Technical Review for WES-2025-73

Summary:

This paper addresses the limitations of current airborne wind energy (AWE) simulations, which often rely on simplified aerodynamic models that cannot capture unsteady effects, flow separation, or the interaction between control surfaces and aircraft dynamics. To overcome this, the authors develop a geometry-resolved CFD framework—called the Virtual Wind Environment (VWE)—and couple it with the AWEbox simulation and control toolbox using an explicit aero-servo approach. They simulate a 1-loop power cycle of the MegAWES aircraft under realistic wind conditions and demonstrate accurate trajectory tracking and power output, achieving 96% of the reference performance. The study reveals aerodynamic effects missed by low-fidelity models and offers a high-fidelity tool for improving AWE design and control.

nomenclature:

It is better to add this section before the introduction to ease tracking the paper content.

Introduction:

Make this section right after adding a nomenclature section.

In the Introduction, the authors point out that most AWE simulations use simple aerodynamic models that miss unsteady flows and how moving control surfaces really work together. They give a fair overview of past studies, but don't show how big the errors usually are or explain in detail why earlier coupled methods fall short. They make a good case for using full-geometry CFD to see things like flow separation, but don't warn readers that this requires a lot more computing power. The term "Virtual Wind Environment" is a helpful name, but it isn't defined until after its first mentioned.

CFD

The authors use a linear aerodynamic model (AAM) for a fixed-wing aircraft, where the key aerodynamic effects are captured with matrix-based derivatives. They calculate those derivatives by running CFD simulations on an overset (Chimera) mesh, which is great for handling complex shapes and moving parts without having to remake the grid each time.

Still, the paper misses a few practical details. First, there's no information on how much computer time or memory these CFD runs need when you add more moving surfaces. Second, the authors never check their results against a known aircraft or wind-tunnel data, which would help prove their model really works under different flight conditions. Third, they don't show a mesh-independence study—refining the grid until the results stop changing—to confirm their CFD setup is reliable. Because their model is linear, it may lose accuracy when control surfaces move a lot or during strong unsteady effects; discussing where that approach breaks down would round out the paper. The methods are solid, but adding notes on run-time, a basic validation case, a mesh-sensitivity check, and the limits of linearization would make the work much stronger.

AWE system dynamics and control

The presented work is clear that describes the six-degree-of-freedom dynamics, tether modeling, and MPC framework. To help readers fully understand and trust your approach, could you please elaborate on a few points? For instance, a brief explanation of how you selected the Baumgarte stabilization parameter and the MPC horizon length would be very insightful, as would a discussion of the controller's sensitivity to errors in the analytical aerodynamic model—have you tested how deviations in those coefficients affect tracking performance? It would also be helpful to know the computational effort required to solve the MPC every 5 ms (for example, whether it runs in real time on standard hardware or needs a high-performance cluster). Additionally, could you comment on the straight-tether assumption under strong crosswinds or slack conditions and whether you plan to extend the model to include tether sag or bending? Since the tether's mass grows continuously as it reels out, some discussion of how that changing mass and inertia influence the dynamics and controller performance over a full power cycle would greatly clarify the robustness and applicability of your framework.

Aero-servo coupling

The aero-servo coupling you've implemented is very elegant in how it links the high-fidelity CFD solver with the AWEbox dynamics and MPC in a single explicit time-stepping loop, and using CoCoNuT to shuttle motion and force data keeps the workflow organized. To help readers appreciate and trust this approach, could you expand on a few aspects? For example, you mention that no coupling iterations are performed within each 5 ms timestep—have you observed any drift or stability issues over longer simulations, and how did you decide that single-step coupling was sufficient? It would also be useful to know the additional runtime cost and data-transfer overhead introduced by the CoCoNuT interface, especially when moving multiple overset zones each step. In the transition period where you blend AAM and VWE forces, what guided your choice of n_1 and n_2, and how sensitive are your results to that weighting schedule? Could you briefly describe any tests you ran to verify that the explicit coupling preserves energy consistency or avoids unphysical oscillations? A few sentences on these points would greatly strengthen confidence in the robustness and practicality of your aero-servo coupling.

Results

The Results section presents a clear demonstration of your coupled framework, showing both the trajectory tracking—where you achieve a close match within 4 m of the reference path—and the power output, with 96 % of the target average power captured. The side-by-side plots of aerodynamic forces and moments from the VWE and the analytical model effectively highlight model agreement and key discrepancies. To deepen the reader's understanding, it would be helpful to include quantitative error metrics (e.g., root-mean-square deviation over the cycle) for trajectory, power, and force comparisons