S.M.A.C.

(Smart Models for Automated Control)

Wilke Grosche
Benoît Gherardi
Zhen Wei
Project PI: Pascal Fua

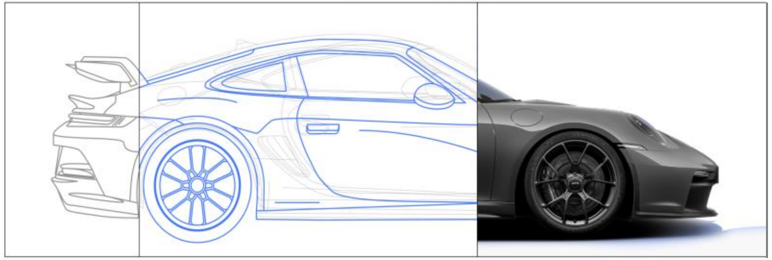
Swiss JRC
Workshop

CVLab

Computer Assisted Engineering

Machines need to be designed for:

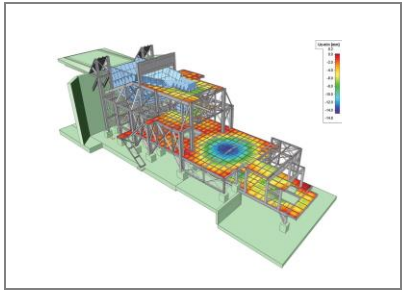
- Performance
- Durability
- Ease of control
- …..



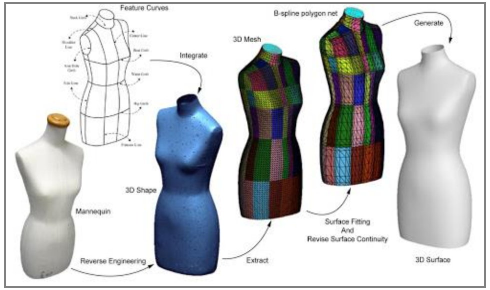
Automotive

Space

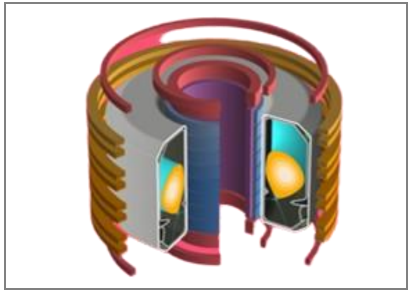
Electrification



Architecture

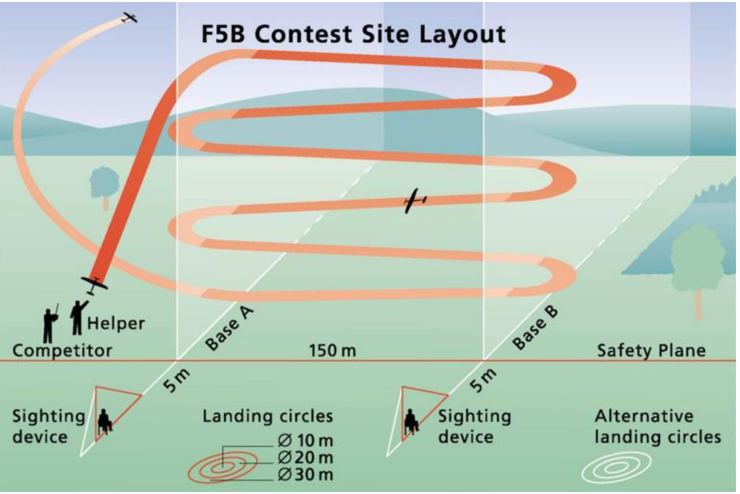


Textile



Nuclear fusion

Long-Term Objective



- F5B Electric Glider Competition

Designing a glider to:
- perform a complex task
- be easy to control
- be structurally sound

Advantages

- Conventional:
  - 1 Operating Point
  - 1 Performance Metric
  - Computationally Expensive
  - Labour Intensive
  - Not Reusable

>6-7h

- Our Approach:
  - Whole Trajectory
  - Adaptable Performance Metric
  - Training is Expensive
  - Inference is ⚡
  - Reusable

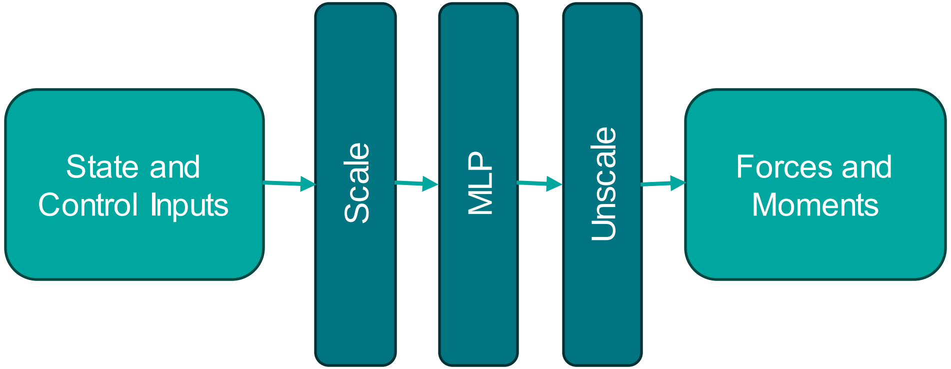
<1s

Road Map

Given an existing glider:

