

2016 5th OpenFOAM Korea Users' Community Conference

9월 29일 (목-금), 2016, 한국해양대학교 해양과학기술대학관 100호 (부산)

오픈폼을 이용한 120W급 동축반전팬의 유동해석

이 명 성 선임연구원

지능메카트로닉스 연구센터

 **KETI** 전자부품연구원
Korea Electronics Technology Institute

발표순서

동축반전구조

1차 설계와 풍동 실험

수치해석

파라미터 스터디

요약 및 다음연구

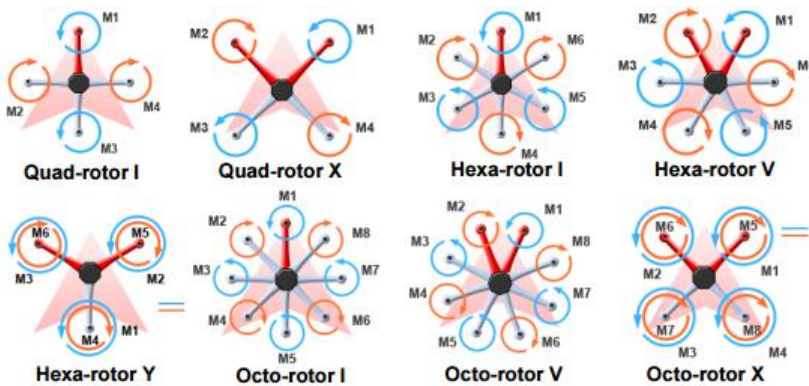
동축반전구조

항공과 유체기계 분야

항공분야 동축반전구조

단일 로터 방식

- 정방향과 역방향을 적절히 배치하여 기체의 회전 방지



동축반전 로터 방식

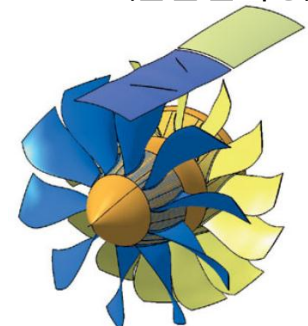
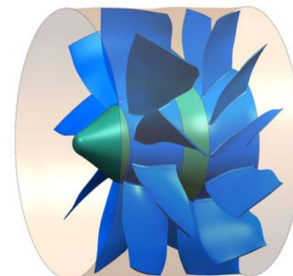
- 하나의 축에 서로 반대 방향으로 회전하는 로터로 인해 토크 상쇄 → 조종 안정성 및 추력 증가



Ka-20 (1961)



