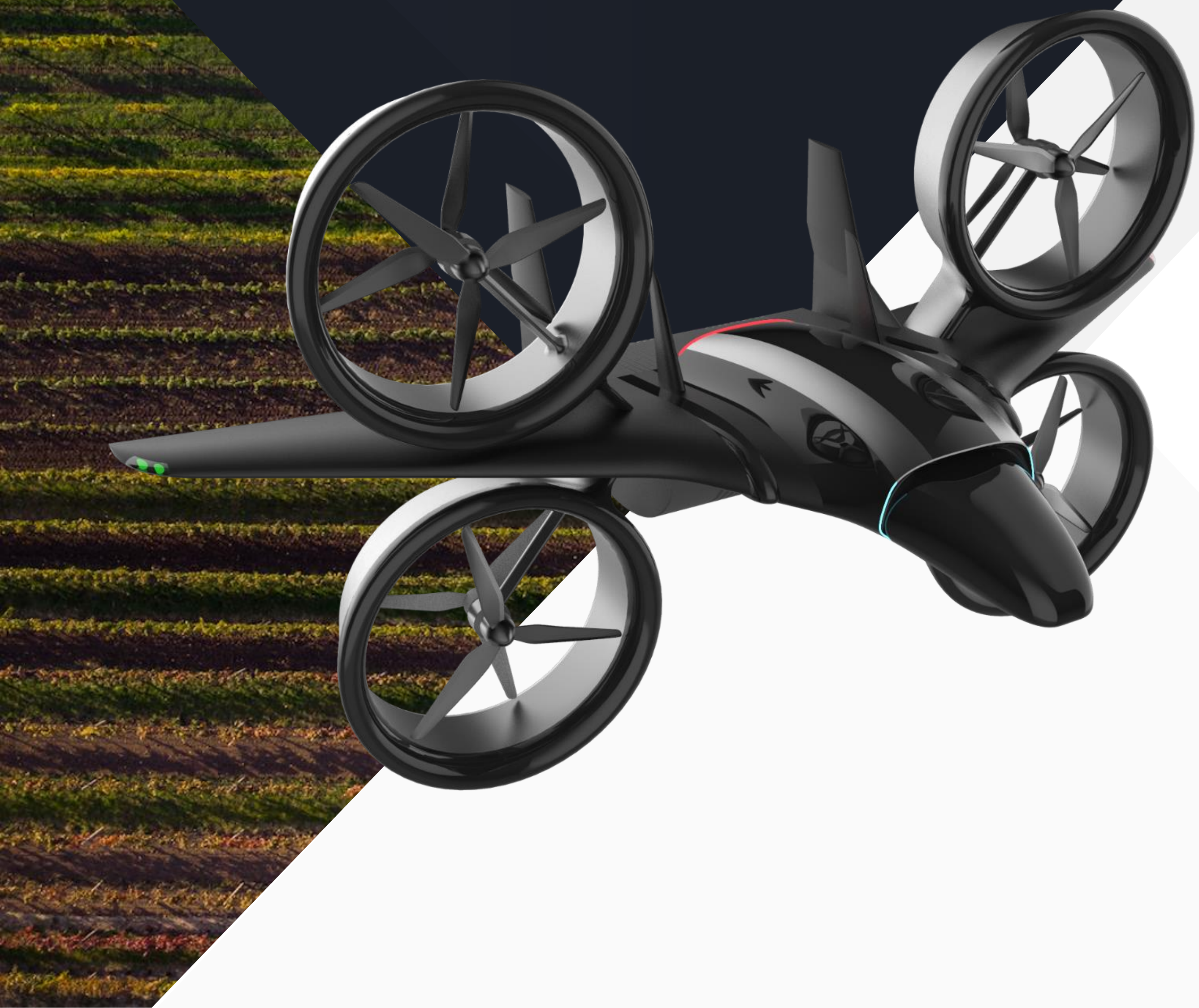


Design of a Ducted Propeller System for a Drone using OpenFOAM

Lifting heavier, flying faster, reaching further

NINANO COMPANY

Powered by NINANO COMPANY INC.

Introducing

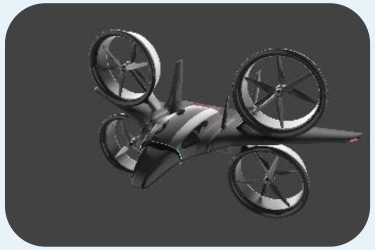
“ 테일시터형 무인비행체 시스템 ”

수직이착륙과 고속 수평 전진비행이 가능하여
더 빠르게, 더 멀리 비행할 수 있습니다.

구조가 단순한 만큼 **더 많은 화물이나 장비**를
싣고 비행할 수 있습니다.

공기역학적 설계로 바람에 대한 적응력이 높습니
다.

Why Tail-sitter?

Items	Multicopter	Tilt rotor	Hybrid VTOL (Lift & Cruise)	Tail-sitter
				
Features	• Controls altitude using multiple propellers • Most widely used	• Transits from vertical to horizontal by tilting the prop direction	• Fixed wing type • Separated thruster for vertical and horizontal manoeuvers	• “Sitting” on a ground using its “tail” • Hybrid of Multi-copter & Fixed wing
Cons	• Easy to control • Multipurpose	• Fast forward flight • Long range & endurance	• Relatively easy to control • Long range & endurance	• Easy to control • Long range & endurance • Simple structure
Pros	• Short range & endurance • Low speed • Low payload	• Complex structure for tilting thruster • Hard to control	• Low payload • Low aerodynamic efficiency	• Less stable in vertical flight mode • Transition flight • Loading cargo

Tail-sitter Products

FALCONET (NDV-P01)

- Small-size Tail-sitter
- ISR OP/Light Armed

PAYLOAD	1kg
CRUISE SPEED	60km/h
RANGE	>40km
ENDURANCE	>1Hr
REMARKS	Launching in Oct '22

KESTREL (NDV-P05)

- Small-size Tail-sitter
- ISR OP/Delivery

PAYLOAD	5kg
CRUISE SPEED	80km/h
RANGE	>60km
ENDURANCE	>2Hr
REMARKS	Launching in Oct '22

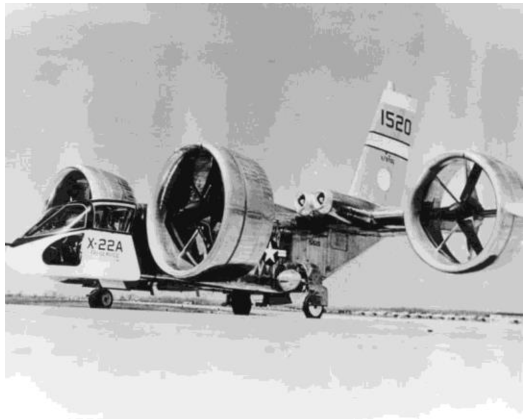
ORCA (NDV-P20)

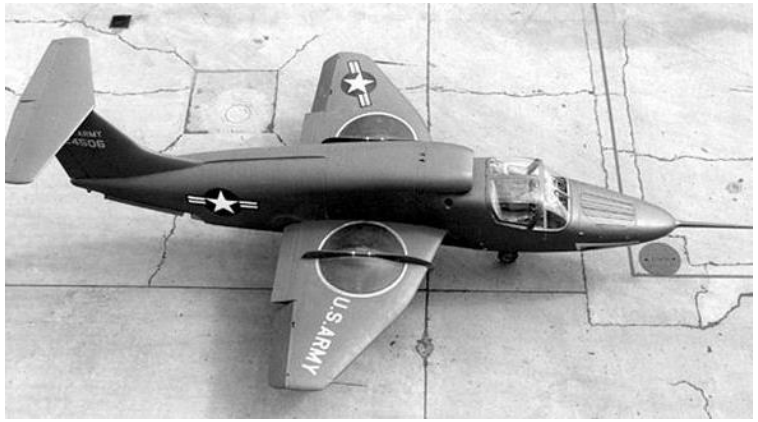
- Mid-size Tail-sitter
- ER ISR OP/Delivery

PAYLOAD	20kg
CRUISE SPEED	108km/h
RANGE	>80km
ENDURANCE	>6Hr
REMARKS	Under Development

NINANO COMPANY

Ducted Propulsion System





ADVANTAGES

- Reducing tip losses - more efficient at low speed
- Smaller diameter for the same static thrust
- Quieter than the open propeller
- Enhanced safety on the ground

DISADVANTAGES

- Less efficient at cruise
- Requires very small gap between blade and duct
- Requires high RPM and minimal vibration
- Complex duct design and weight increase
- Duct stalls at high angle of attack
- Produces aerodynamic drag

NINANO COMPANY

Overall Design Procedure

Design Requirements

- Target thrust
- Maximum thrust
- Specific thrust
- Maximum allowable torque

Max 500N thrust with 18kW

Parameterize

- Propeller chord & beta distributions
- Propeller and duct cross section
- Duct length & leading edge diameter

Parametric Study

Stock Design

- Selection of stock (or mother) propeller & duct
- Reference propeller
- Reference duct

Starting Point of the Design Process

Commercial Propeller (Inverse Design)

Analysis & Selection

- Potential flow analysis
- CFD analysis
- Machine Learning

Low- & High-fidelity Simulations

Automation & Optimization

Iterations

NINANO COMPANY

Numerical Methods

System configurations

Pre-processing

Rhinoceros

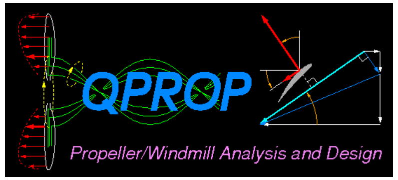
Mesh Generator

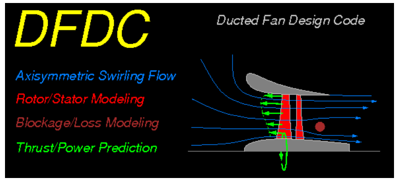
STAR-CCM+

Solver

OpenFOAM®

Low-fidelity Simulation





- Open and ducted propeller analysis tools
- Potential flow theory
- M, Drela & H. Youngren

