QUADROTOR CYCLOGYRO AIRCRAFT IN FORWARD FLIGHT CFD MODEL

LOUIS GAGNON[1], GIUSEPPE QUARANTA[2], MEINHARD SCHWAIGER[3], DAVID WILLS[4]
[1]*Department of Aerospace Engineering, Politecnico di Milano, Italy, louis.gagnon@polimi.it*
[2]*Department of Aerospace Engineering, Politecnico di Milano, Italy, giuseppe.quaranta@polimi.it*
[3]*Innovative Aeronautics Technologies GmbH, Austria, meinhard.schwaiger@iat21.at*
[4]*Innovative Aeronautics Technologies GmbH, Austria, david.wills@iat21.at*

Keywords: 3D CFD, AMI/GGI interfaces, Cycloidal rotors, Experimental validation, RANS-Euler, Parallel computing

A 3D CFD model for the aerodynamic analysis of a quadrotor cyclogyro was built. This paper covers its creation and its validation. The model matched very well with experiment for vertical forces and rather poorly for longitudinal forces. The validation was extended to a series of flight velocities for which the trends of the experimental and numerical analyses agreed. Comparison of wind-tunnel video footage and CFD field snapshots validated the shape of the flow. The model used the Euler equations and showed a nearly linear speedup on up to 4 CPUs.

1 Background

The concept of the cycloidal rotor is still new to the scientific community. As opposed to conventional propellers, cycloidal rotors can produce forces in a 360° plane and change the thrust direction almost instantly. They consist of an arrangement of constant section and symmetric blades that rotate about a central drum. That drum transmits the rotation motion to the blades through a series of pivot rods. The blade pitching motion is transmitted through a series of pitch rods which are themselves offset by a central mechanism within the drum. The blades pitch individually about the intersection point between the blade and the pivot rod. A rotor is illustrated in Fig. 1a. The total thrust the rotor creates comes from the individual blade lift and drag forces. The D-Dalus is an aircraft prototype developed by inventor Meinhard Schwaiger of IAT21 [1, 2]. It relies solely on cycloidal rotors for thrust generation. The actual aircraft prototype is shown in Fig. 1c.

(a) Cycloidal rotor concept. (b) D-Dalus render. (c) Real D-Dalus.

Figure 1: Rotor and vehicle illustrations.

The CFD model was developed in order to gain a deeper understanding of the aerodynamic interaction between the front and rear rotors of the D-Dalus aircraft. It allows to make better informed choices for future aircraft geometries and rotor configurations. Finally, a better understanding of rotor-aircraft-rotor flow interaction arises from this project.

2 CFD Model

The model is a tridimensional CFD Euler laminar simulation. It uses the *pimpleDyMFoam* solver of the OpenFOAM 2.4.x toolkit. The Euler simulation is achieved by setting the viscosity to zero. A total of 14 moving meshes, which use the Arbitrary Mesh Interface (AMI) algorithm, are within the model. Each rotor blade is inserted into a double embed oscillating AMI. The double embed moving mesh algorithm [3] and an accompanying moving slip boundary condition [4] were previously created [5] and publicly released. A single rotor mesh uses roughly 1 million cells, of which the two thirds are due to mesh refinement near the endplates. The endplates have the particularity of making the flow more two-dimensional and thus qualitatively closer to the experimental results. The CFD models the aircraft with a symmetry about the x-y plane. The simulations are ran using the first harmonic sinusoidal pitching schedule function described by Eq. (1).

$$\theta = \theta_o + \theta_s \sin(\omega t + \phi) \tag{1}$$

It has negligible differences with the actual aircraft pitching schedule and θ_o is the fixed pitch angle offset, θ_s is the magnitude of the pitch angle variations, ϕ is the imposed phase angle, and ω is the rotor angular velocity.

The aircraft is fixed in space and thus the model disregards the inertial effects of gravity and aerodynamic forces. It was possible to maintain the real-world geometry of the aircraft by carefully tuning the mesh interfaces. The space available between the rotor blades pivot points and the airframe allowed to have an AMI cylinder radius at least equal to the maximum distance between pivot point and edge of the blade. This allows for any pitch angle.

Diverse strategies were considered for the parallelization of the domain which contains AMI zones. The strategy which yielded the best results was to decompose with the *simple* method. Doing so, the communication across processors for the AMI was reduced to a minimum. Figure 2a shows the 4 processor submeshes obtained with the *simple* algorithm and the refinement near the AMIs. The mesh on the airframe and rotors is shown in Fig. 2b along with the symmetry plane.

(a) 4 processors shown by 4 colors.

(b) Airframe mesh and symmetry plane.

Figure 2: Decomposition and mesh illustrations.

3 CFD Calibration and Validation

The values of drag and lift obtained by CFD were compared to the wind tunnel experimental data. At first, the highest wind tunnel velocity was chosen as a basis for validation. The major challenge was to find the appropriate angular velocity for the unpowered rear rotors which were used during the wind tunnel tests. For these rotors, there was unfortunately no information on their angular velocity. Correspondence between wind tunnel flow visualization and CFD streamlines was observed. The trends of CFD power and thrust matched those of the wind tunnel at different flight velocities. Velocities of 10 m/s, 15 m/s, 20 m/s, and 25 m/s were considered, as shown in Fig. 3.

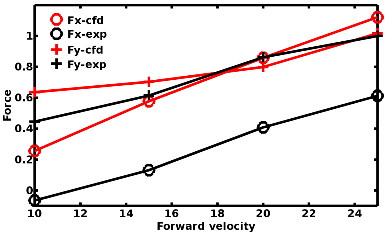

Figure 3: Trend match between CFD (red) and wind tunnel (black).

References

[1] M. Schwaiger, "Aircraft: Us patent 7735773 b2," http://www.freepatentsonline.com/7735773.html, 2010, last accessed Feb. 2016.

[2] ——, "Aeroplane: Us patent usd709430 s1," http://patents.google.com/patent/USD709430S1, 2014, last accessed Feb. 2016.

[3] Gagnon, L., "Interface within an interface c++ code," http://www.cfd-online.com/Forums/openfoam-solving/124586-dynamic-mesh-within-dynamic-mesh.html#post476869, 2014, last accessed Feb. 2016.

[4] ——, "Slip moving wall boundary condition c++ code," http://www.cfd-online.com/Forums/openfoam-solving/105274-free-slip-moving-wall-bc.html#post509989, 2014, last accessed Feb. 2016.

[5] L. Gagnon, M. Morandini, G. Quaranta, V. Muscarello, G. Bindolino, and P. Masarati, "Cyclogyro Thrust Vectoring for Anti-Torque and Control of Helicopters," in *AHS 70th Annual Forum*, Montréal, Canada, May 20-22 2014.