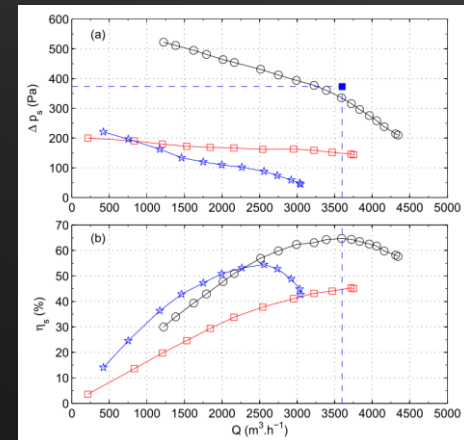
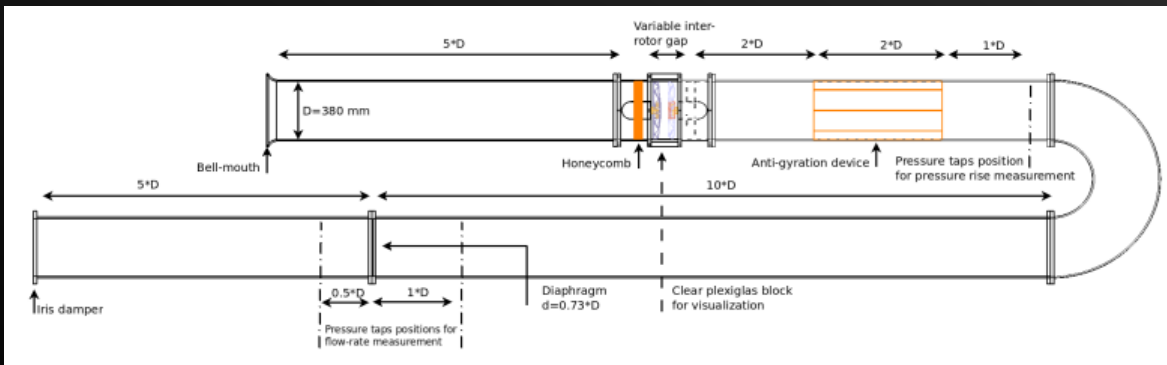
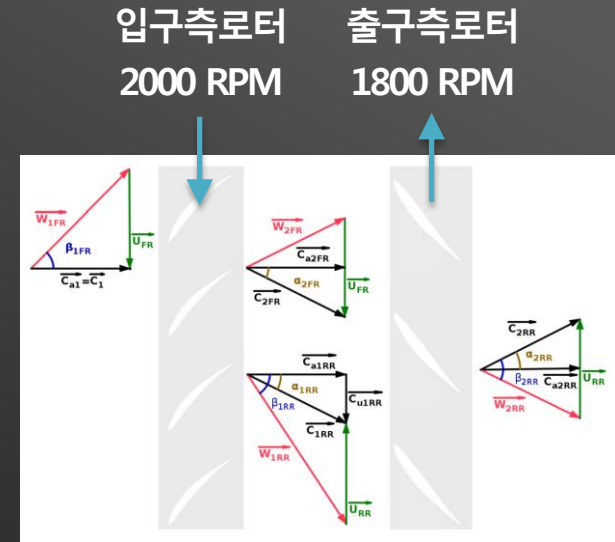
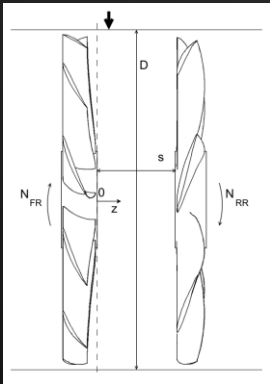
Ka-32 (산불진화용)



유체기계분야 동축반전구조

동축반전 축류팬과 단일 축류팬의 성능 비교

- 팬효율 : 단일로터 45% → 동축반전 67%



○ 동축반전

□ 입구축로터

★ 출구축로터

산요전기 동축반전팬

동축반전 축류팬 제품동향

- 냉각팬 전문 메이커인 **산요전기**에서 다양한 용량의 동축반전 축류팬 개발함
- **국내 부품 업체**들에서 동축반전 축류팬 설계기술 확보에 **관심을 보이고 있음**