Machine Learning Framework

Keras

Automation

BASH THE BOURNE-AGAIN SHELL

python

Computing H/W

AMD

rescale

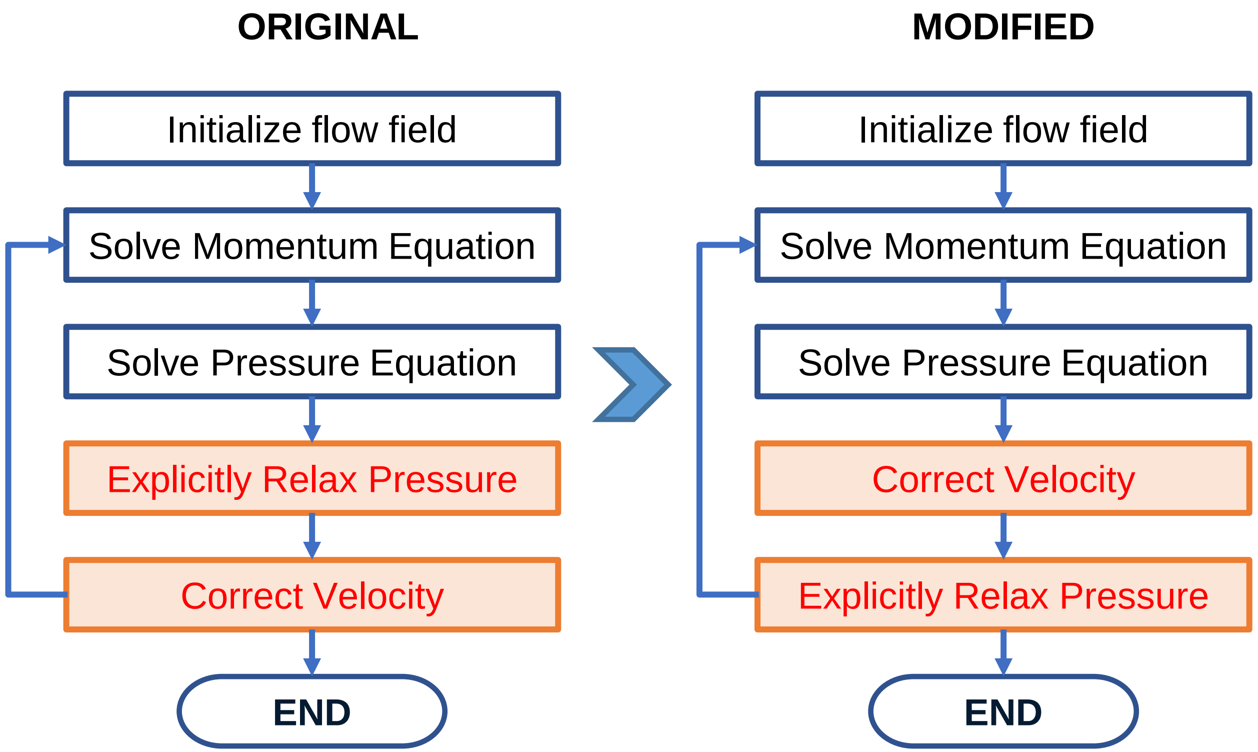
Simulation Methods

- MRF for initialize flow field
- Transient simulation by using rotating body
- Artificial mesh interface

Numerical Methods

Revision on the OpenFOAM – Pressure-velocity Coupling & Matrix Solver

Pressure relaxation
In PIMPLE ALGORITHM of the OpenFOAM



Matrix Diagonal Dominance

`Ueqn.relax();` VS `Ueqn.relax(1.0);`

`Ueqn.relax();`

- Does not relax the matrix at PIMPLE final iteration

`Ueqn.relax(1.0);`

- Relax the matrix by using the relaxation factor 1.0
- Ensure the diagonal dominance

Effective for poor quality grids
(But good grid first!)

Numerical Methods

Revision on the OpenFOAM – TRANSIMPLE (Transient SIMPLE)

PIMPLE algorithm of the OpenFOAM tightly solve the matrix at every corrector steps. This is quite time consuming process.

Solve the governing equations to meet relative tolerance at every inner iterations but the final iteration step: Solver solves the matrix until it meets the absolute tolerance without relaxing the matrix.

```
if (transimple)
{
        p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.finalIter())));
}
else
{
        p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.finalInnerIter())));
}
```

Numerical Methods

Revision on the OpenFOAM – Deferred Correction

Diagonal dominant matrix
→ important for a stable and robust simulation

Deferred correction[1] is an iterative but good way to improve the solver robustness.

$$\frac{\partial u}{\partial t} + (\nabla \cdot uu)_{IM}^{HOS}$$

$$\frac{\partial u}{\partial t} + (\nabla \cdot uu)_{IM}^{LOS} + (\nabla \cdot uu)_{EX}^{HOS} - (\nabla \cdot uu)_{EX}^{LOS}$$

Source terms

In the source code of the momentum equations,

ORG

```
fvVectorMatrix UEqn
(
    fvm::ddt(rho, U)
      // + fvm::div(rhoPhi, U)
  + MRF.DDt(rho, U)
  + turbulence->divDevRhoReff(rho, U)
 ==
    fvOptions(rho, U)
);
```

MOD

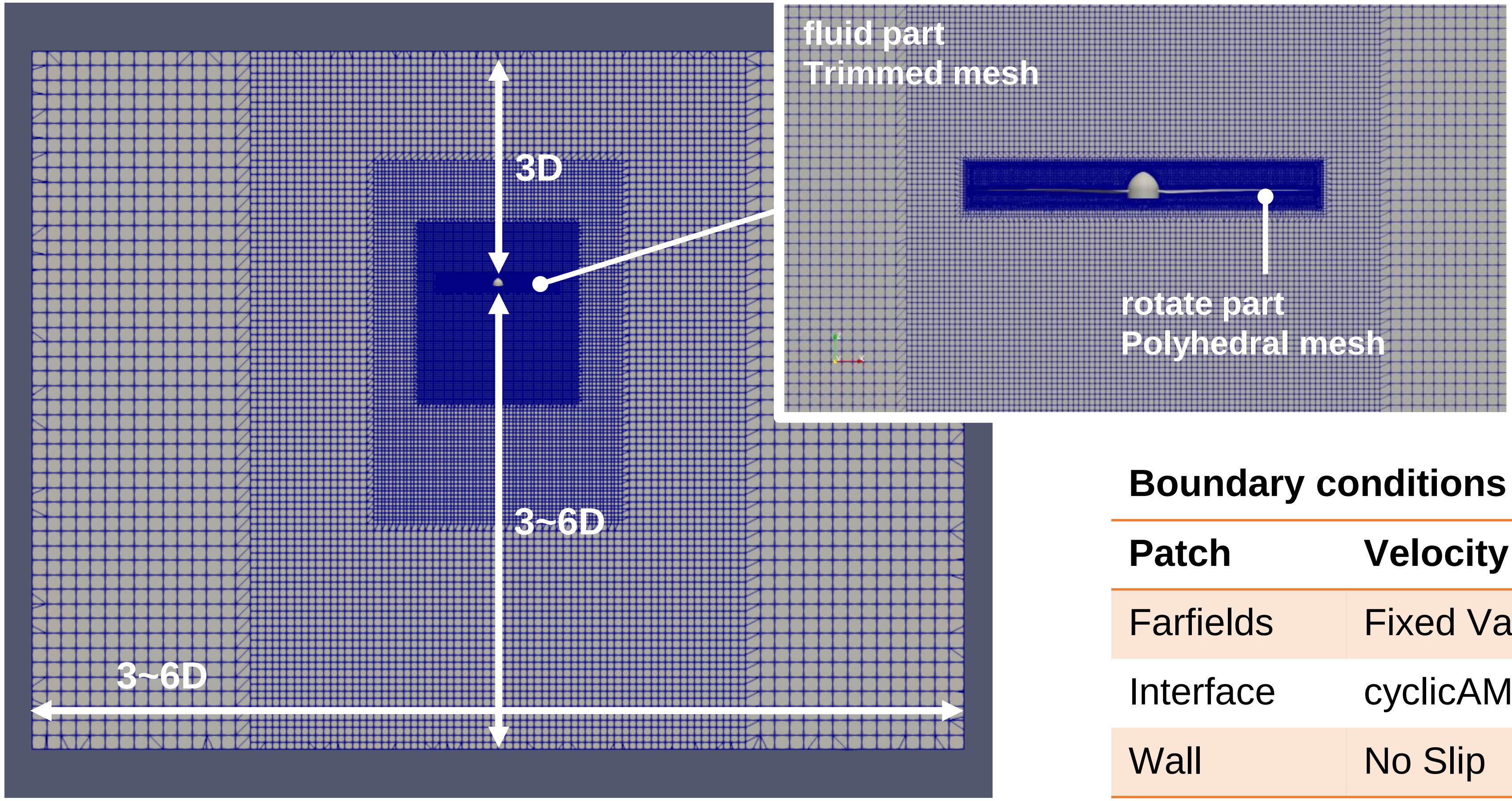
```
//-----------------------------------------------------------------------------
// deferred correction
Switch deferredCorr
(
        pimple.dict().lookupOrDefault<Switch>("deferredCorr", "off")
);

if (deferredCorr)
{
        UEqn +=
        (
                fv::gaussConvectionScheme<vector>
                (
                        mesh,
                        rhoPhi,
                        upwind<vector>(mesh, rhoPhi)
                ).fvmDiv(rhoPhi, U)
          ==
                fv::gaussConvectionScheme<vector>
                (
                        mesh,
                        rhoPhi,
                        upwind<vector>(mesh, rhoPhi)
                ).fvcDiv(rhoPhi, U)
           - fvc::div(rhoPhi,U)
        );
}
else
{
        UEqn += fvm::div(rhoPhi, U);
}
//-----------------------------------------------------------------------------
```

