in Fig. 7, by mirroring the scene on the television into the goggles.

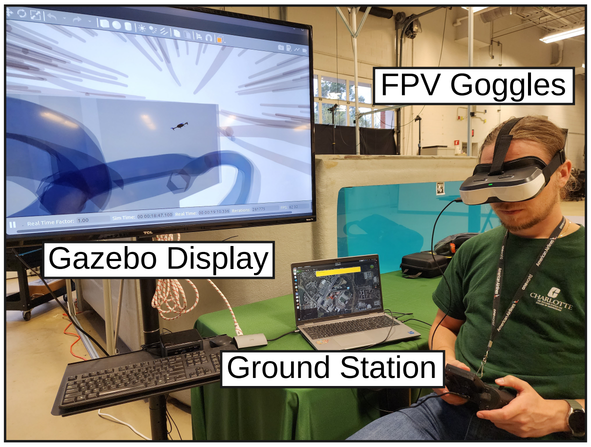

Figure 7: The drone was piloted in the Gazebo simulation environment with the aid of FPV goggles to create an immersive experience.

Quadrotor Simulator and Three-dimensional Wind Plugin

The PX4 autopilot [4] along with QGroundControl (QGC) station are popular open-source software tools used by UAS operators and are adopted in this work to support SITL simulation. The `px4_gazebo_wind3d` wind plugin[2] was developed to modify the wind field experienced by the quadrotor and is an extension of the existing PX4 wind plugin. First, the plugin reads the wind velocity data imported from the .CSV file and stores the information into an efficient k-d tree spatial partitioning data structure. At each instant that the wind plugin is called, the k-d tree is queried with the current $\boldsymbol{x} = [x, y, z]^{\mathrm{T}}$ location of the quadrotor to return N of the nearest vertices locations $\boldsymbol{v}_i = [v_x, v_y, v_z]^{\mathrm{T}}$ for $i = 1, \ldots, N$ along with the corresponding wind velocities at each vertex $\boldsymbol{u}_i = [u_x, u_y, u_z]^{\mathrm{T}}$. The interpolated wind velocity at $\boldsymbol{x}$ is then calculated as

$$\bar{\boldsymbol{u}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{N}(1/d_i^2)\boldsymbol{u}_i}{\sum_{i=1}^{N}(1/d_i^2)} , \tag{1}$$

where $d_i = ||\boldsymbol{x} - \boldsymbol{v}_i||$ is the Euclidean distance between the query point and each vertex. The interpolated wind velocity is published on the existing wind topic and is consumed by downstream processes related to the quadrotor dynamics.

3. Simulation Results and Discussion

To illustrate the simulation framework the standard Iris quadrotor model available with PX4 Gazebo-Classic SITL was flown in the aforementioned urban environment and wind field (see Fig. 3 and Fig. 5). The quadrotor flight consisted of a lawnmower type pattern planned using the QGroundControl mission planner survey plan function. Two autonomous flights were conducted, one in the absence of wind and one with the urban wind field enabled. The parameters used to generate the survey pattern were an altitude of 75 ft, a trackline spacing of 150 ft, and a trackline angle of 311 degrees. Figure 8 illustrates the resulting flight paths in both cases. The path in both cases is very similar, indicating the wind magnitude (inlet velocity of 10 m/s) did not significantly impede the quadrotor's ability to closely follow preprogrammed waypoints at the desired speed of 5 m/s. A video recording of a fragment of the simulated flight is available online [3].