동축반전 구조에서 스윙 속도성분 감소



- 동축반전 축류팬 구조가 일반축류팬 2단 구조에 비해 소비전력 10% 절감



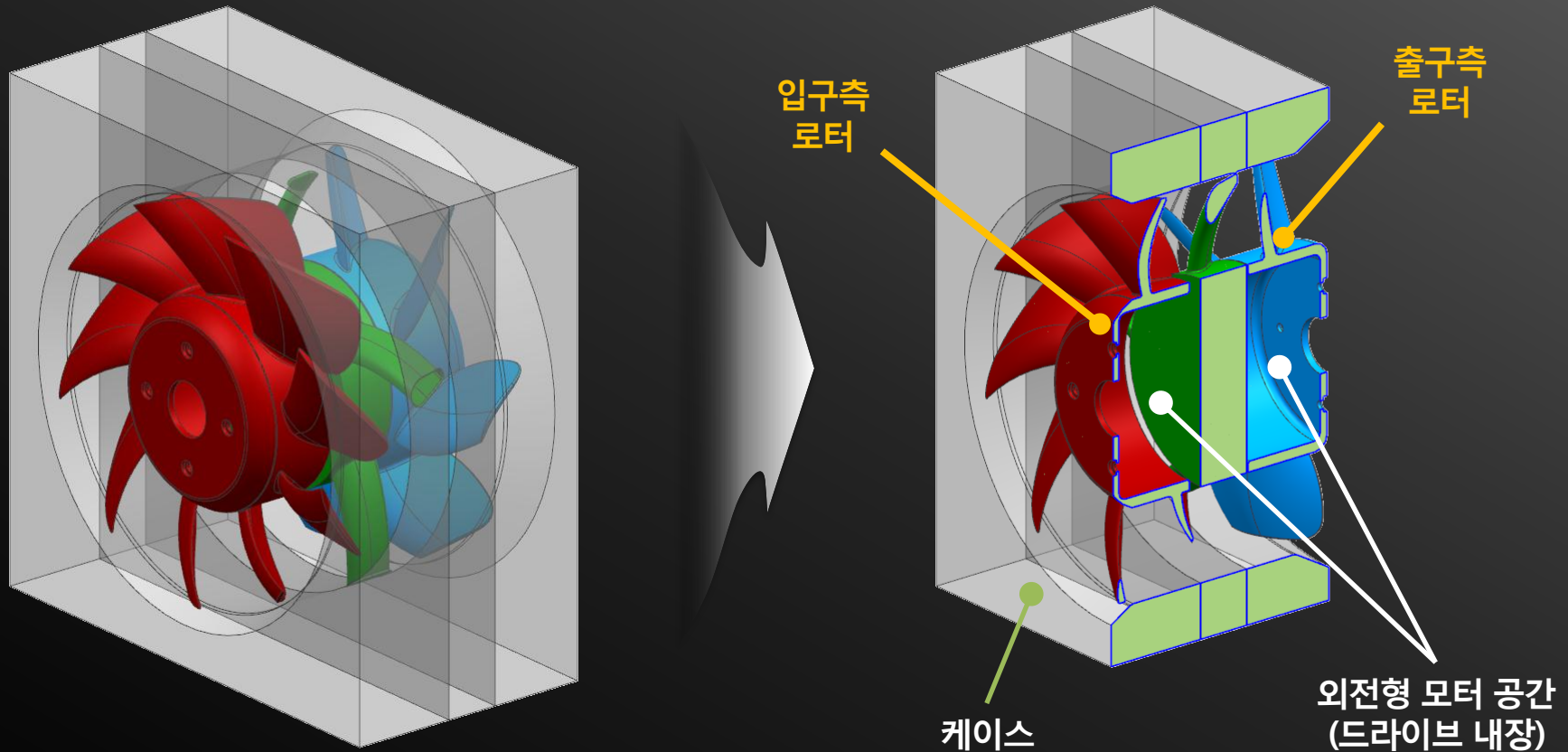
1차설계와 풍동실험

Cambered 4-digit NACA airfoil & Wind tunnel experiment

1차 설계