1. Build a dataset and train a Deep Network to predict its dynamics
2. Compute optimal trajectories for its current shape
3. Optimize the shape to further improve the trajectories

Deep Control Laws

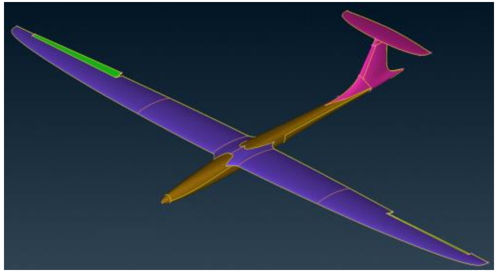


- Two supervision sources to train this model:
  - Simulation data to predict the response of the aircraft to given control inputs
  - Wind tunnel data for greater accuracy

⚠️ There might be inconsistencies between the two

Creating the Simulation Data

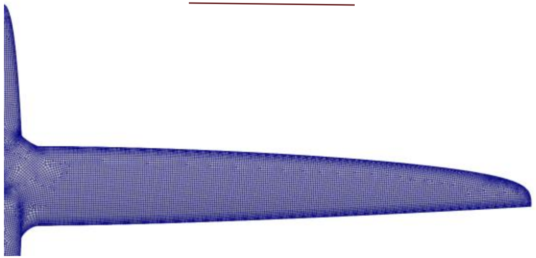
CAD

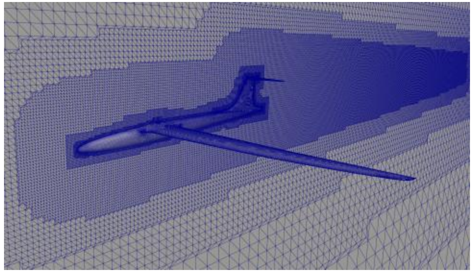


Rhino / Ansa ~1-3h

- Different configurations
- Surface identifications and cleaning
- General preprocessing for meshing

MESH

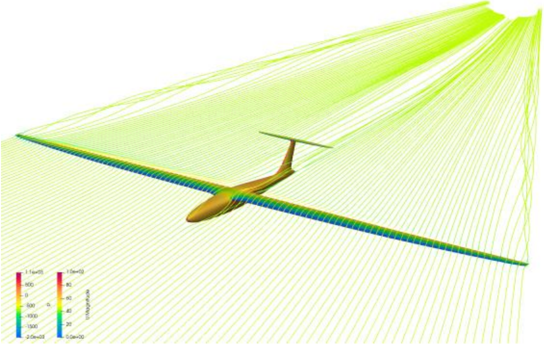




Ansa ~1-4h

- Automated batch mesh workflow
- Anisotropic mesh on L.E & T.E.
- Wake refinement