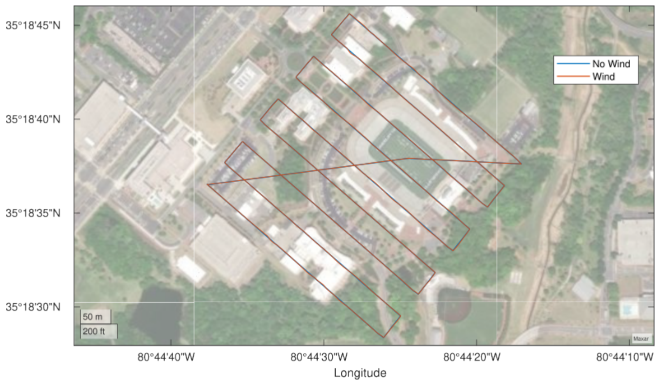

Figure 8: Lawnmower pattern used to test the effects of the urban wind field on the Iris quadrotor model in the PX4 SITL Gazebo simulation environment.

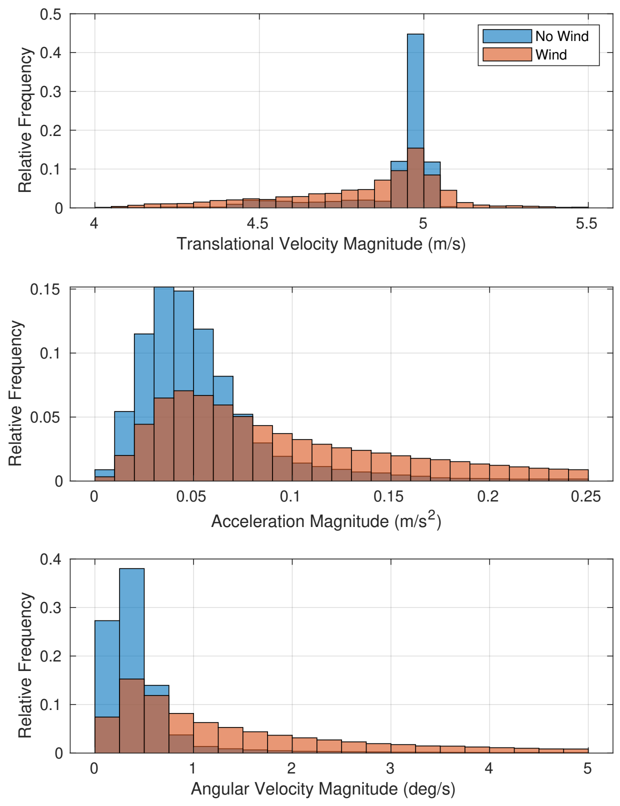

Figure 9: Relative frequency histogram plots for translational velocity magnitude, acceleration magnitude, and angular velocity magnitude, corresponding to the survey pattern in Fig. 8 in the presence and absence of the urban wind field.

Flight data logs were collected using the standard PX4 logging format (.ULG log files) and parsed with the python utility `ulog2csv` [45]. Flight logs contained information such as the vehicle's GPS position, velocity, angular rate, acceleration, battery voltage, and battery capacity.

The two flights were compared on the basis of translational

[2]`https://github.com/uncc-visionlab/px4_gazebo_wind3d`

[3]`https://www.youtube.com/watch?v=CbQeVzHut7c`

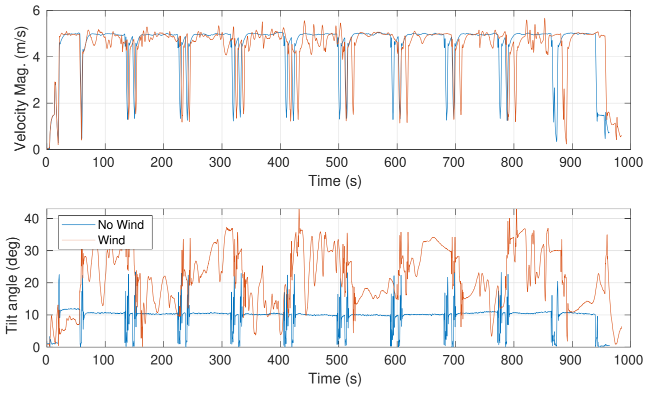

Figure 10: Comparison of velocity magnitude and tilt angle over time during flights without and with wind.