동축반전 축류팬 3D 모델링

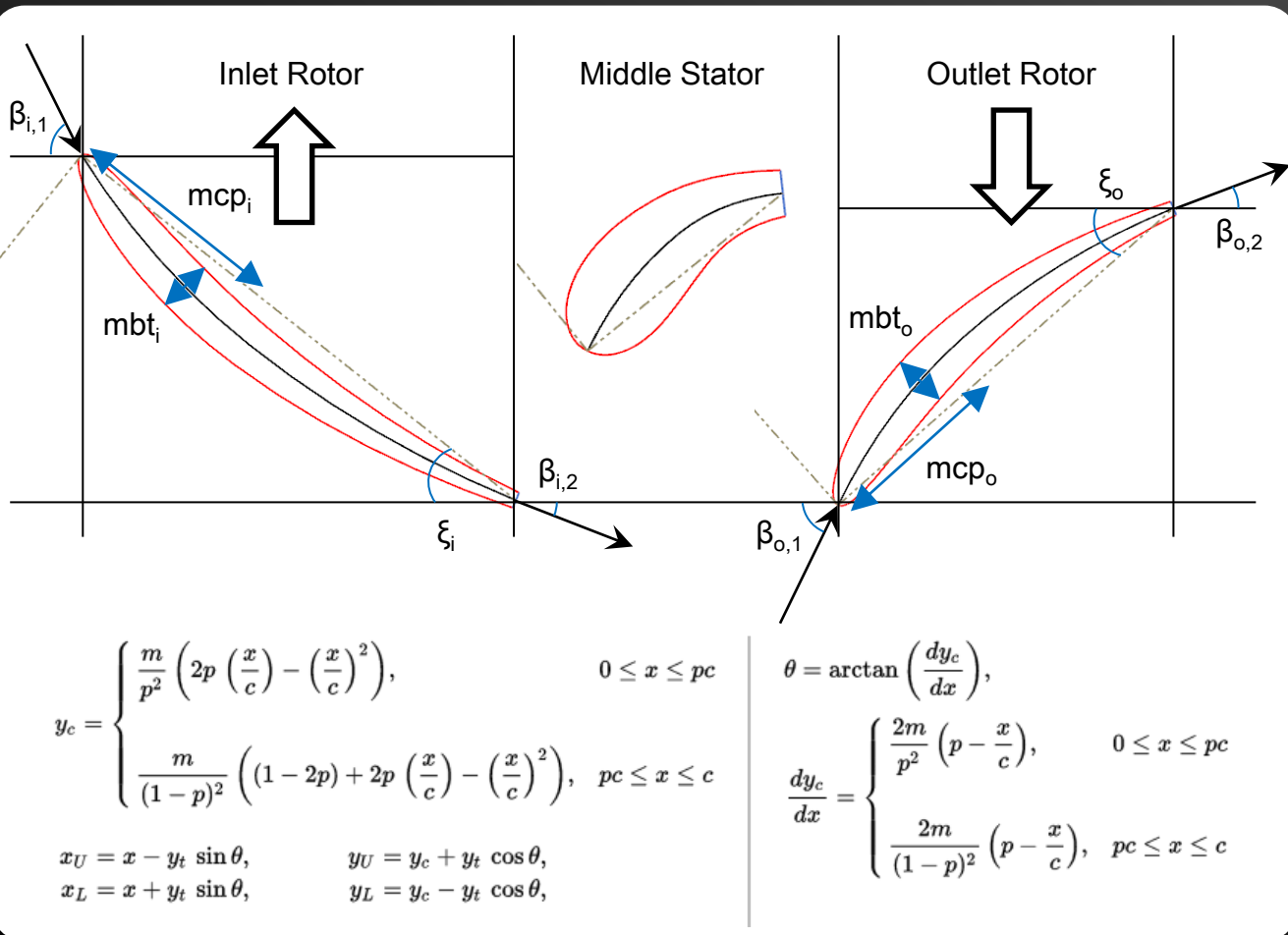
- 산요전기 선진샘플의 형상에 기반함 : 블레이드 개수 및 형상



블레이드 설계방법

Cambered 4-digit NACA airfoil 모델링

- 허브, 중간, 슈라우드 면에 각각 airfoil 방식으로 2D 스케치 → 3D 블레이드 생성

