RotorPy: A Python-based Multirotor Simulator with Aerodynamics for Education and Research

Penn Engineering GRASP LABORATORY General Robotics, Automation, Sensing & Perception Lab

Spencer Folk*, James Paulos, and Vijay Kumar

*corresponding author: sfolk@seas.upenn.edu

Overview

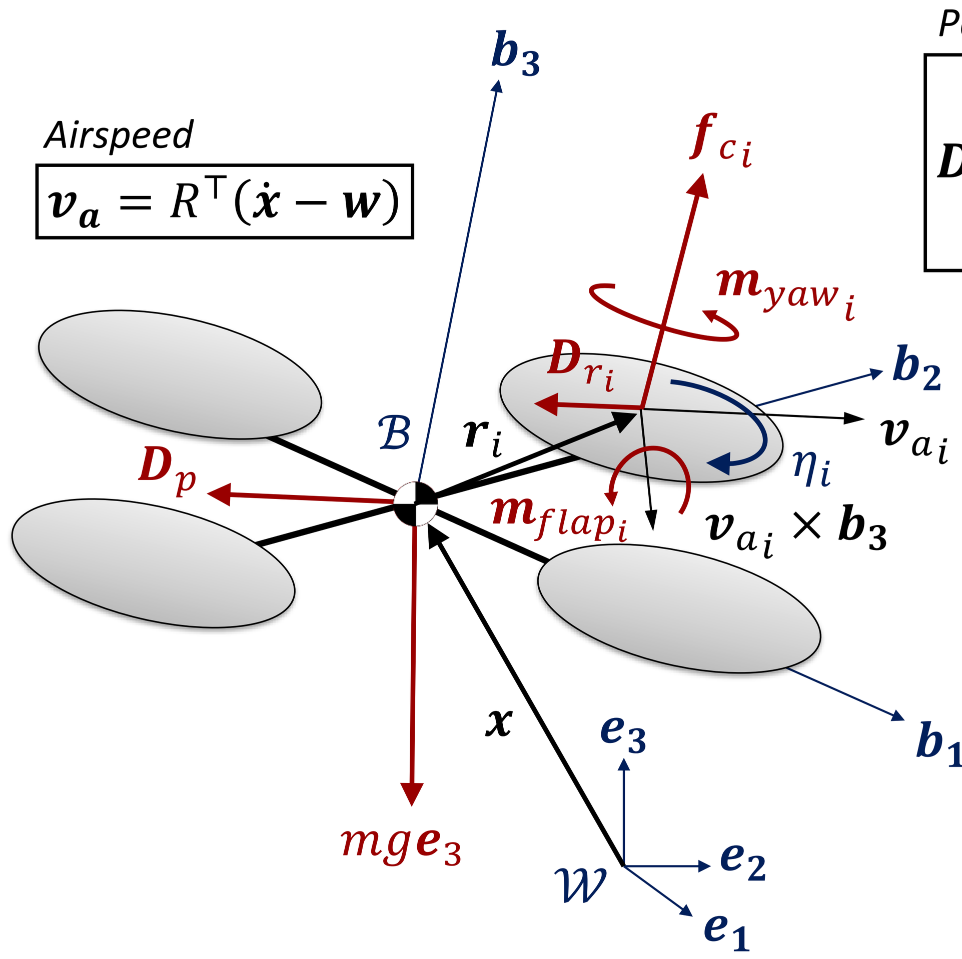
- **UAV simulators** play an increasingly important role in education and research of estimation, planning, and control for aerial robotics. **Existing codebases can be complex and elaborate, making it hard to install, interpret, and incorporate into your workflow.**
- We present ***RotorPy**:* an **opensource UAV simulator** that is **written entirely in Python**—simplifying the installation and usage—and structured to be **accessible** to a much broader audience.
- **Key features:** 1) Generic multirotor model with lumped parameter **aerodynamic models, motor dynamics, and input saturation**; 2) **Realistic sensor models** with noise and bias; 3) **Obstacle map** and occupancy grid generation; 4) Various **wind profiles** (spatial, temporal, or both).

Aerodynamic Modeling

- We use **lumped parameter models** of the **drag-induced forces and torques** on both the airframe and rotors.