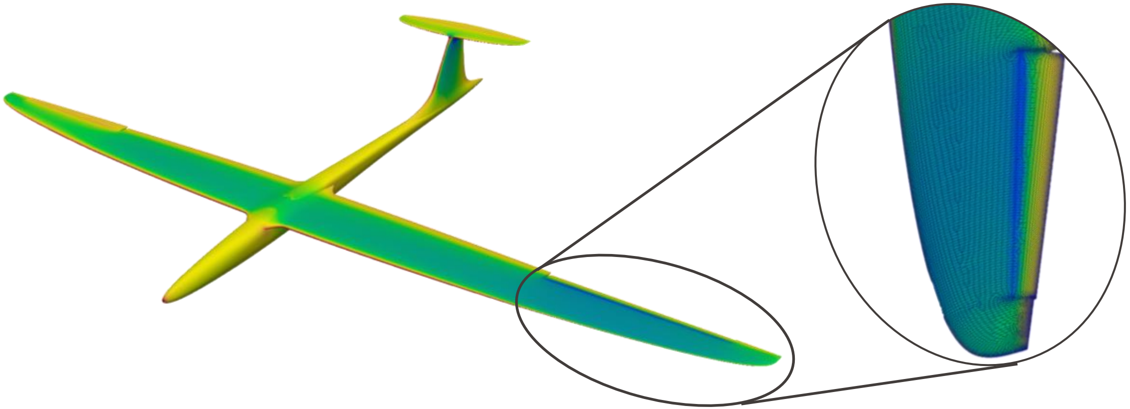
SOLVER

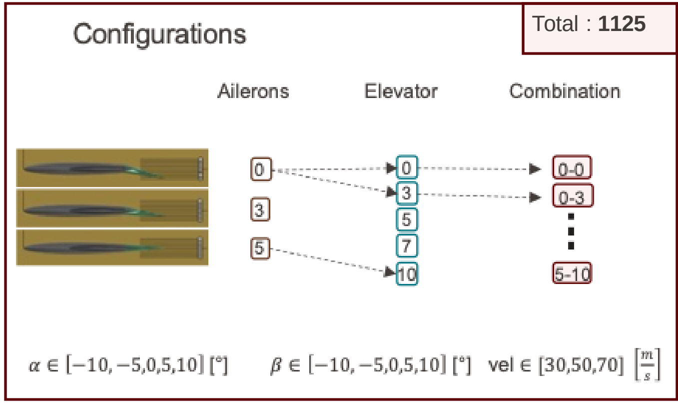


OpenFOAM >5h

- RANS, Spallart-Allmaras
- Automated B.C. changes
- Output forces, moments, pressure

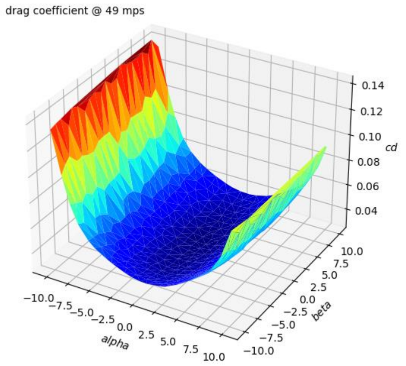
Deflecting the Controls



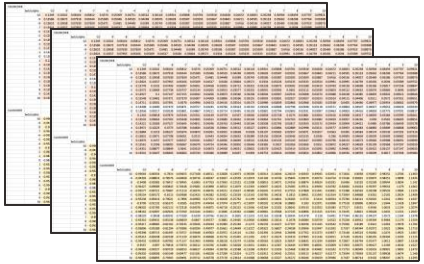


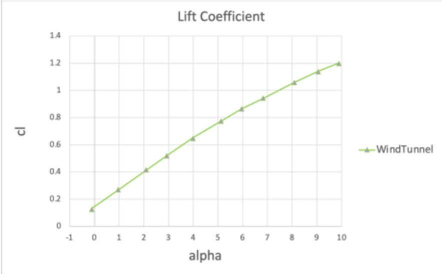
Number of CPU core per simulation: 32 / Average run time: ~5h30