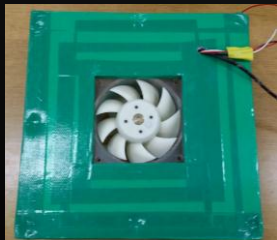
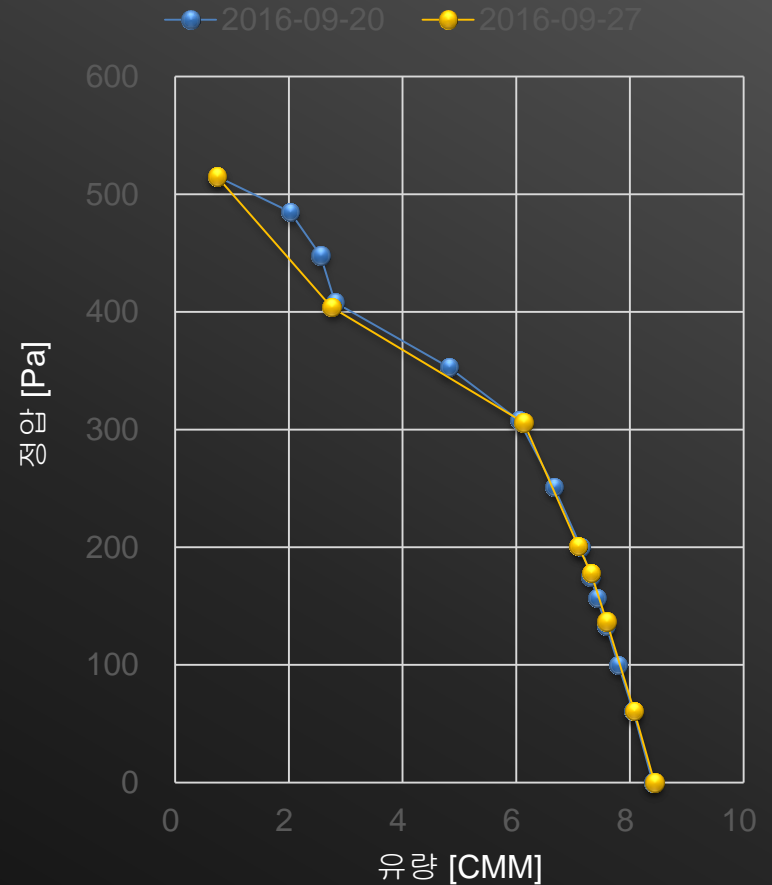
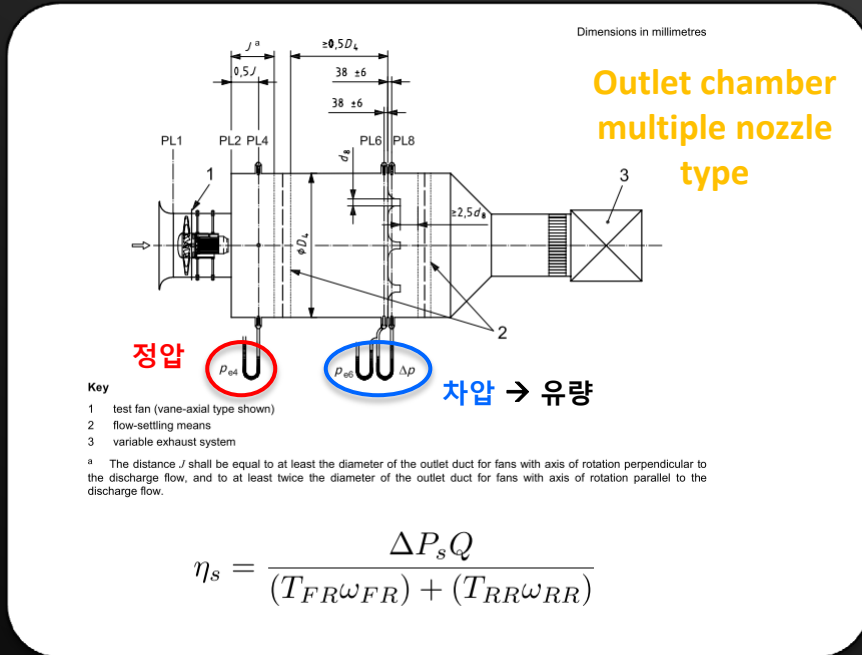


풍동실험

팬의 공력 성능 측정

- 유량별 정압 측정 (KSB 6311, 송풍기 시험규격)