[1] M.Peric and Ferziger, Computational Fluid Dynamics, Springer

Simulation Results

Computational domain



Computational Domain

Tests on the following items were performed using a commercial propeller

- Grid configuration
- Domain size
- Boundary condition

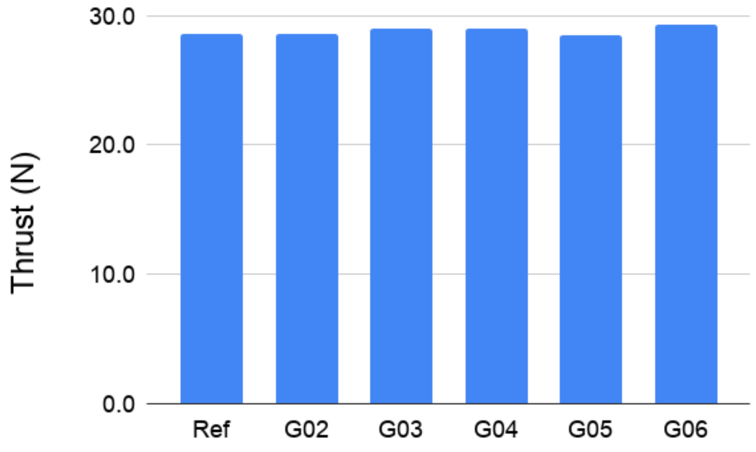
Boundary conditions

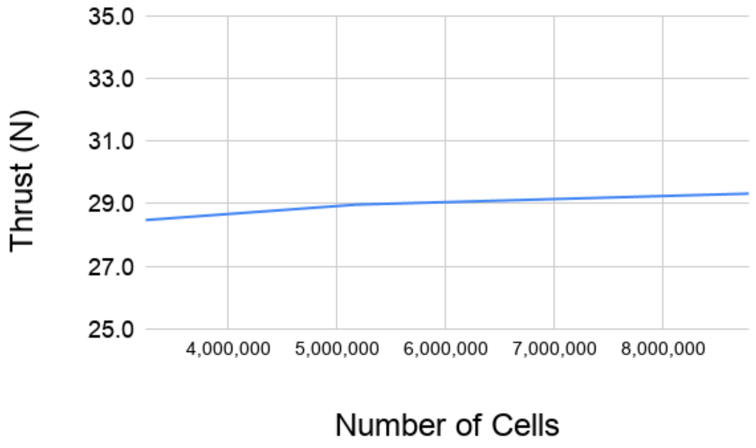
Patch	Velocity	Pressure
Farfields	Fixed Value	Zero Gradient
Interface	cyclicAMI	cyclicAMI
Wall	No Slip	Fixed Flux

Simulation Results

Tests on the grid configuration

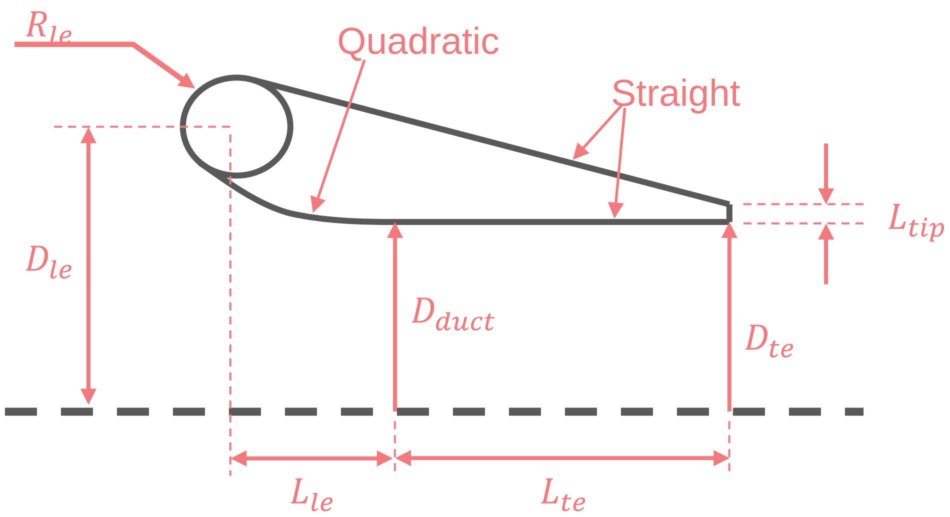
Category	Items	Ref	G02	G03	G04	G05	G06
flow field	base size	0.09375	0.09375	0.09375	0.09375	0.1171875	0.075
	diameter	3	6	6	6	6	6
	inflow Length	3	3	3	3	3	3
	downstream Length	6	6	6	3	3	3
rotate	base size	0.01171875	0.01171875	0.01171875	0.01171875	0.0146484375	0.009375
	n Prism Layer	5	10	10	10	10	10
	Target srf size	15	15	25	25	25	25
	Minimum srf size	2	2	2	2	2	2
nCells		5,226,761	5,150,131	5,729,377	5,180,420	3,240,440	8,792,131
Results	Thrust (N)	28.5657	28.5989	28.9526	28.9703	28.4782	29.3213
	Torque (N-m)	1.2026	1.2024	1.2186	1.2183	1.2121	1.2173

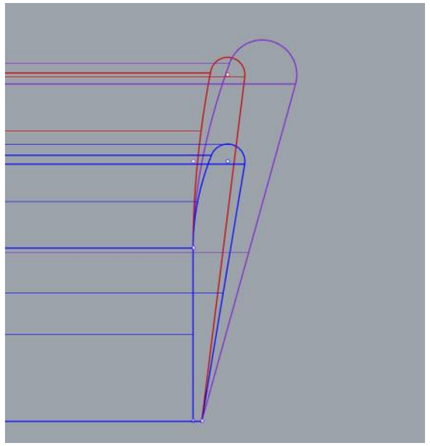


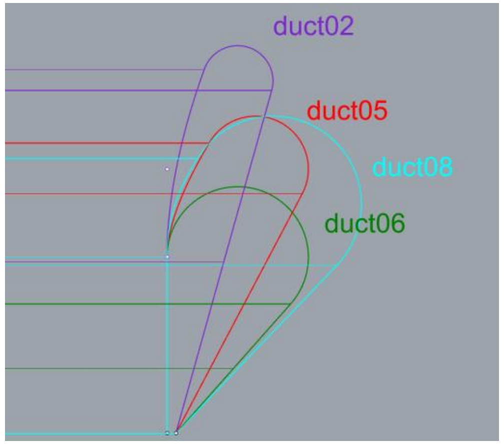


Designing Duct

Parameterizing a duct section & initial design





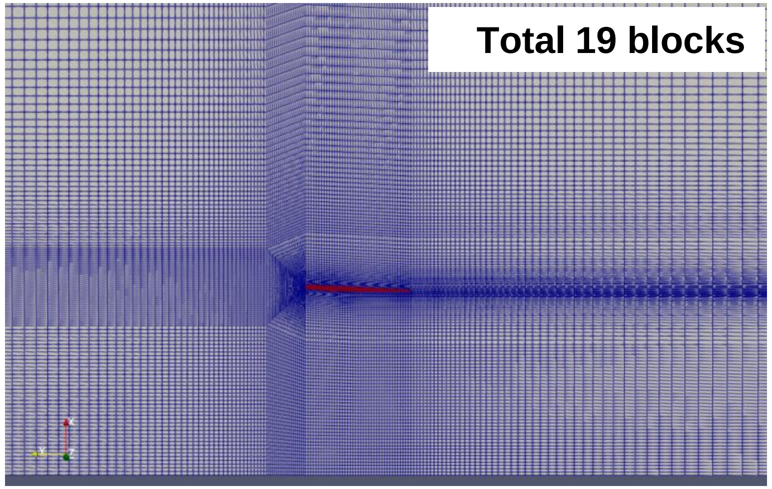


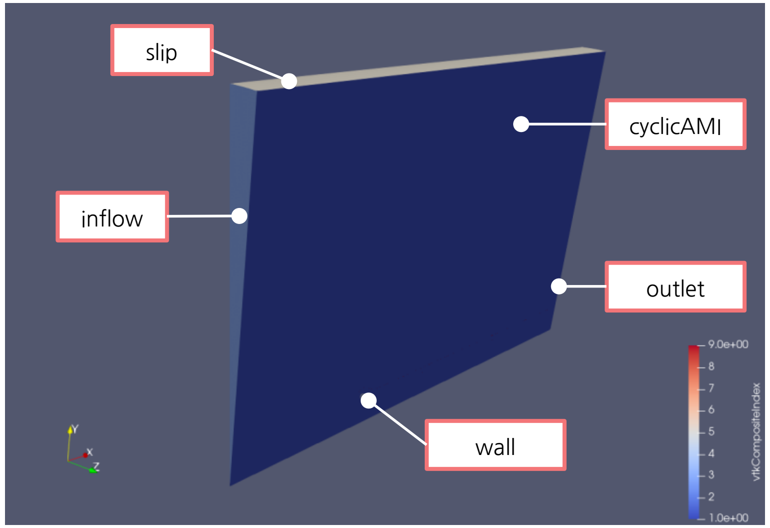
Unit : mm

Parameters	Descriptions	duct01	duct02	duct03	duct04	duct05	duct06	duct07	duct08	duct09	duct10
L_{le}	Distance from duct center to LE	100	100	50	50	50	0	50	30	30	30
D_{le}	Diameter at LE	820	860	820	860	880	860	880	900	900	900
R_{le}	Radius of the lip circle	10	20	10	20	30	40	30	50	20	20
D_{te}	Diameter at TE	780	780	780	780	780	780	790	780	780	820
L_{te}	Distance from duct center to TE	100	100	100	100	100	100	100	100	100	100
L_{tip}	Thickness at TE	5	5	5	5	5	5	5	5	5	5
D_{duct}	Diameter of the duct	780	780	780	780	780	780	780	780	780	780