velocity magnitude, acceleration vector magnitude, and angular velocity magnitude, as shown in the relative frequency histogram plots in Fig. 9. The relative frequency can be interpreted as a empirical probability of a sample occurring within a particular range. Only a subset of each variable's range is plotted in these histograms for clarity. The histogram results indicate that the translational velocity during the flight without wind was consistently near the commanded value of 5 m/s. With wind, the quadrotor's speed had greater variability, including lower speeds for a significant portion of the survey. Both the acceleration magnitude and angular velocity magnitude were generally larger for the case with wind than without wind. Larger acceleration and angular velocity magnitude can be indicative of more difficult and dangerous flight conditions, as expected in the presence of a complex urban wind field.

Figure 10 plots the velocity magnitude and the tilt angle as a function of time during the two flights. The tilt angle depends on the roll angle ϕ and pitch angle θ [46]:

$$\lambda = \text{acos}\left(\frac{-\boldsymbol{\eta}\cdot(\boldsymbol{b}_x \times \boldsymbol{b}_y)}{||\boldsymbol{\eta}||\cdot||\boldsymbol{b}_x \times \boldsymbol{b}_y||}\right), \tag{2}$$

where $\boldsymbol{b}_x = [0, \cos\phi, \sin\phi]^{\mathrm{T}}$, $\boldsymbol{b}_y = [\cos\theta, 0, -\sin\theta]^{\mathrm{T}}$, and $\boldsymbol{\eta} = [0, 0, 1]^{\mathrm{T}}$. The total duration of the flight from takeoff to landing was 964.0 sec. (without wind) and 984.8 sec. (with wind). The velocity magnitude over time exhibits greater variability with wind than without wind around the nominal value of 5 m/s during the long legs of the survey. Reductions in speed down to about 1.5 m/s during turn maneuvers are observed in both cases. Without wind the quadrotor is nominally at a tilt angle of 10 deg. throughout most of the flight (except during turns, takeoff, and landing). With wind, the tilt angle varies up to 40 deg. at some instants.

4. Conclusion

A software pipeline was developed that enables simulating quadrotor motion through an urban wind field over arbitrary building geometry data obtained from the OpenStreetMap database specified by a GPS origin location and radius. CFD simulations using OpenFOAM were used to compute the steady-state, three-dimensional wind velocity vector field. The vector field is used to create streamtube visualizations and interpolated in real-time in a PX4-based software-in-the-loop Gazebo flight simulation environment. The approach was demonstrated using building geometry corresponding to a portion of UNC Charlotte's campus and assuming a wind inlet velocity of 10 m/s to produce a simulation environment in which both autonomous flight and manual flight can be tested. Two autonomous flights, consisting of a lawnmower style survey mission, were conducted in this flight simulation environment to compare flight telemetry with and without the wind field. When the wind field was active, the magnitude of the translational velocity, acceleration, and angular velocity displayed higher variance than when the wind field was disabled.

Future work may consider enhancing the realism of the simulation by incorporating terrain from digital elevation models, using a non-uniform freestream to model earth's boundary layer, enabling unsteady flow fields, applying textures to the building geometries, and integrating reduced-order models that more accurately capture quadrator-wind interaction dynamics. For ease-of-use the processing pipeline can also be further automated by scripting several of the steps that currently require user interaction. The simulation environment can be used to evaluate various autonomous control/planning strategies. Pilot-in-the-loop studies can also evaluate whether visualization of flow conditions (e.g., via streamtube or other 3D volumetric displays) aid in improving safety or performance.

Acknowledgments

The work was supported in part by the National Science Foundation Award No. 2301475.

References

[1] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017, accessed: 2023-10-05.

[2] H. Jasak, "OpenFOAM: Open source CFD in research and industry," *International Journal of Naval Architecture and Ocean Engineering*, vol. 1, no. 2, pp. 89–94, 2009.

[3] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 2149–2154.

[4] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*, 2015, pp. 6235–6240.