Wind Tunnel Data



University of Washington Aeronautical Laboratory





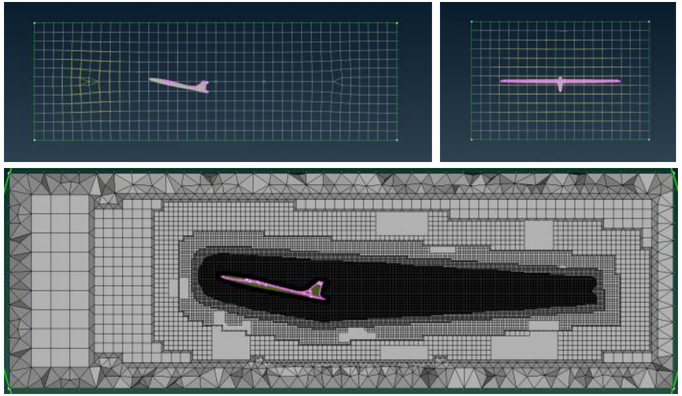
SJRC Workshop

Merging Simulation and Wind Tunnel Data

Windtunnel

Mesh check

Data merging



Freestream

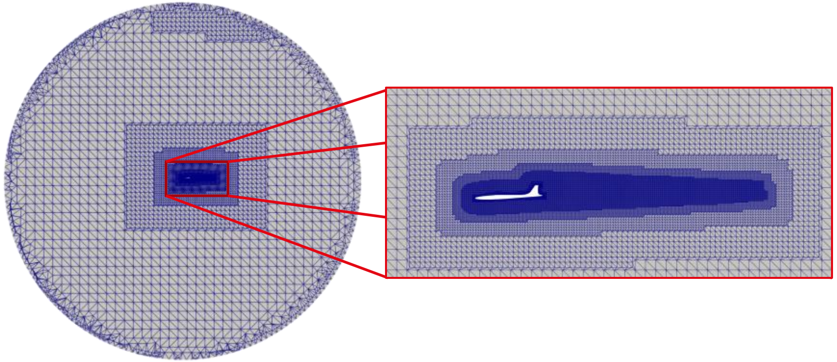
Algebraic correction

blockage effect

Experimental data
Freestream

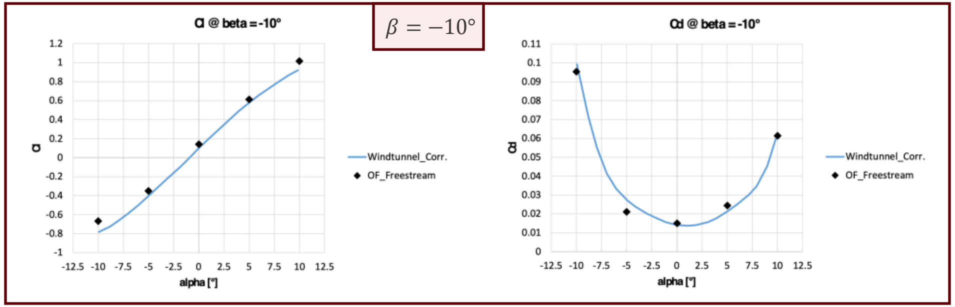
Mesh check

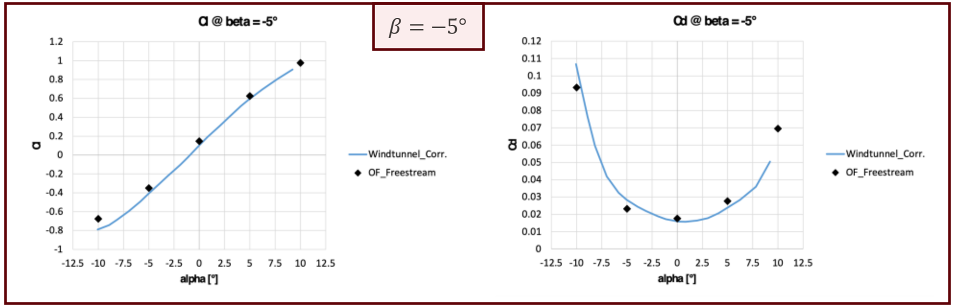