Duct Design

Parametric Study – pre-processing





Define parameters

↓

Generate duct section profile

↓

Generate blockMeshDict using blockMeshDict.m4

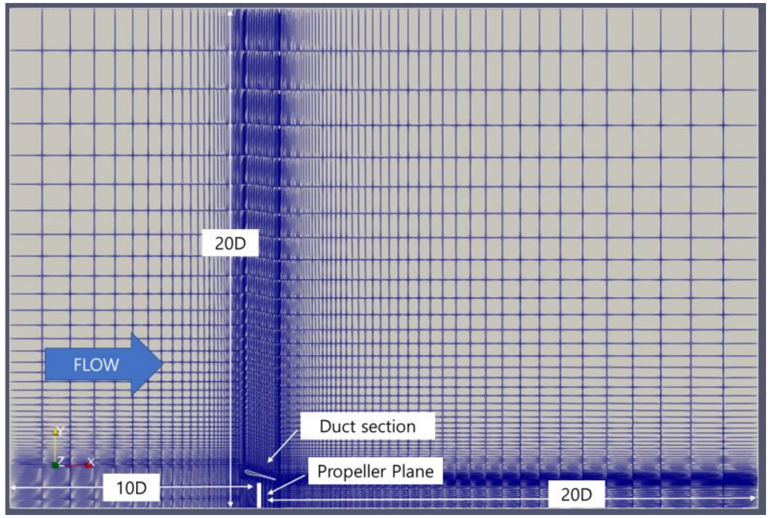
↓

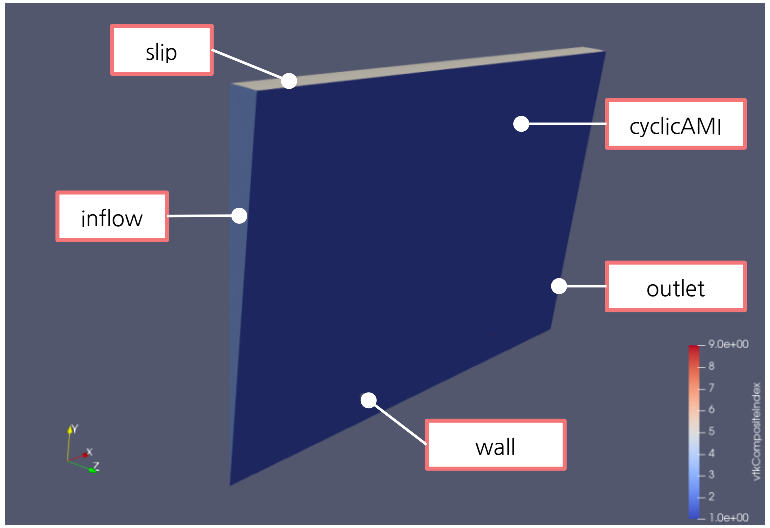
Run blockMesh

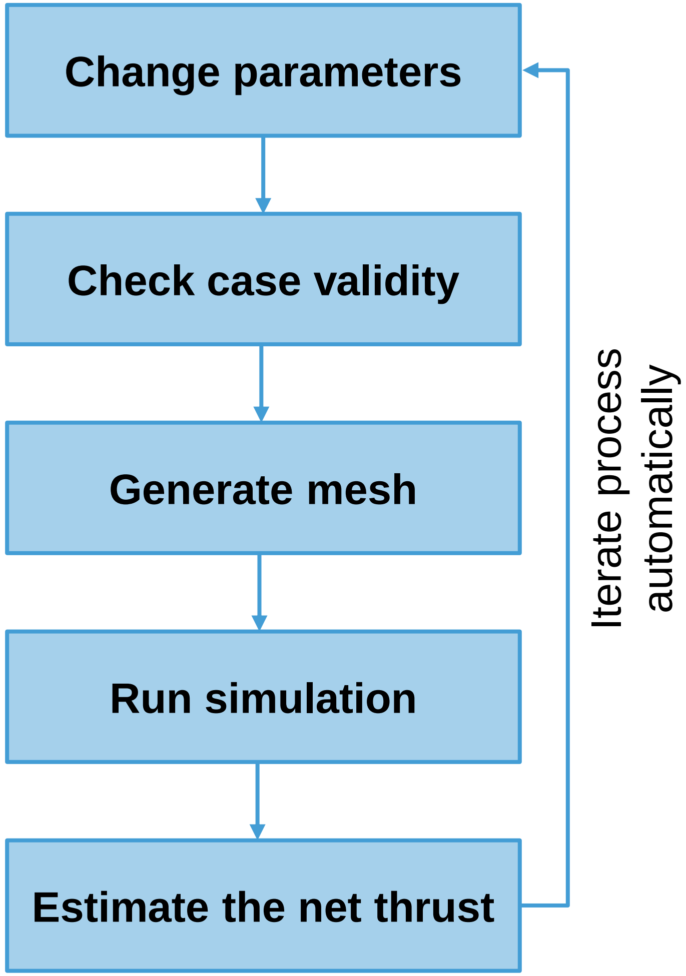
```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2006                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dnl
changecom(//)changequote([,]) dnl
define(calc, [esyscmd(perl -e 'print ($1)')]) dnl
dnl
dnl ==============
dnl  <PARAMETERS>
dnl ==============
dnl - constants
define(pi, 3.14159265359) dnl
define(d2r, calc(pi/180)) dnl
define(cos45, calc(cos(45*d2r))) dnl
dnl
dnl - variables
define(R, 0.35)      dnl
define(doff, 0.1)    dnl
dnl define(xle, calc(R+0.02))    dnl x-coord. of leading edge circle
dnl define(yle, 0.10)    dnl y-coord. of leading edge circle
dnl define(r, 0.01)      dnl radius of the leading edge circle
dnl define(xte, calc(R+0.005))  dnl x-coord. of trailing edge
dnl define(yte, -0.1)    dnl y-coord. of trailing edge
dnl define(xinn, 0.36019412)    dnl
dnl define(yinn, 0.10196080)    dnl
dnl define(xout, 0.37995027)    dnl
dnl define(yout, 0.09900404)    dnl
define(loadpars, include([parameters.m4])) dnl
loadpars dnl
dnl
dnl - resultant points
define(xibi, calc(xle-((doff+r)*cos45)))    dnl
define(xibo, calc(xle+((doff+r)*cos45)))    dnl
define(yib, calc(yle+((doff+r)*cos45))) dnl
define(xlci, calc(xle-(r*cos45)))   dnl
define(xlco, calc(xle+(r*cos45)))   dnl
define(ylc, calc(yle+(r*cos45)))    dnl
dnl
define(ptno, 0) dnl
```

Duct Design

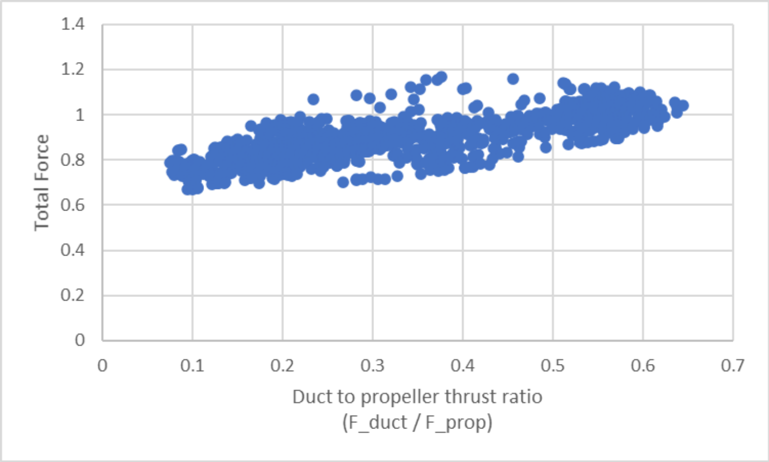
Parametric Study

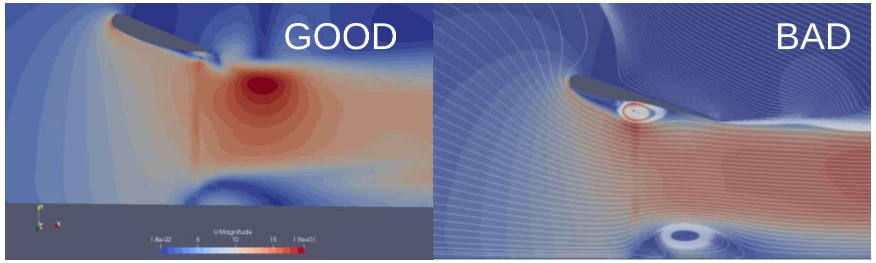






- Total 2,175 cases were evaluated
- Steady RANS simulation with k-omega SST turbulence model
- Momentum source method was used for reflecting the effect of a propeller