실험 재현성 확인



수치해석

오픈폼

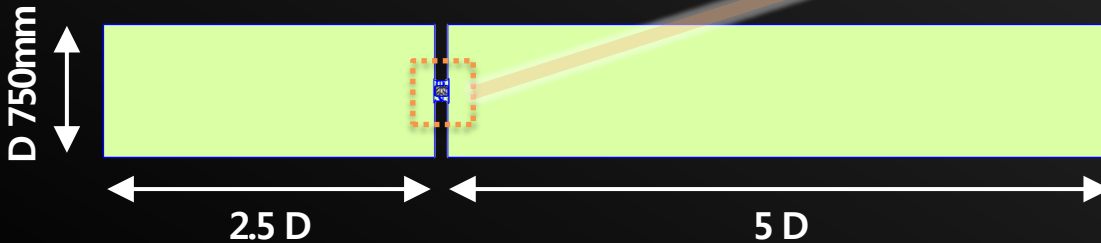
계산 영역 및 경계 조건

■ 풍동 모사한 계산 영역 구성

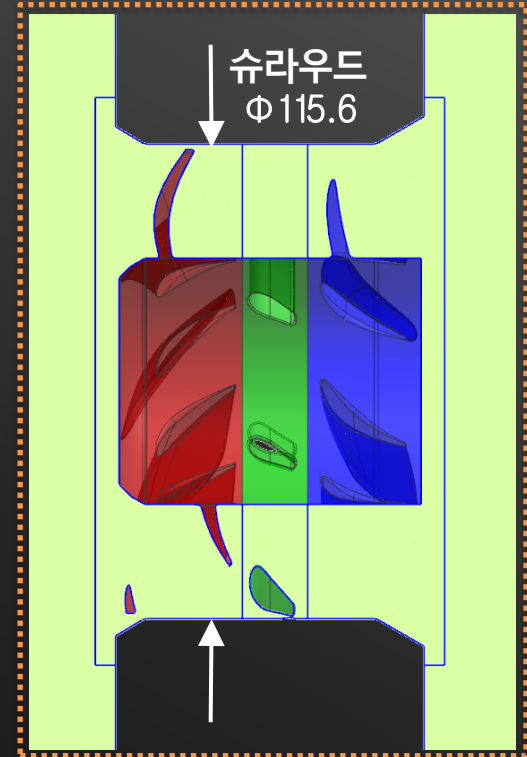
- 경계조건 : 입구 유량 / 출구 압력
 - 유량 : 5~8 CMM

입구
(유량조건
5~8 CMM)

- 로터회전조건
 - 입구측로터 6200 RPM
 - 출구측로터 3800 RPM



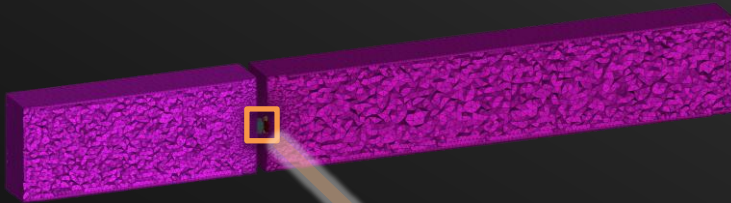
출구
(압력조건, $p=0$)