Data merging

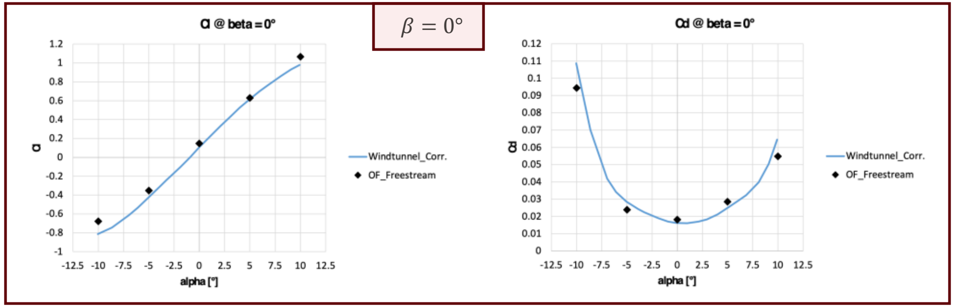


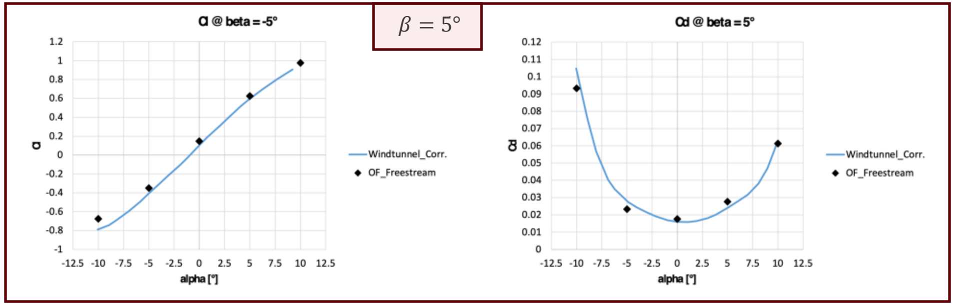
Freestream Results

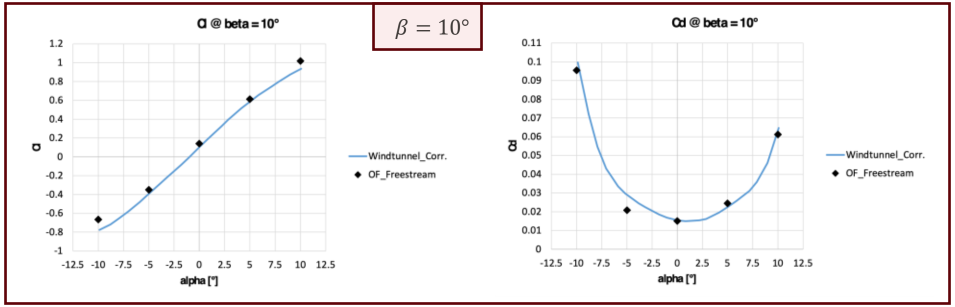
Experimental / OpenFOAM











Training the Networks

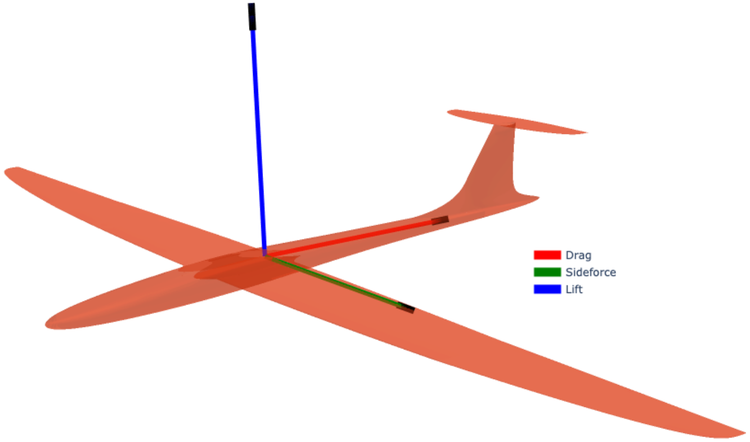
Virtual Data

CFD + other aerodynamic modeling techniques

Experimental Data

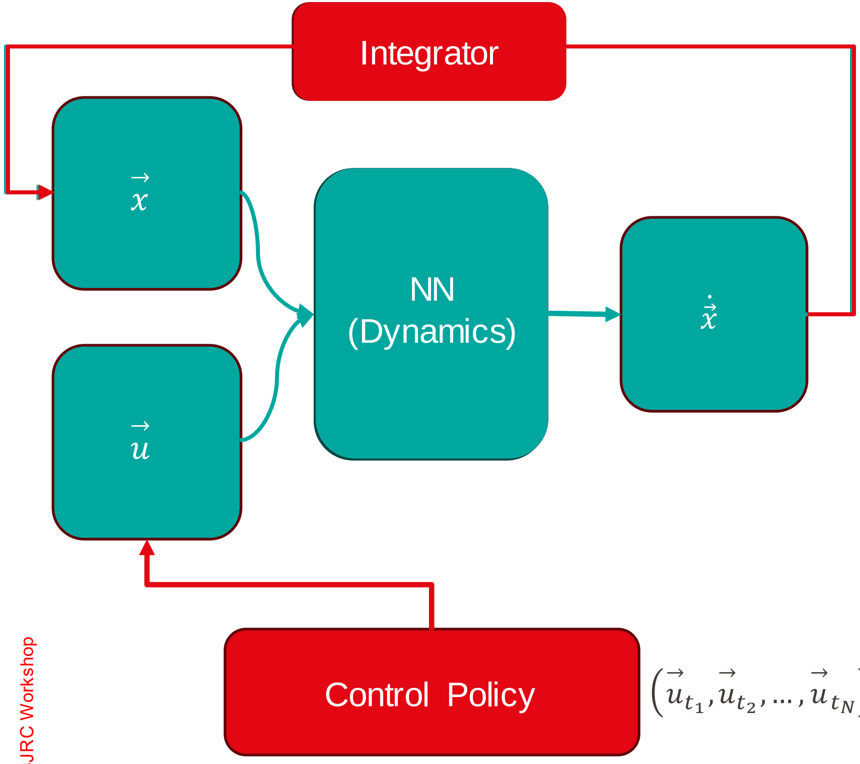
Wind tunnel tests

NN (Dynamics)