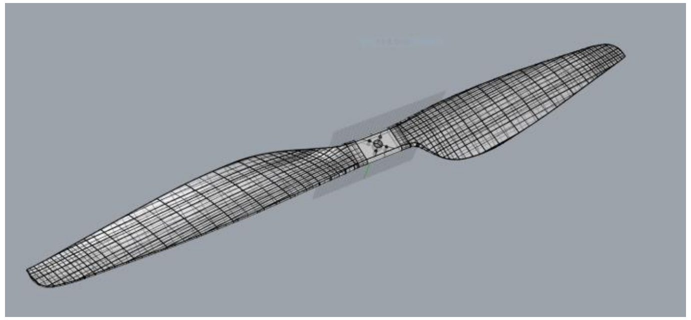




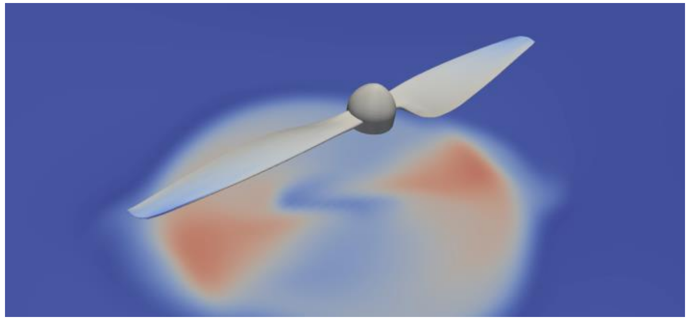
Propeller Design

Stock Design

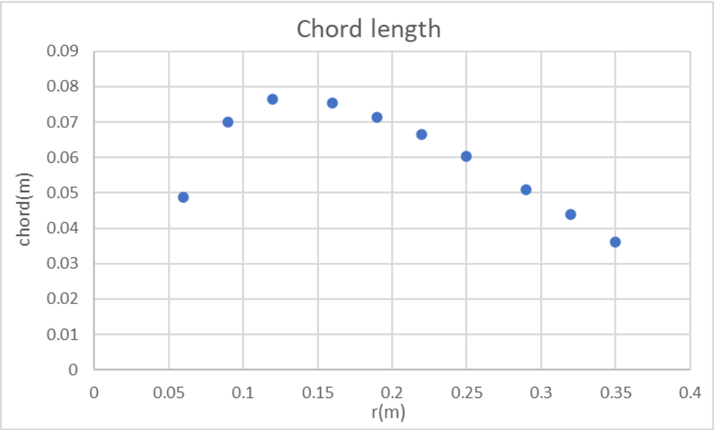
T-motor G30x10.5 Propeller

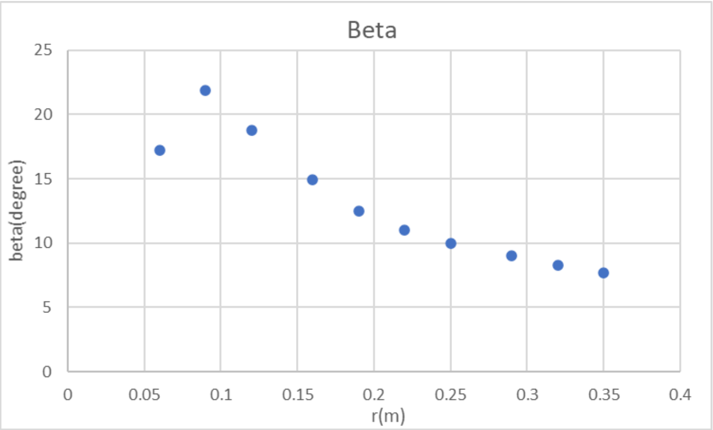


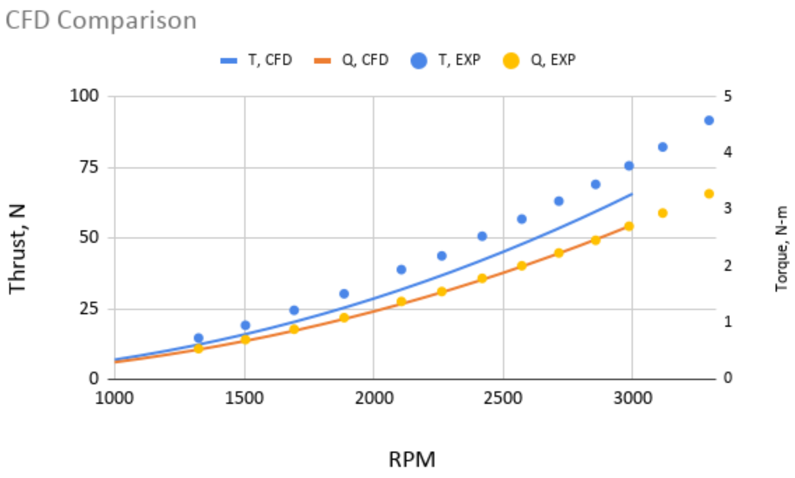
3D Scanned Geometry



CFD Simulation



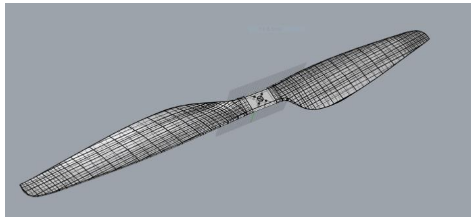




Propeller Design

Evolution of the propeller design

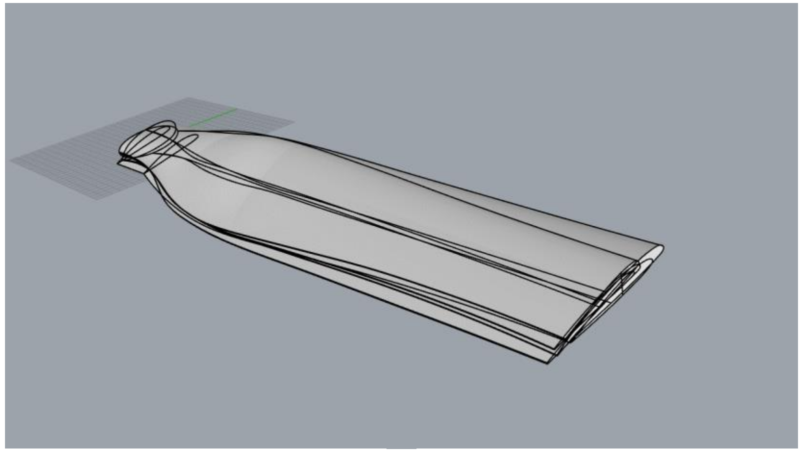
STOCK PROPELLER

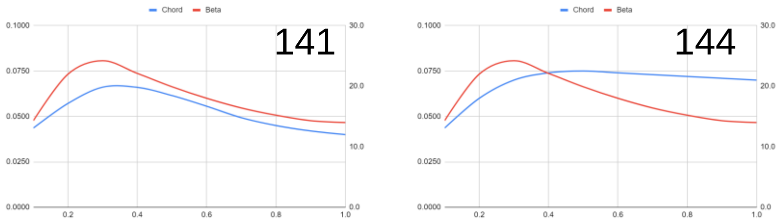


Computational Fluid Dynamics

- Validation of the computational method
- Propeller characteristics

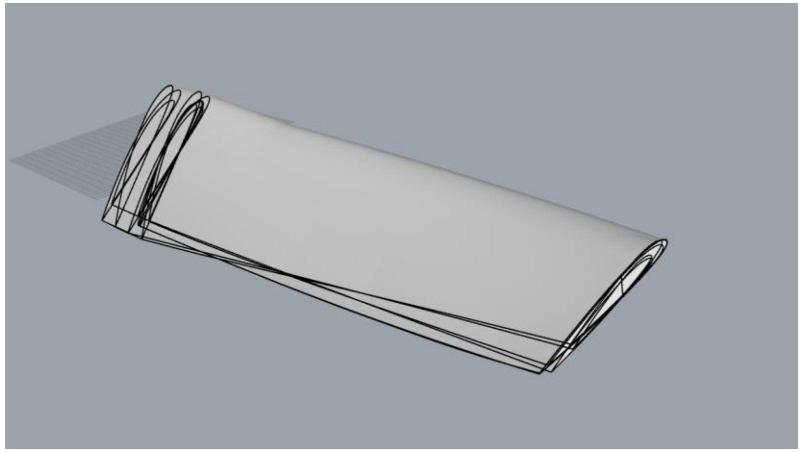
Generation 1

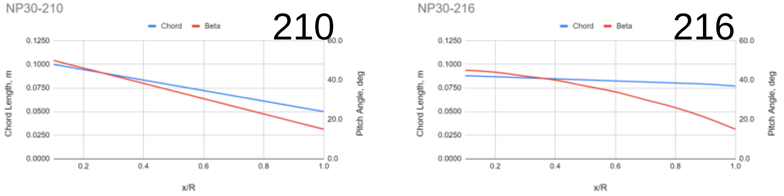




- Commercial propeller based variation
- Increased tip loading

Generation 2

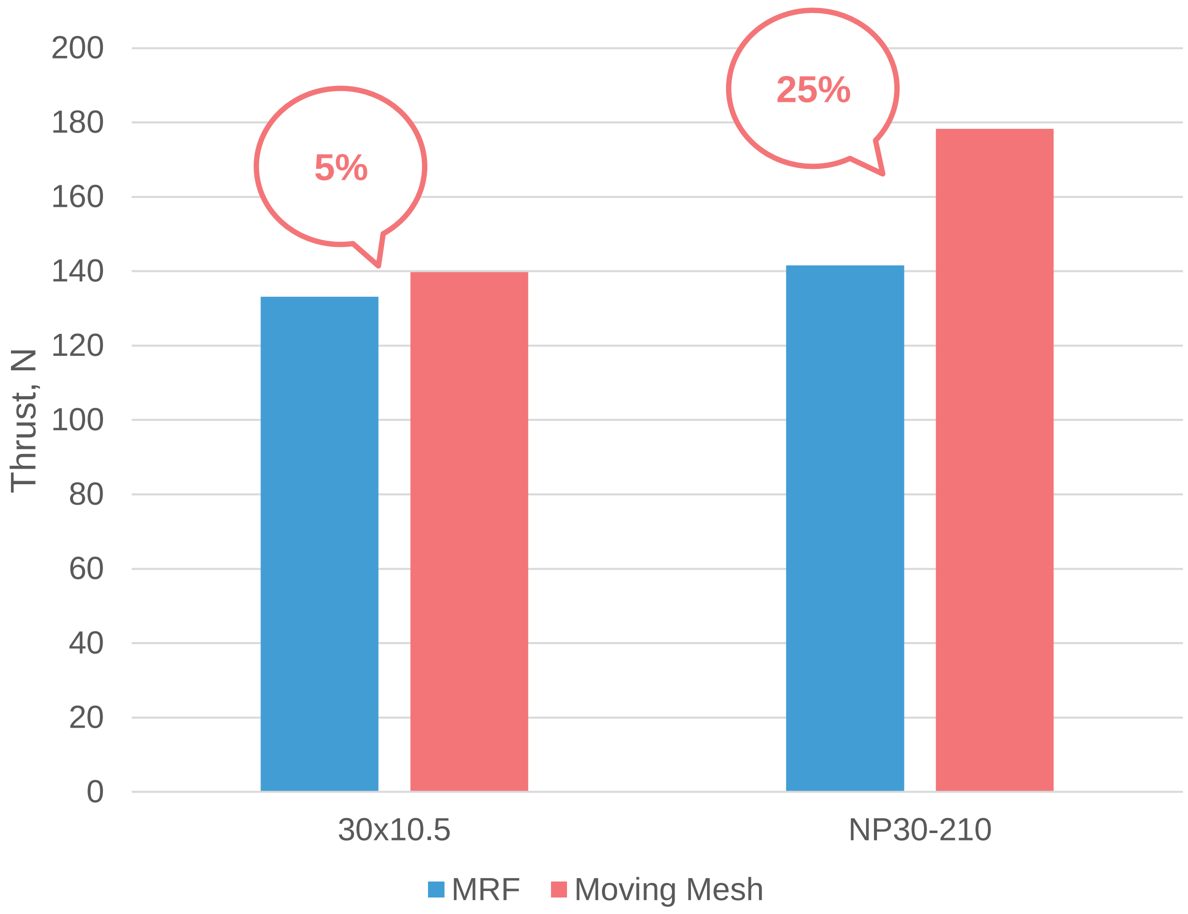




- Gradually increasing blade loading
- Increased total thrust

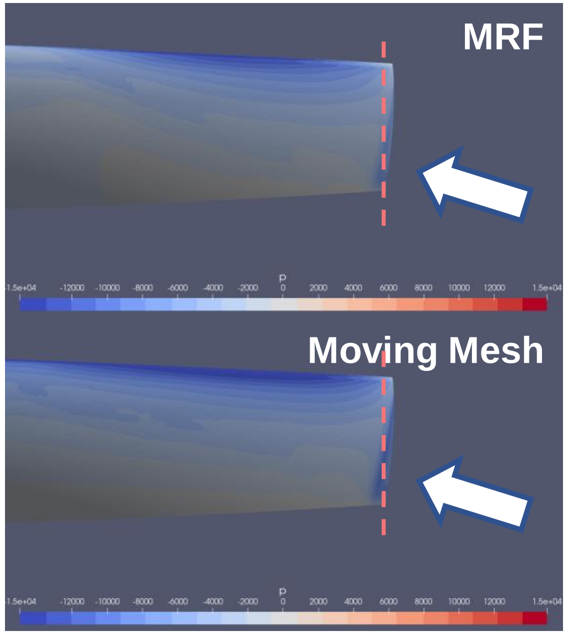
Propeller Design

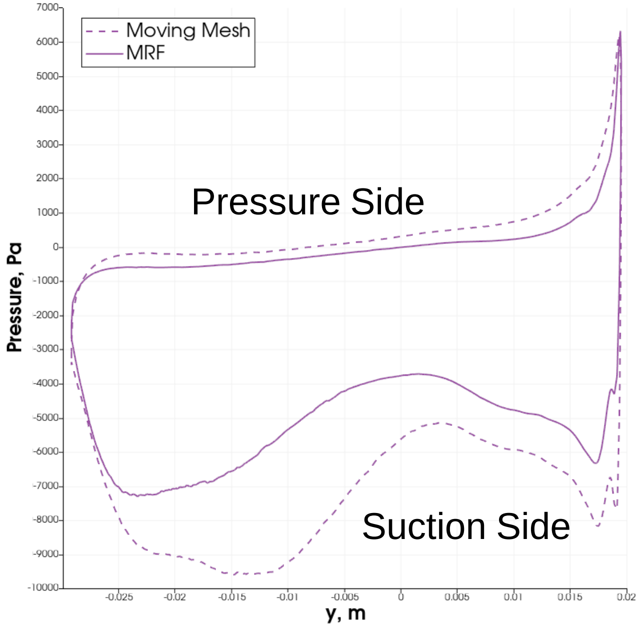
Remarks on the MRF & Moving Mesh on Propeller Simulations