Airspeed

$$\boldsymbol{v_a} = R^\top(\dot{\boldsymbol{x}} - \boldsymbol{w})$$



Parasitic (frame) drag

$$\boldsymbol{D}_p = -\begin{bmatrix} c_{D_x} & 0 & 0 \\ 0 & c_{D_y} & 0 \\ 0 & 0 & c_{D_z} \end{bmatrix} |\boldsymbol{v}_a| \boldsymbol{v}_a$$

Rotor drag + Induced thrust losses

$$\boldsymbol{D}_{r_i} = -\begin{bmatrix} k_d & 0 & 0 \\ 0 & k_d & 0 \\ 0 & 0 & k_z \end{bmatrix} \eta_i \boldsymbol{v}_{a_i}$$

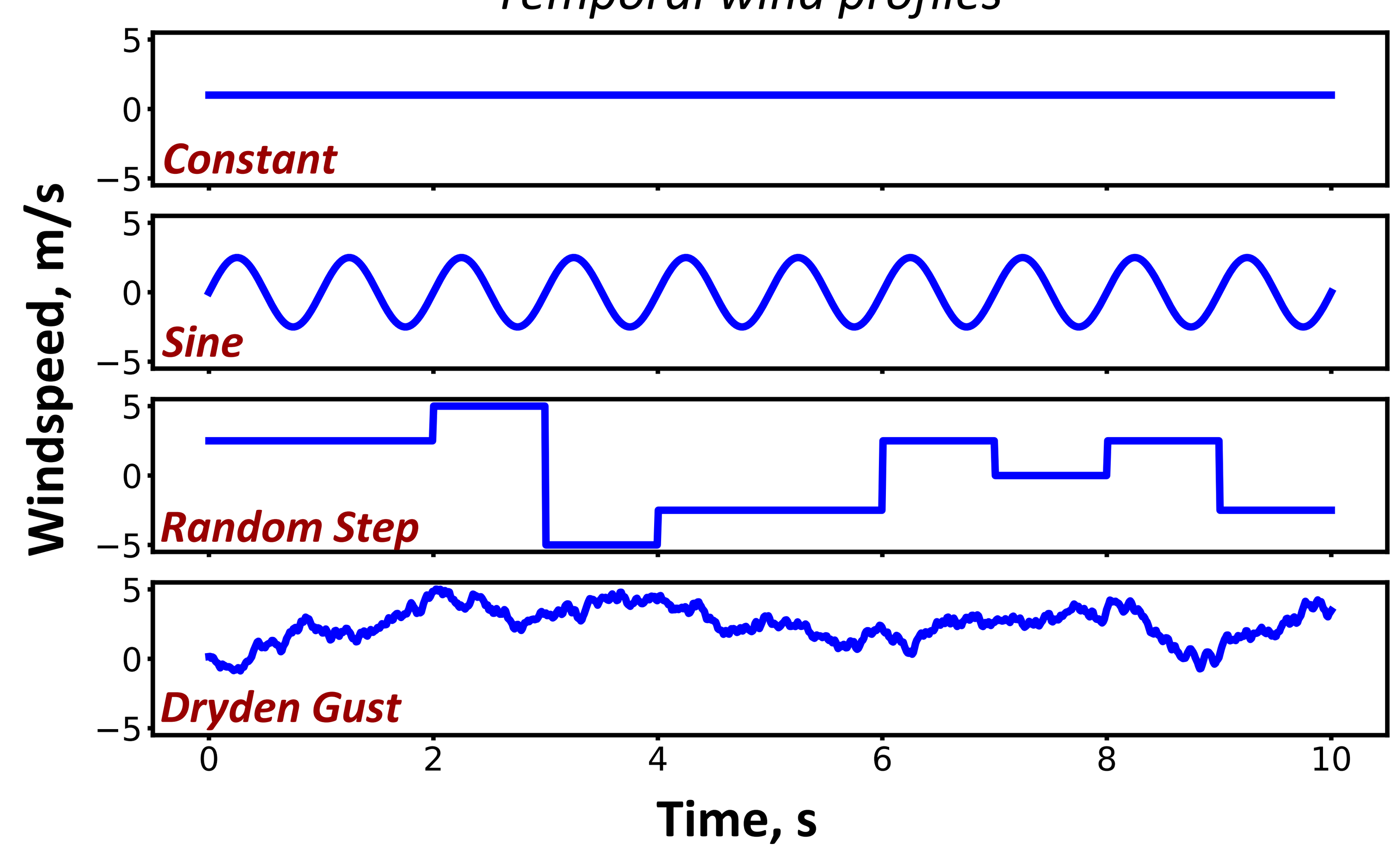
Flapping-induced moments

$$\boldsymbol{m}_{flap_i} = -k_{flap} \eta_i (\boldsymbol{v}_{a_i} \times \boldsymbol{b}_3)$$