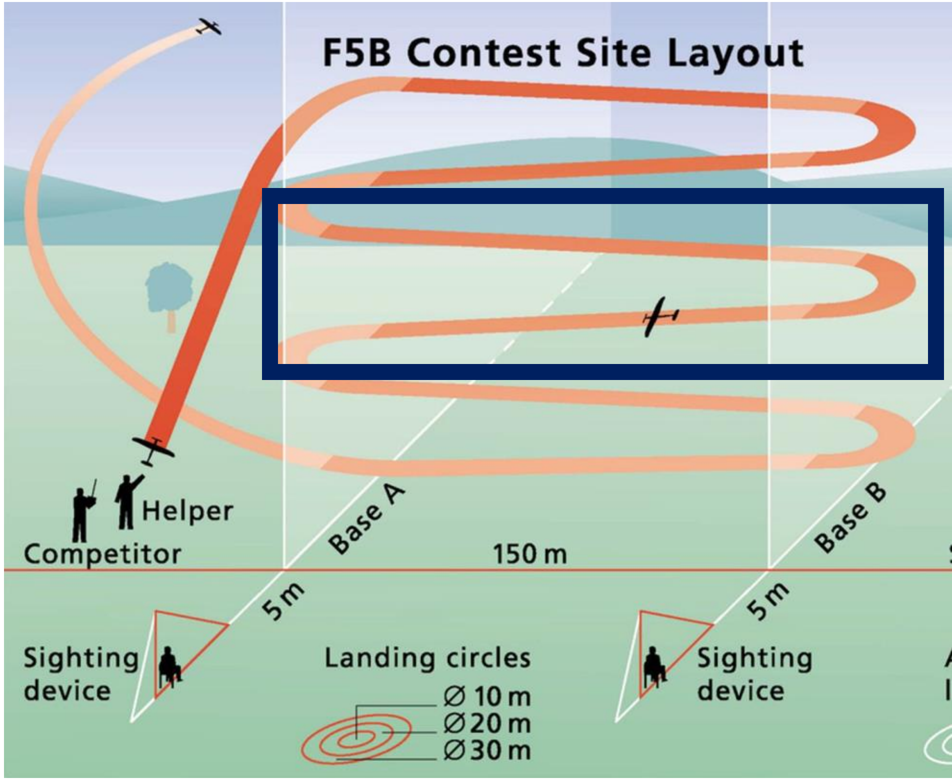


Dynamics Simulation

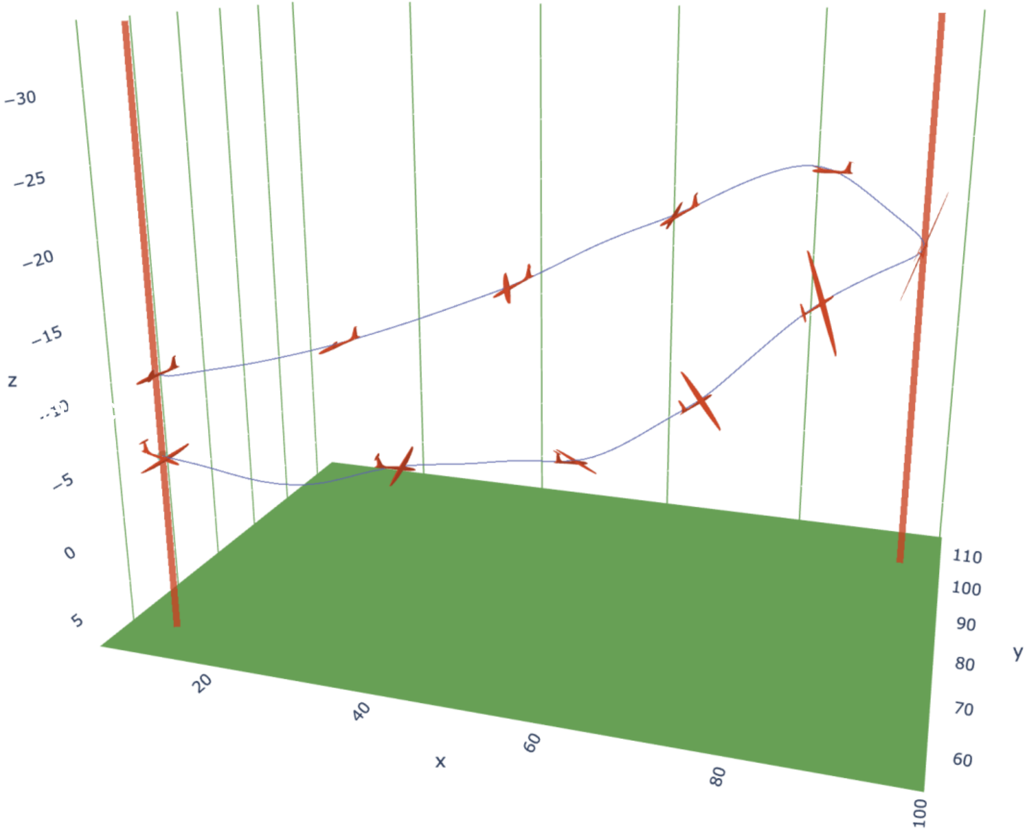
$$\vec{x}_{t+1} = \vec{x}_t + \dot{\vec{x}}\delta t$$



$$\left(\vec{u}_{t_1}, \vec{u}_{t_2}, \dots, \vec{u}_{t_N}\right) = argmin_{(\vec{u}_{\dots})} t_{final}$$