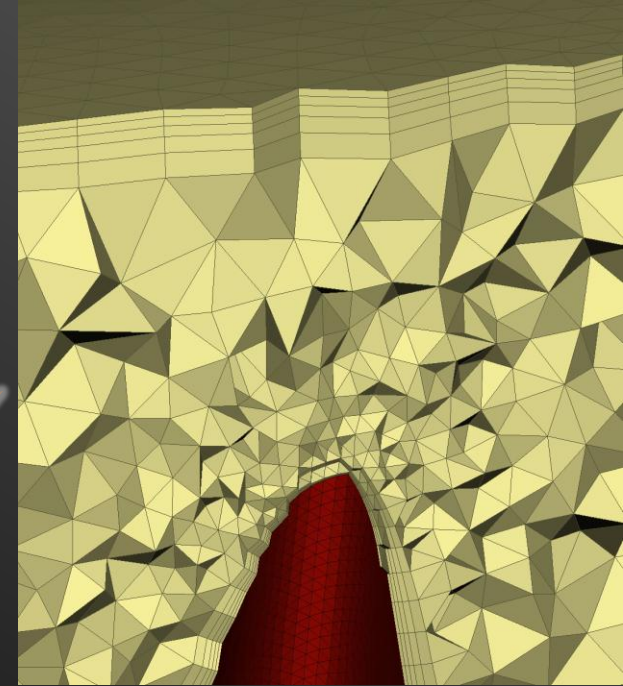
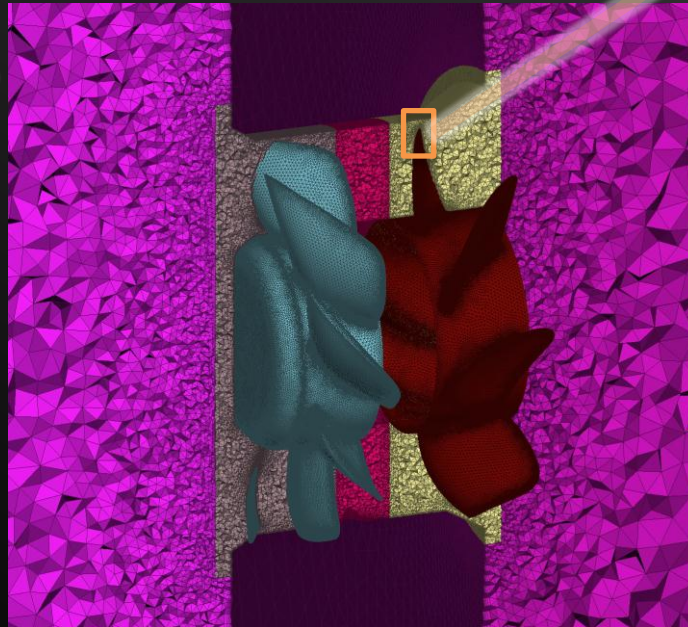
격자 분포

비정렬 격자 생성

- 총 2천 4백 만개 격자 사용 (tetra meshing)
 - 회전영역에 70% 이상 격자 집중 (1천 7백 만개)
- 상용격자생성 프로그램 이용
 - ANSA v16 (OpenFOAM 격자 파일 내보내기 지원)



회전영역에
격자 집중



벽면 레이어 격자

오픈폼 케이스 디렉토리

0

- epsilon
- k
- p
- U

- 초기 epsilon 값 w/ epsilonStdWallFunction
- 초기 k 값 w/ kqRWallFunction
- 출구 압력 0 (zero, kinematic pressure 주의)
- 입구 유량 ? [m²/s] t/ flowRateInletVelocity

constant

- polyMesh
 - boundary
 - cellZones
 - faces
 - faceZones
 - neighbour
 - owner
 - points

ANSA 에서 내보낸
파일 그대로 사용

- RASProperties
- **rotatingZoneProperties_rotatingIn**
- **rotatingZoneProperties_rotatingOut**
- transportProperties

- 서로 맞는 인터페이스들을 boundary 파일 내에서 짝 지워줌
- rotatingIn_interface_stationaryMid

```
{ type cyclicAMI;
startFace 52970277;
nFaces 74924;
matchTolerance 0.0001;
transformnoOrdering;
neighbourPatch stationaryMid_interface_rotatingIn;
}
```
- nonRotatingPatches

```
( rotatingIn_interface1_stationaryIn
rotatingIn_interface2_stationaryIn
rotatingIn_wall_shroud_CR
rotatingIn_interface_stationaryMid );
origin (0 0 0);
axis (0 0 1);
omega -649.262; // -6,200 RPM
```
- standardKEpsilon;
- nu=1.55e-5; // air@25C

system

- controlDict
- decomposeDict
- fvOptions
- fvSchemes
- fvSolution
- **torquePressureFunc**