Why are the MRF and Moving Mesh results so different?

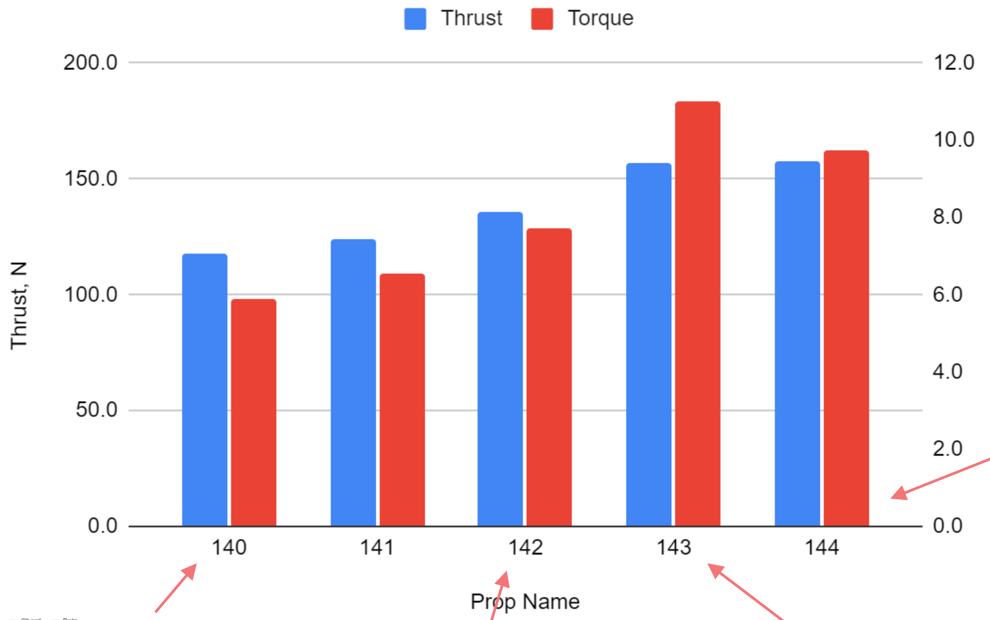
[Hypothesis]
MRF and Moving Mesh predict the tip vortical flow structure differently.

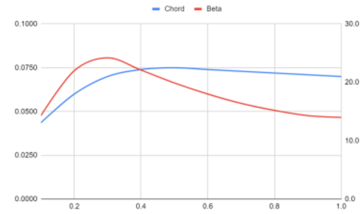




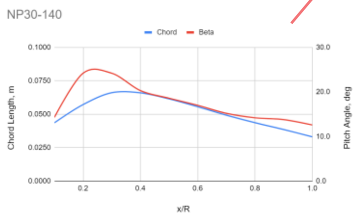
Propeller Design

Numerical simulations on the propeller

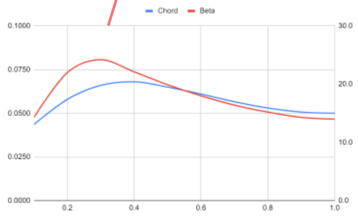




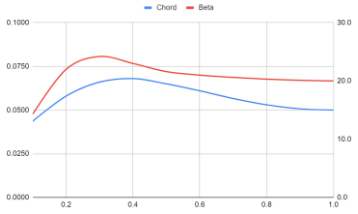
142 variation
Increased chord



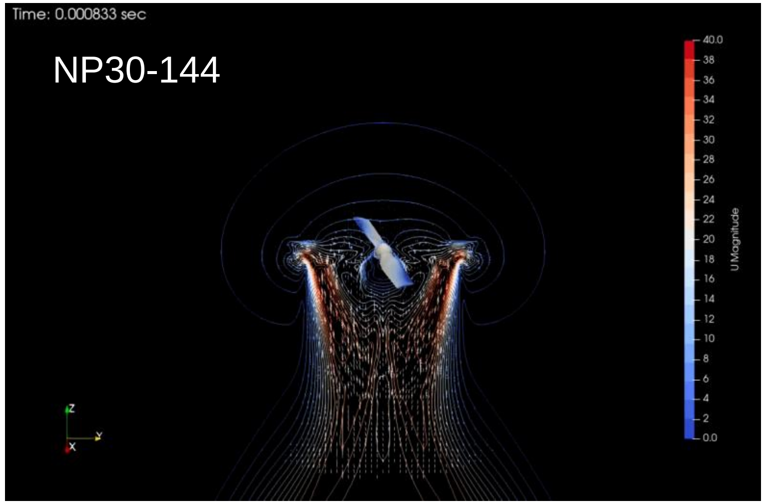
Optimum design derived from 30x10.5 commercial propeller using potential code

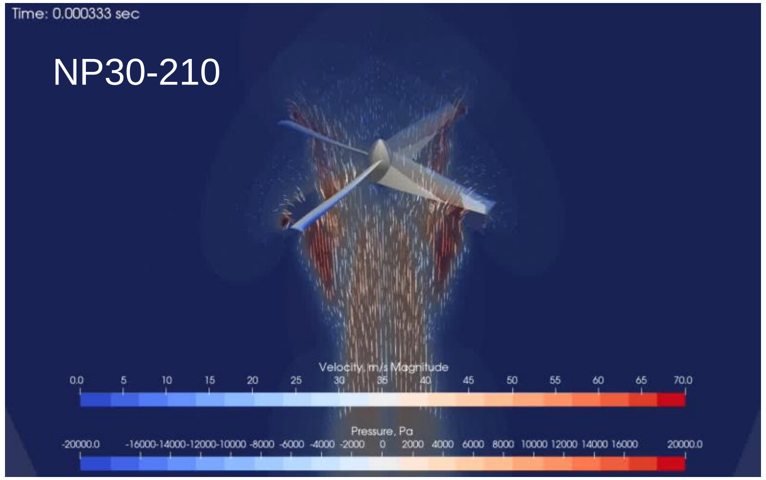


Redistributed blade angle & extended chord length at tip



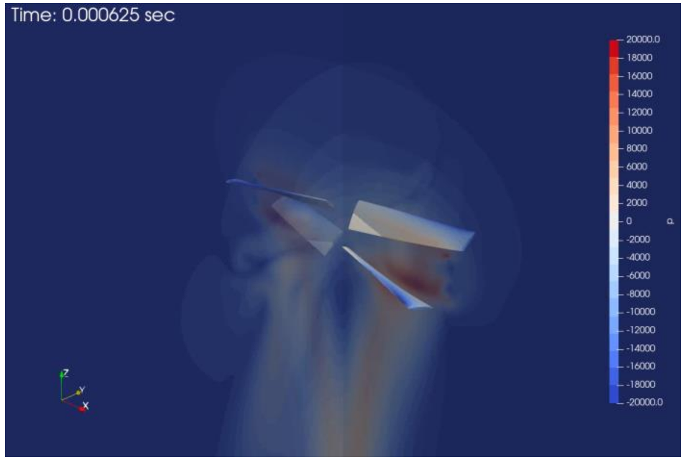
142 variation
Higher beta

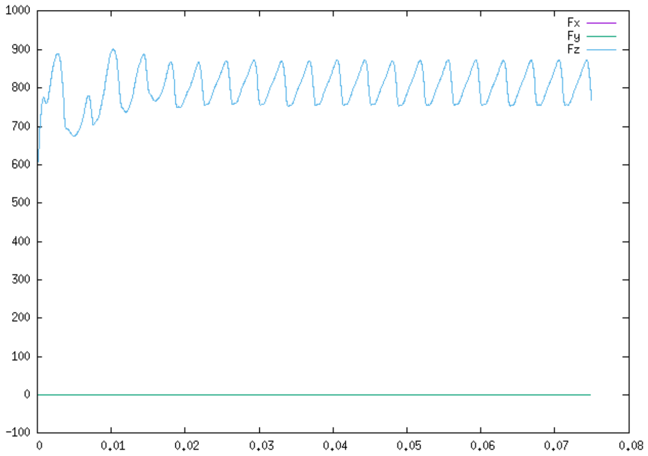




Various Configurations

Counter-rotating Configuration





		Thrust [N]	Torque [N-m]	S.Thrust [g/W]
	4-Blade	670.40	64.24	2.54
Counter-rotating	**Top**	371.67	33.14	2.73
	Bottom	436.64	42.01	2.53
	Total	808.32	75.15	2.62