[5] Dronecode Foundation, "QGC - QGroundControl - Drone Control," http://qgroundcontrol.com/, 2023, accessed: 2023-10-05.

[6] S. Giersch, O. El Guernaoui, S. Raasch, M. Sauer, and M. Palomar, "Atmospheric flow simulation strategies to assess turbulent wind conditions for safe drone operations in urban environments," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 229, pp. 1–20, 2022.

[7] A. Mohamed, M. Marino, S. Watkins, J. Jaworski, and A. Jones, "Gusts encountered by flying vehicles in proximity to buildings," *Drones*, vol. 7, no. 1, pp. 22–48, 2023.

[8] J. Ware and N. Roy, "An analysis of wind field es-

timation and exploitation for quadrotor flight in the urban canopy layer," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, 2016, pp. 1507–1514.

[9] J. Patrikar, V. Dugar, V. Arcot, and S. Scherer, "Real-time motion planning of curvature continuous trajectories for urban UAV operations in wind," in *Proceedings of the 2020 International Conference on Unmanned Aircraft Systems*, 2020, pp. 854–861.

[10] M. W. Orr, S. J. Rasmussen, E. D. Karni, and W. B. Blake, "Framework for developing and evaluating mav control algorithms in a realistic urban setting," in *Proceedings of the 2005 American Control Conference.*, 2005, pp. 4096–4101.

[11] S. A. Raza, M. Sutherland, M. Etele, and G. Fusina, "Experimental validation of quadrotor simulation tool for flight within building wakes," *Aerospace Science and Technology*, vol. 67, pp. 169–180, 2017.

[12] TruWeather Solutions, "TruWeather Solutions Prototypes Urban Weather Sensing Infrastructure," https://truweathersolutions.com/weather-sensing-infrastructure/, accessed: 2023-10-05.

[13] M. Gianfelice, H. Aboshosha, and T. Ghazal, "Real-time wind predictions for safe drone flights in Toronto," *Results in Engineering*, vol. 15, pp. 1–15, 2022.

[14] R. K. Vuppala and K. Kara, "Wind field prediction in urban spaces for small unmanned aerial systems using convolutional autoencoders," in *Proceedings of the 2022 AIAA AVIATION Forum*, 2022, pp. 1–13.

[15] D. Galway, J. Etele, and G. Fusian, "Development and implementation of an urban wind field database for aircraft flight simulation," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 103, pp. 73–85, 2012.

[16] M. A. Razzak and M. Damodaran, "Computational multiphysics platform for virtual controlled flight of quadrotor unmanned aerial vehicles," in *Proceedings of the 2022 AIAA AVIATION Forum*, 2022.

[17] M. Uddin, S. Nichols, C. Hahn, A. Misar, S. Desai, N. Tison, and V. Korivi, "Aerodynamics of landing maneuvering of an unmanned aerial vehicle in close proximity to a ground vehicle," SAE Technical Paper, Tech. Rep. No. 2023-01-0118, 2023.

[18] B. Davoudi, E. Taheri, K. Duraisamy, B. Jayaraman, and I. Kolmanovsky, "Quad-rotor flight simulation in realistic atmospheric conditions," *AIAA Journal*, vol. 58, no. 5, pp. 1992–2004, 2020.

[19] A. Harmat, I. Sharf, and M. Trentini, "A hybrid particle/grid wind model for realtime small UAV flight simulation," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3780–3785.

[20] M. Sutherland, J. Etele, and G. Fusina, "Urban wake-field generation using large-eddy simulation for application to quadrotor flight," *Journal of Aircraft*, vol. 53, no. 5, pp. 1224–1236, 2016.

[21] B. Z. Cybyk, B. E. McGrath, T. M. Frey, D. G. Drewry, J. F. Keane, and G. Patnaik, "Unsteady airflows and their impact on small unmanned air systems in urban environments," *Journal of Aerospace Information Systems*, vol. 11, no. 4, pp. 178–194, 2014.