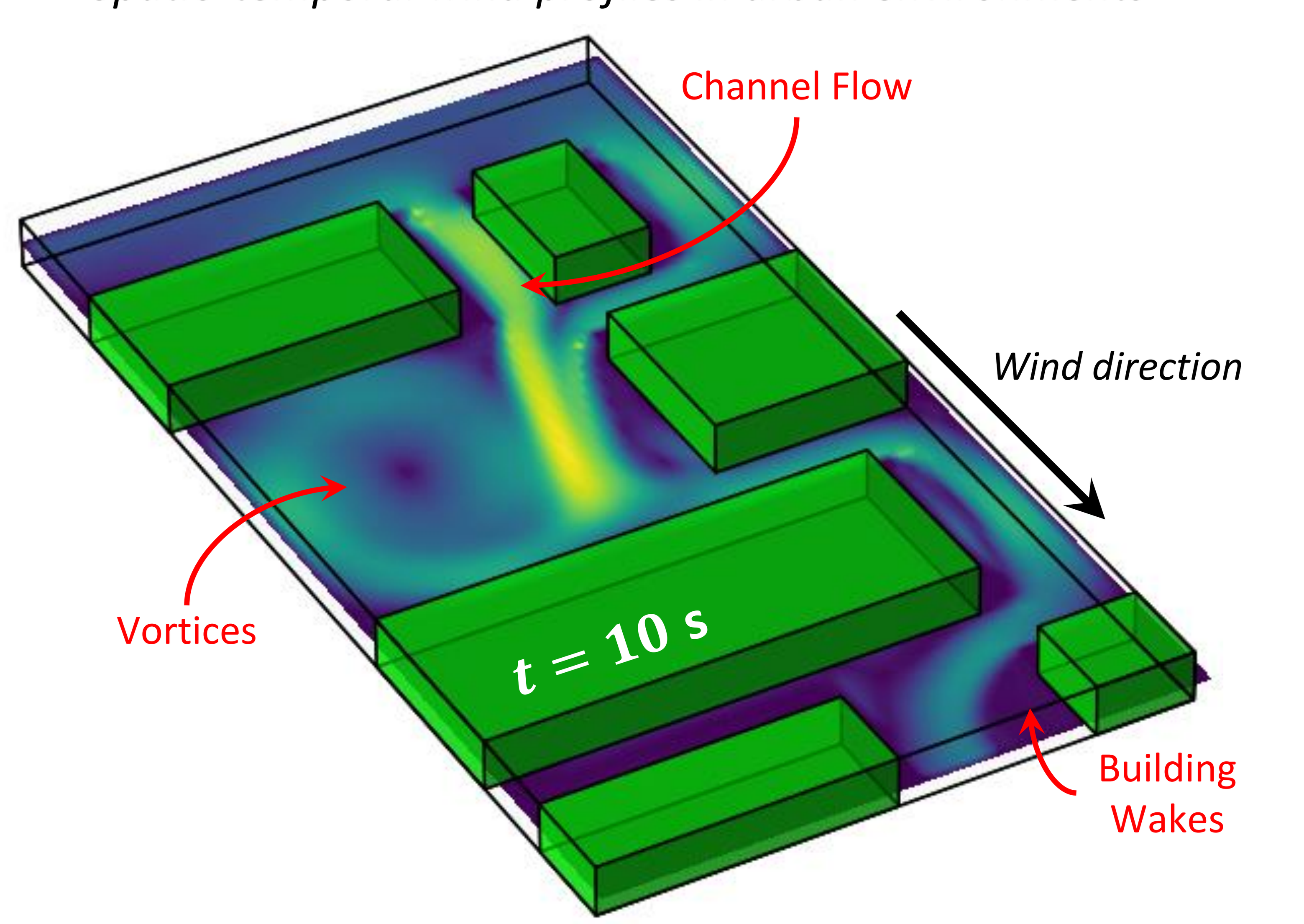
Wind Modeling

- *RotorPy* supports **spatially and temporally varying wind fields.**

Temporal wind profiles



Spatio-temporal wind profiles in urban environments

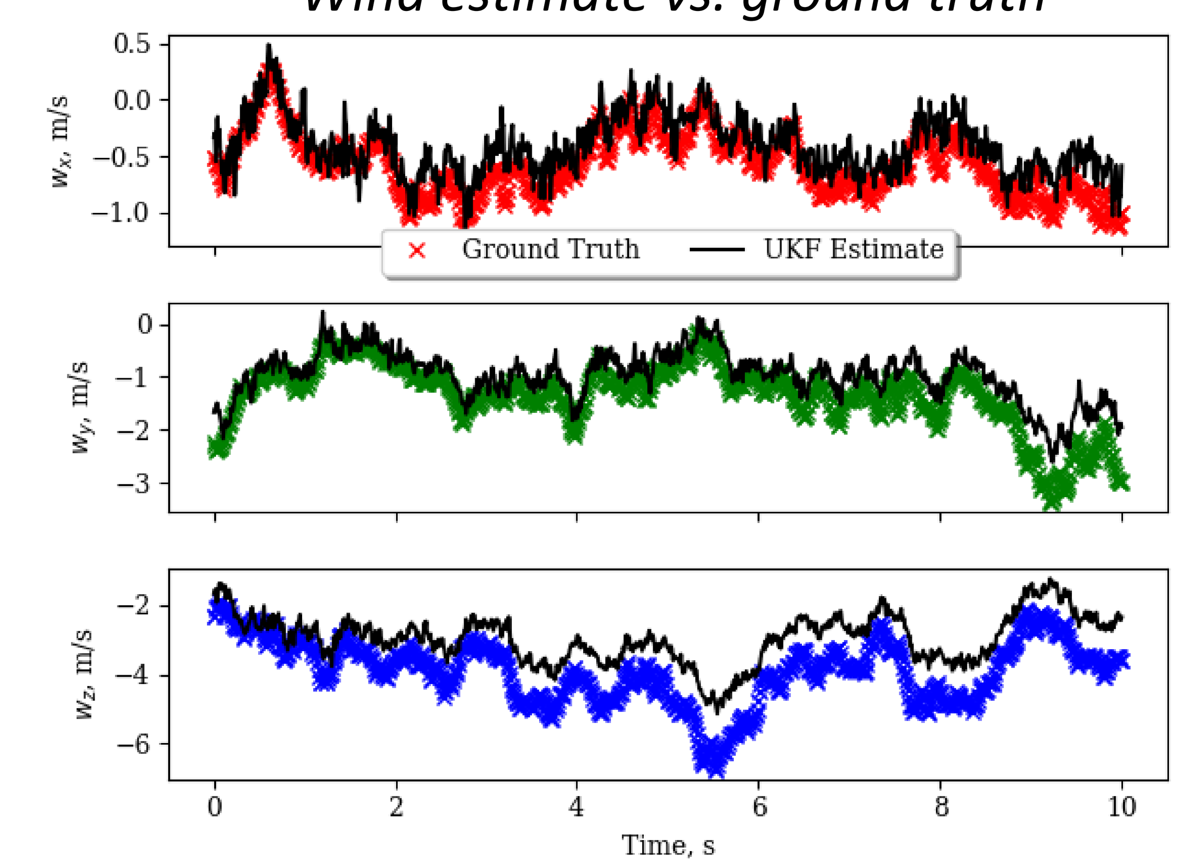


- Generate **unsteady wind** through **urban environments.**
- Use the available wind profiles to **develop and test robust controllers or estimators.**

Proprioceptive Wind Estimation

- As an example, we design and evaluate a **wind estimator** that uses IMU and odometry measurements.

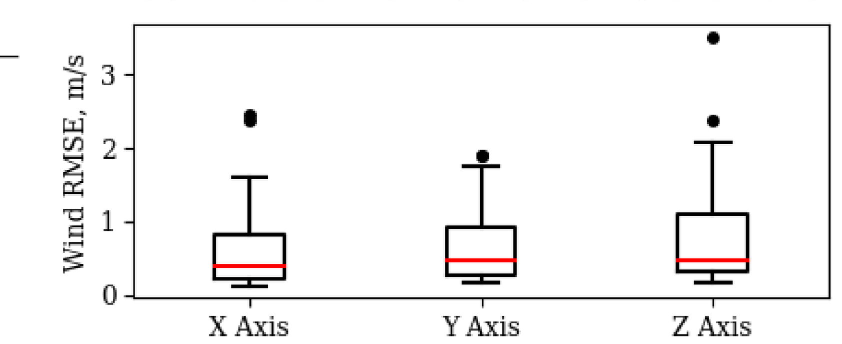
Wind estimate vs. ground truth



Randomized vehicle parameters

Parameter	Unit	Range (min-max)
m	kg	0.375–0.9375
c_{Dx}	N-(m/s)$^{-2}$	0–1(10^{-3})
c_{Dy}	N-(m/s)$^{-2}$	0–1(10^{-3})
c_{Dz}	N-(m/s)$^{-2}$	0–2(10^{-2})
k_d	N-rad-m-s^{-2}	0–1.19(10^{-3})
k_z	N-rad-m-s^{-2}	0–2.32(10^{-3})

Estimation error over 50 trials



Planned Improvements

- Expanding support for **different vehicle morphologies**.
- More generic aerodynamic models using **Blade Element Theory**.
- **Native integration** of numerical fluid solvers with the existing obstacle map representation for **integrated spatio-temporal winds**.
- **Custom Gymnasium environments** for reinforcement learning.

CHECK OUT OUR REPO!

ICRA LONDON 2023