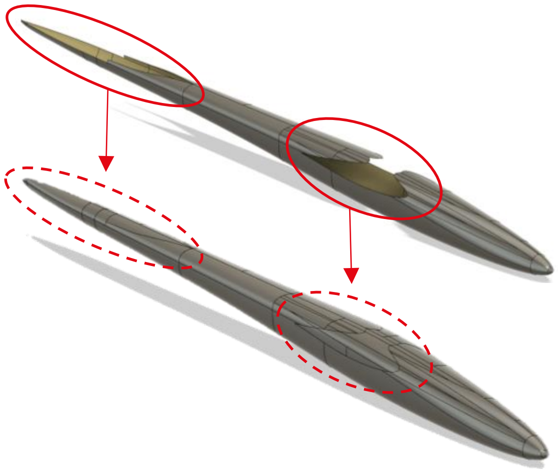


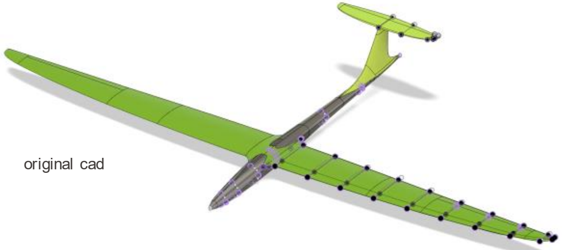
Dynamics Simulation

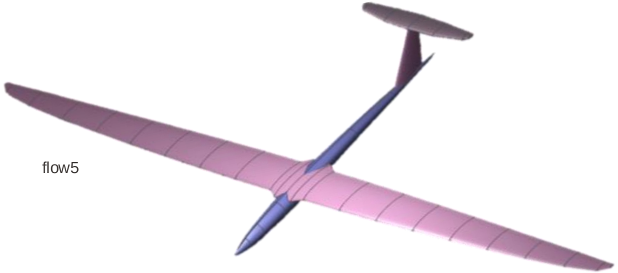


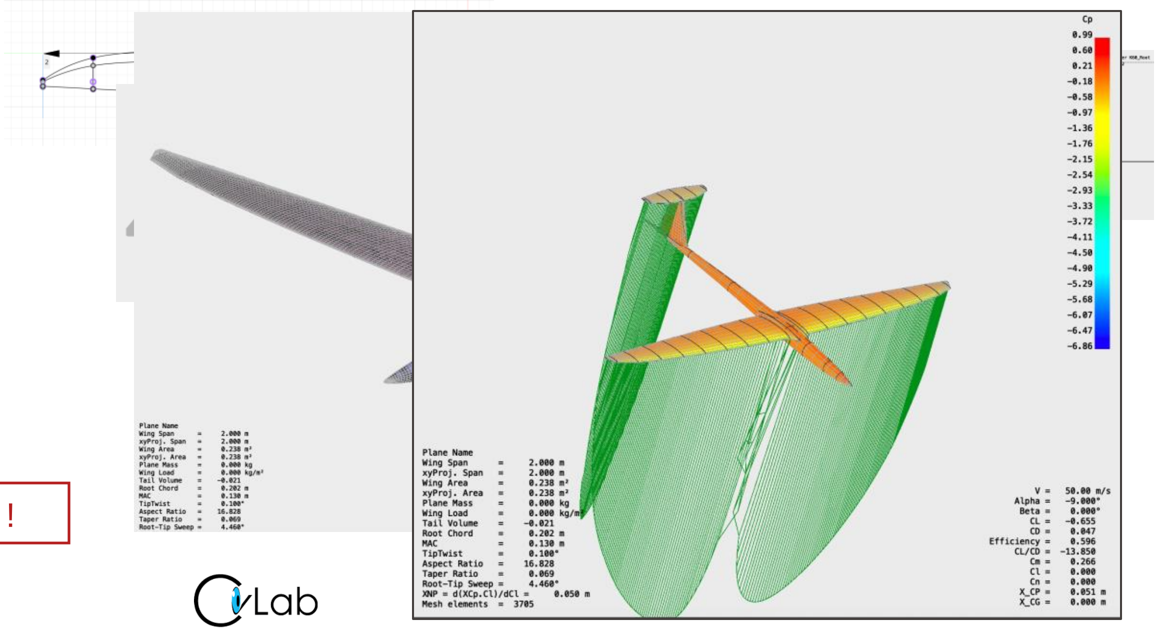
Towards Deformation

Glider reverse engineering







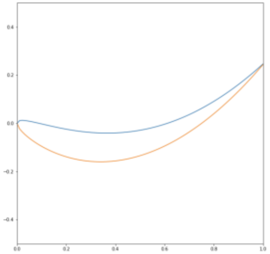


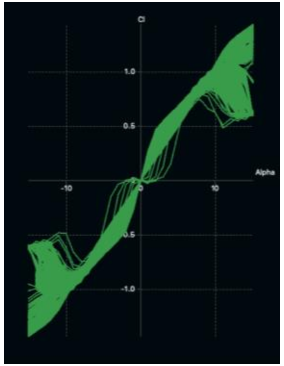
Run time for ~200 points = 5min !