[22] M. Paravisi, D. H. Santos, V. Jorge, G. Heck, L. M. Gonçalves, and A. Amory, "Unmanned surface vehicle simulator with realistic environmental disturbances," *Sensors*, vol. 19, no. 5, pp. 1–20, 2019.

[23] A. Tabassum, H. Bai, and N. Fala, "A study on workload assessment and usability of wind-aware user interface for small unmanned aircraft system remote operations," *arXiv preprint arXiv:2309.04543*, pp. 1–18, 2023.

[24] C. R. Aragon, "Using visualization in cockpit decision support systems," *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2396–2401, 2005.

[25] Blender Online Community, "Blender - a 3D modelling and rendering package," http://www.blender.org, accessed: 2023-10-05.

[26] Google Maps, "UNC Charlotte Campus," https://maps.google.com, accessed: 2023-10-05.

[27] W. P. Jones and B. E. Launder, "The prediction of laminarization with a two-equation model of turbulence," *International Journal of Heat and Mass Transfer*, vol. 15, no. 2, pp. 301–314, 1972.

[28] B. E. Launder and B. I. Sharma, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc," *Letters in Heat and Mass Transfer*, vol. 1, no. 2, pp. 131–137, 1974.

[29] W. Jones and B. E. Launder, "The prediction of laminarization with a two-equation model of turbulence," *International Journal of Heat and Mass Transfer*, vol. 15, no. 2, pp. 301–314, 1972.

[30] P. Davis, A. Rinehimer, and M. Uddin, "A comparison of rans-based turbulence modeling for flow over a wall-mounted square cylinder," in *20th Annual Conference of the CFD Society of Canada*, vol. 8, 2012.

[31] A. Yakhot, H. Liu, and N. Nikitin, "Turbulent flow around a wall-mounted cube: A direct numerical simulation," *International Journal of Heat and Fluid Flow*, vol. 27, no. 6, pp. 994–1009, 2006.

[32] A. Curley and M. Uddin, "Direct numerical simulation of turbulent flow around a surface mounted cube," in *22nd AIAA Computational Fluid Dynamics Conference*, 2015, pp. 1–12.

[33] I. Castro and A. Robins, "The flow around a surface-mounted cube in uniform and turbulent streams," *Journal of Fluid Mechanics*, vol. 79, no. 2, pp. 307–335, 1977.

[34] M. C. Goldbach and M. Uddin, "High-resolution RANS simulations of flow past a surface-mounted cube using eddy-viscosity closure models," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 4, no. 1, p. 011005, 2019.

[35] T.-H. Shih, W. W. Liou, A. Shabbir, Z. Yang, and J. Zhu, "A new $k - \varepsilon$ eddy viscosity model for high Reynolds number turbulent flows," *Computers & Fluids*, vol. 24, no. 3, pp. 227–238, 1995.

[36] C. Fu, C. P. Bounds, C. Selent, and M. Uddin, "Turbulence modeling effects on the aerodynamic characterizations of a NASCAR Generation 6 racecar subject to yaw and pitch changes," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, no. 14, 2019.

[37] F. R. Menter, "Zonal two equation k-ω turbulence models for aerodynamic flows," in *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*, 1993, p. 2906.

[38] ——, “Two-equation eddy-viscosity turbulence models for engineering applications,” *AIAA Journal*, vol. 32, no. 8, pp. 1598–1605, 1994.

[39] F. R. Menter, M. Kuntz, and R. Langtry, “Ten years of industrial experience with the SST turbulence model,” *Turbulence, Heat and Mass Transfer*, vol. 4, no. 1, pp. 625–632, 2003.

[40] M. S. Gritskevich, A. V. Garbaruk, J. Schütze, and F. R. Menter, “Development of DDES and IDDES formulations for the k-ω shear stress transport model,” *Flow Turbulence and Combustion*, vol. 88, no. 3, pp. 431–449, 2012.