Pre-processing

Prepare ambient fluid region with multiple rotating interfaces (snappyHexMesh)

↓

Copy the rotate region and run transformPoints

↓

Run mirrorMesh for bottom region

↓

Run mergeMeshes

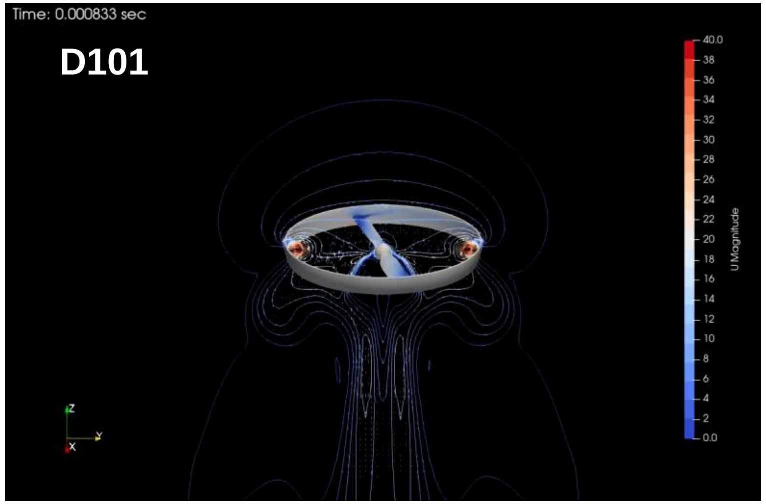
↓

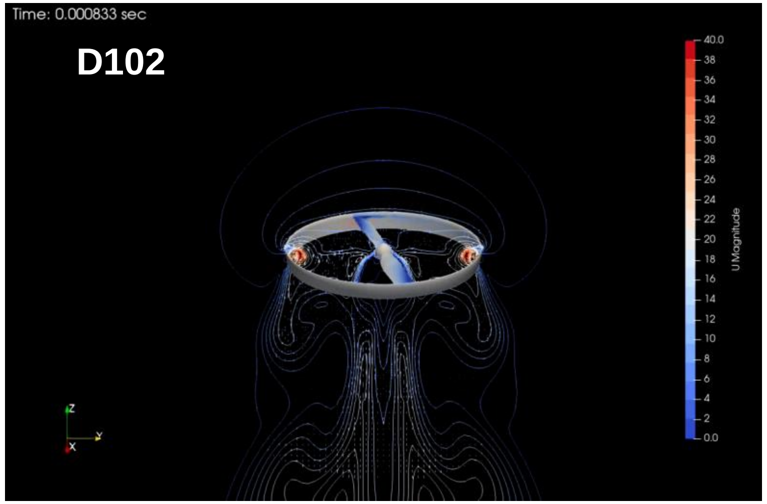
Run createPatch

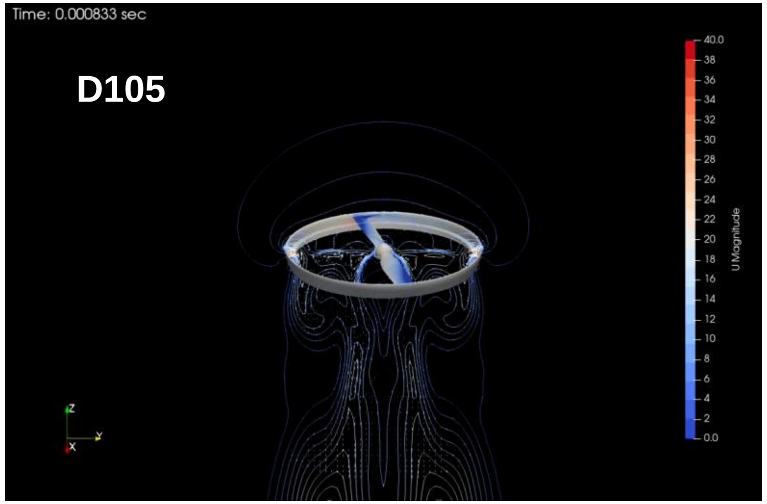
↓

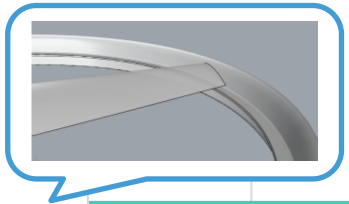
Run simulation

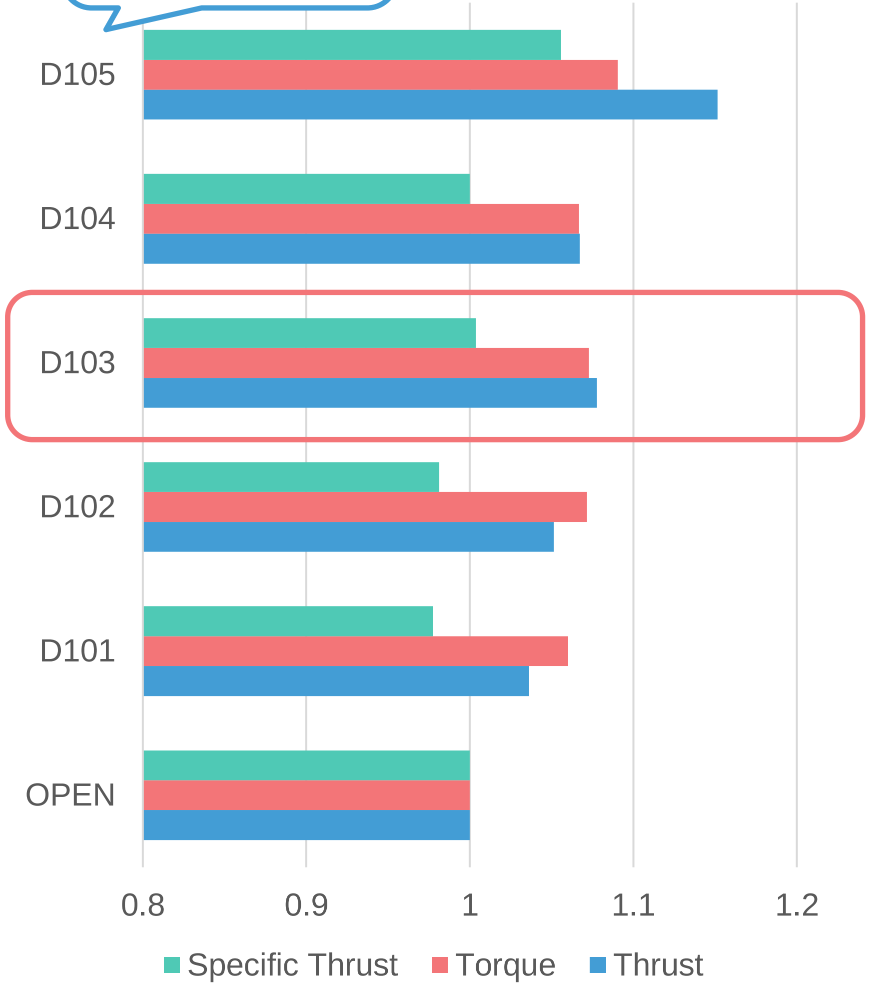
Ducted Propeller Simulations





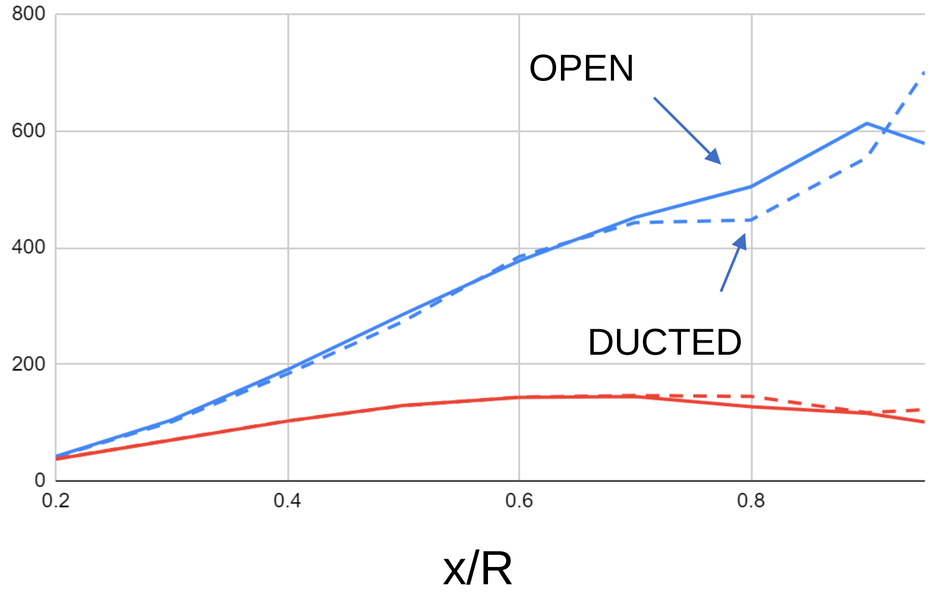


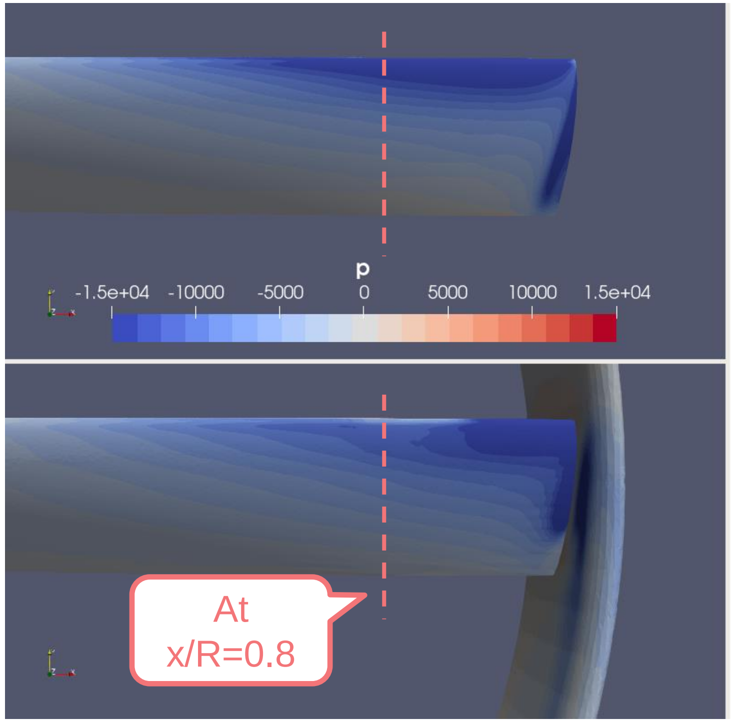


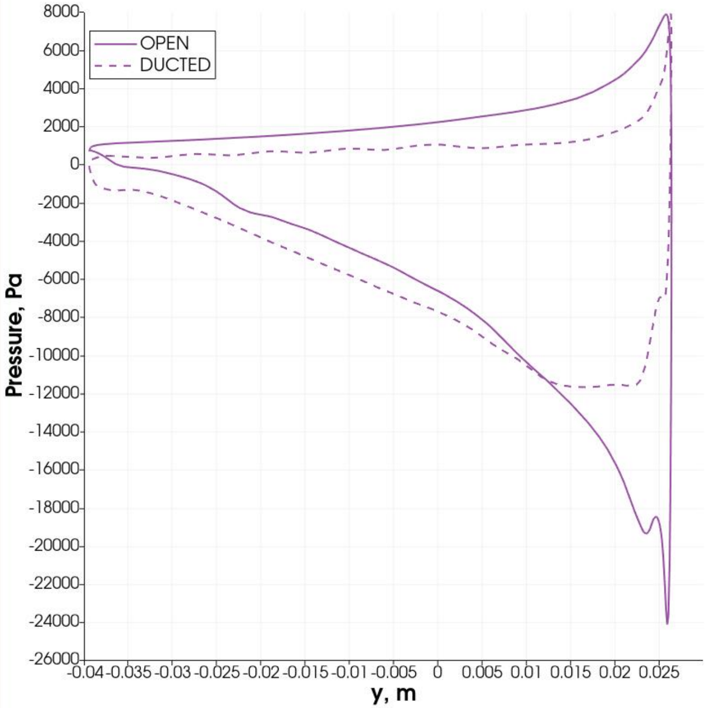


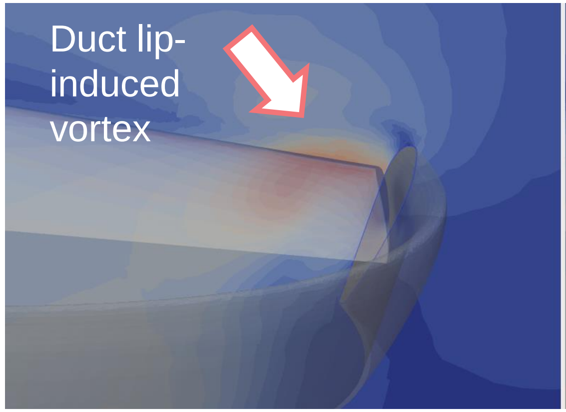
Ducted Propeller Simulations

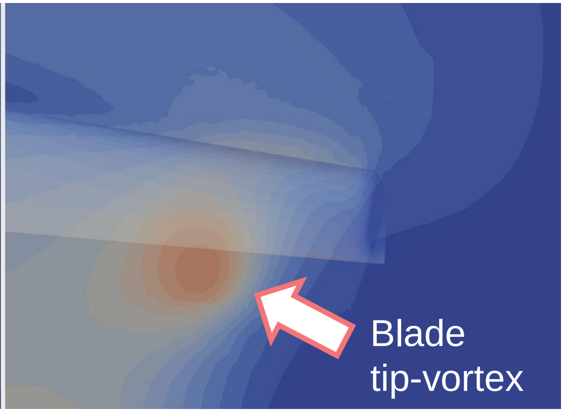
Sectional thrust and torque
P210-OPEN & D103







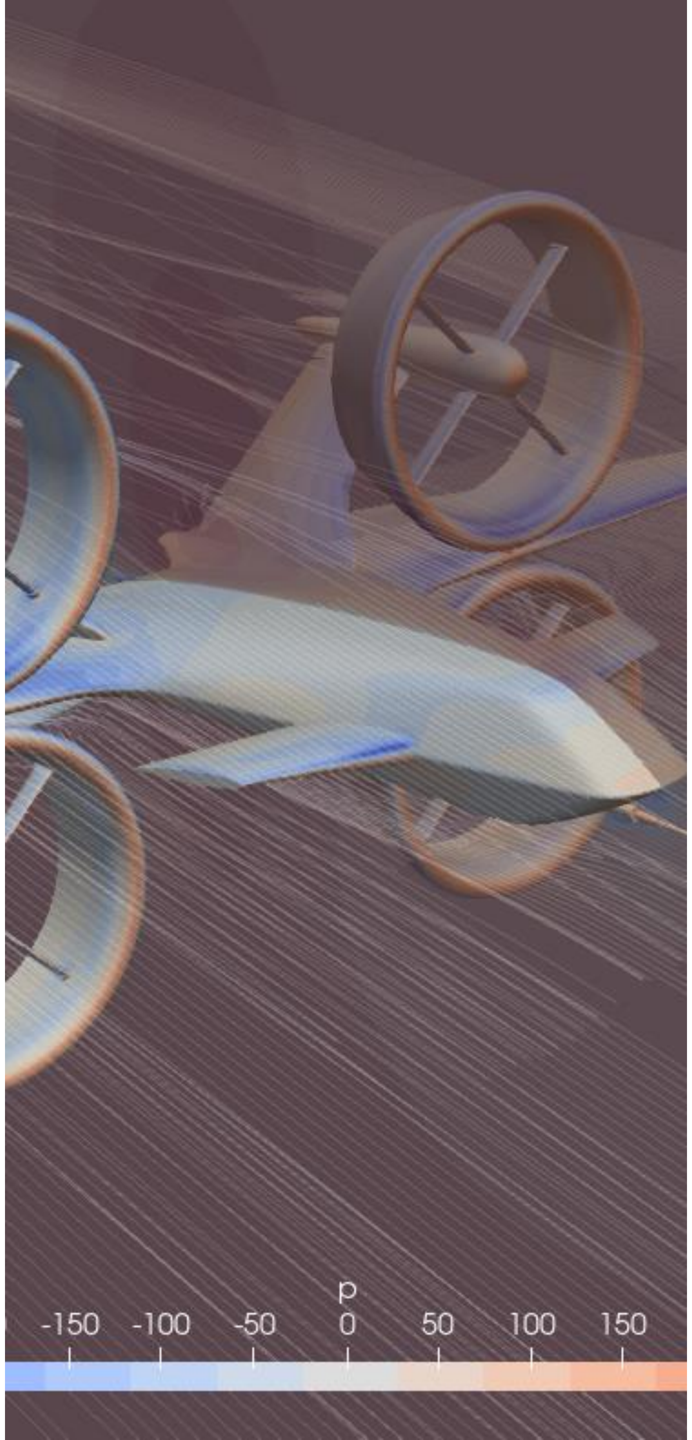




- The interaction between duct lip and propeller generates a vortical flow
- Such a vortical flow was observed to lower the sectional lift which causes thrust loss
- And the vortical flow might adversely effect on the noise performance

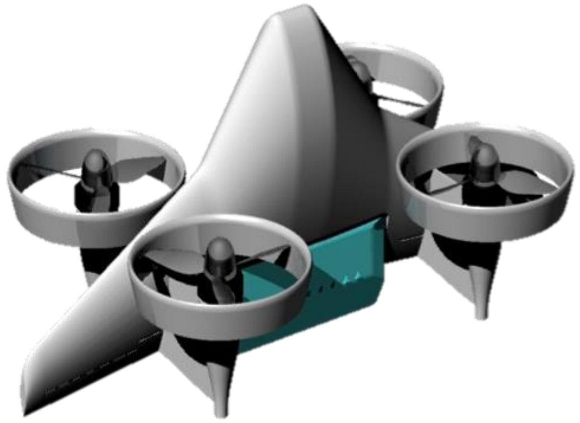
Concluding Remarks

- Design of a ducted propeller system for a tail-sitter UAV has been performed by using the CFD techniques
- Following methods have been applied
  - Potential codes, such as XROTOR and DFDC
  - Star-CCM+ was used for mesh generation
  - Modified OpenFOAM solvers were used to predict the performance of ducted propellers
- For the duct design, over 2,000 cases were investigated by using the simplified numerical simulations
- As a reference propeller, a commercial propeller was scanned and used
- MRF and Moving Mesh methods were used to evaluate the propeller performance
- Finally, a combination of a duct (D103) and propeller (NP30-215) was selected that meets the design requirement.



Future Works

Initial Design



Generation 1 (Present)

Generation 2 (Developing)

- Interaction between the tail-sitter UAV and ducted propeller system
- Duct design optimization over whole flight envelop (including vertical and horizontal flights)
- Pitch control system for a ducted propeller
- Noise reduction design (both blade and duct)

NINANO COMPANY

Thank you

Lifting heavier, flying faster, reaching further

geonhong.kim@ninanobiz.com

https://ninano.company