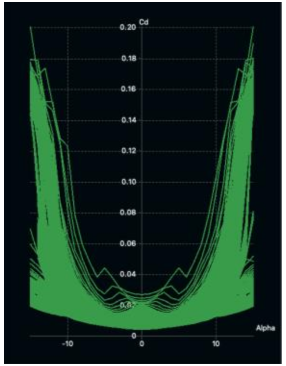
3D Wing Deformation

Flow5

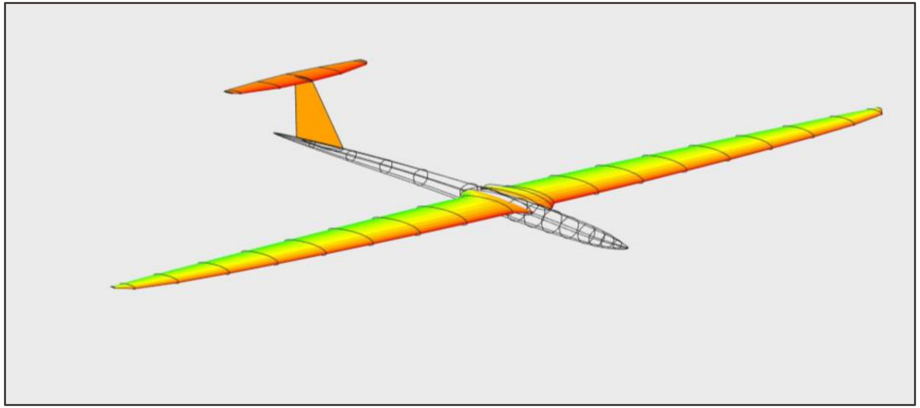
NACA airfoil (Python)

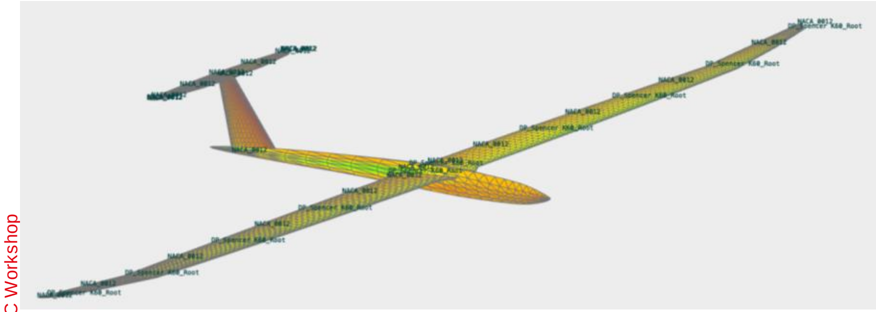


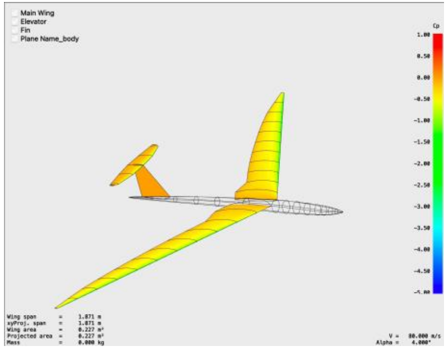


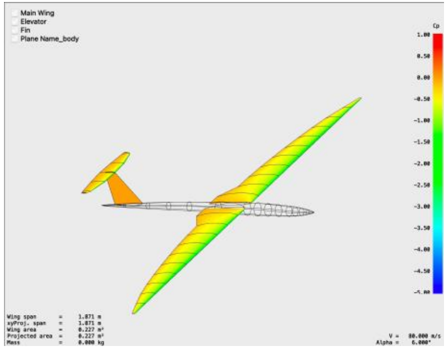


3D deformation



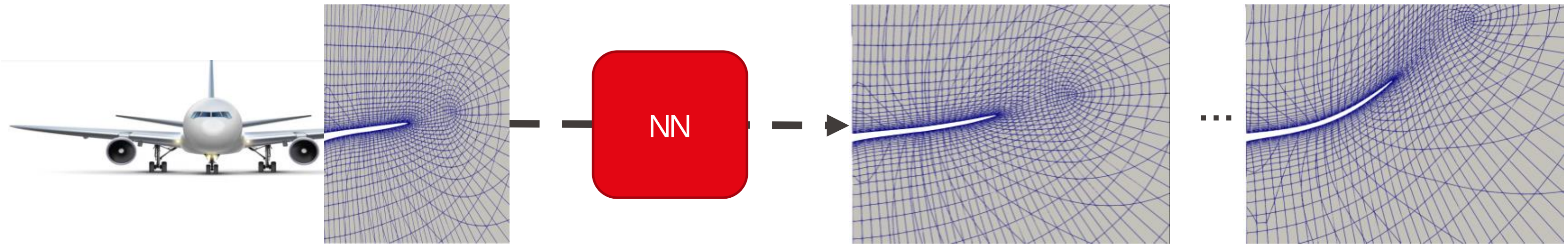






SJRC Workshop

The Road Forward



We are building a deep geometric learning model to:

- manipulate the aircraft's geometry
- deform the computational meshes accordingly for simulation purposes

Questions?