[41] J. Mohrfeld Halterman and M. Uddin, “Systematic reduced order model development of a pitching NACA0012 airfoil,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 235, no. 4, 2021.

[42] A. Misar, N. A. Tison, V. M. Korivi, and M. Uddin, “Application of the DMD approach to high-Reynolds-number flow over an idealized ground vehicle,” *Vehicles*, vol. 5, no. 2, pp. 656–681, 2023.

[43] Gazebo, “Gazebo : Tutorial : Ground plane model with satellite images,” http://classic.gazebosim.org/tutorials?tut=static_map_plugin&cat=build_world, accessed: 2023-10-05.

[44] A. R. Willis, K. M. Brink, and K. M. Dipple, “ROS georegistration: Aerial multi-spectral image simulator for the robot operating system,” *ArXiv*, vol. abs/2201.07863, pp. 1–7, 2022.

[45] PX4 Developers, “pyulog,” https://github.com/PX4/pyulog/tree/main#readme, 2023, accessed: 2023-10-05.

[46] P. Abichandani, D. Lobo, G. Ford, D. Bucci, and M. Kam, “Wind measurement and simulation techniques in multi-rotor small unmanned aerial vehicles,” *IEEE Access*, vol. 8, pp. 54 910–54 927, 2020.

Biography

***Nicholas Kakavitsas** received his B.S. and M.S. degrees in mechanical engineering from UNC Charlotte. He is currently pursuing his Ph.D., and is serving as a Research Assistant in the Autonomous Robots and Systems Laboratory. He has spent time interning as an A320 Fleet Engineer at American Airlines, and as a UAS Engineer at USASOC. His research focuses on multi-agent wind field estimation along with wind-aware path planning for autonomous quadrotors operating in urban environments.*

***Andrew Willis** received B.S. degrees in Electrical Engineering (EE) and Computer Science from WPI (94, 95), M.S. degrees in EE and Applied Math from Brown University (2001, 2003) a Ph.D degrees in Engineering Science from Brown University (2004). He was a Postdoctoral Fellow with Brown university (2005) and subsequently joined the faculty of the Electrical and Computer Engineering department at the University of North Carolina at Charlotte where he is an Associate Professor. Dr. Willis has been a senior researcher for the National Academy of Science (2015, 2016) and holds research positions in Physics and Optical Sciences (UNC Charlotte) and Mechanical and Aerospace (UFL). He works on problems in robotics, computer vision, radar, deep learning and meta-surface antennas.*

***Ryan Jacobik** is an undergraduate Aerospace Engineering student at the University of Tennessee – Knoxville. He is a senior who will graduate in May 2024 and be commissioned into the United States Army. Ryan was an undergraduate research student through the SERVE program at UNC Charlotte and aided in CFD research and simulations. Ryan wishes to continue his aerospace research as a member of the Army Space Operations career field.*

***Mesbah Uddin** is a Professor of Mechanical Engineering at the University of North Carolina at Charlotte, where he leads the ”Digital Design and Optimization (DDO)” research initiative. DDO is dedicated to enhancing Charlotte’s collaborations with defense and security-related private firms. Previously, he served as the Director of the North Carolina Motorsports and Automotive Research Center, as well as a member of the North Carolina Governor’s Motorsports Advisory Council from 2012 to 2017. Dr. Uddin actively participates in prominent professional societies, including SAE, AIAA, ASME, and ASEE. Currently, he serves as the Chair of SAE’s Road Vehicle Aerodynamics Standards Committee.*

***Artur Wolek** received the B.S. and Ph.D. degrees in Aerospace Engineering from Virginia Tech, Blacksburg, VA, USA, in 2010 and 2015, respectively. He was a Postdoctoral Fellow with the Naval Research Laboratory (2015–2018) and with the University of Maryland (2018–2020). He is currently an assistant professor of Mechanical Engineering at the University of North Carolina at Charlotte. His research interests include vehicle dynamics, control, and path planning for atmospheric and ocean vehicles.*