- endTime, writeInterval, purgeWrite
- MRF1_rotatingIn

```
{ type MRFSource;
selectionMode cellZone;
cellZone rotatingIn;
active yes;
MRFSourceCoeffs
{ #include
"${FOAM_CASE}/constant/rotatingZoneProperties_rotatingIn"
}}
```

수렴 판단

torquePressureFunc

- 입구 압력과 팬에 걸리는 토크 출력 기능

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       forceCoeffsDict;
}
// ***** //

inletPressure
{
    type                faceSource;
    functionObjectLibs  ("libfieldFunctionObjects.so");
    enabled              yes;
    outputControl        timeStep;
    outputInterval      1;
    log                  yes;
    writeTotalArea      no;
    valueOutput         no;
    source               patch;
    sourceName           stationaryIn_inlet;
    operation            areaAverage;
    fields
    (
        p
    );
}
```

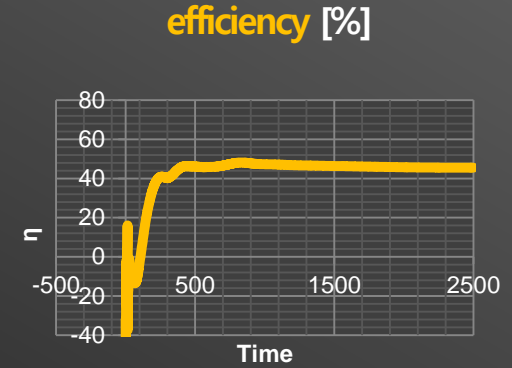
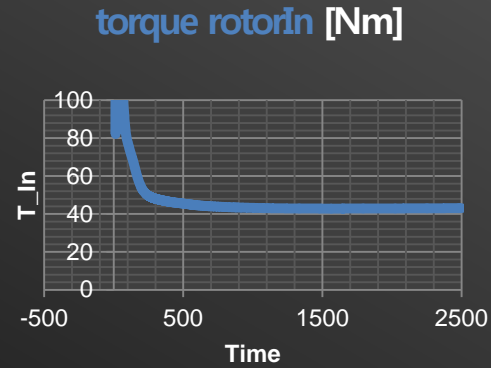
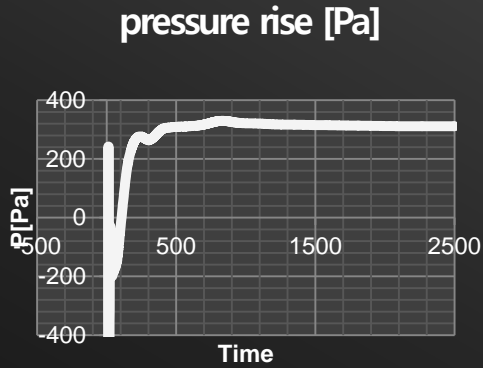
```
forceMoment_rotatingIn
{
    type                forces;
    functionObjectLibs  ("libforces.so" );
    outputControl        timeStep;
    outputInterval      1;
    patches              (rotatingIn_wall_rotor_R);
    pName                p;
    UName                U;
    rhoName              rhoInf;
    log                  true;
    rhoInf               1.185;
    CofR                 (0 0 0);
}

forceMoment_rotatingOut
{
    type                forces;
    functionObjectLibs  ("libforces.so" );
    outputControl        timeStep;
    outputInterval      1;
    patches              (rotatingOut_wall_rotor_R);
    pName                p;
    UName                U;
    rhoName              rhoInf;
    log                  true;
    rhoInf               1.185;
    CofR                 (0 0 0);
}
```

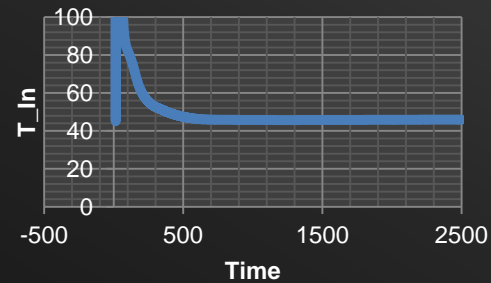
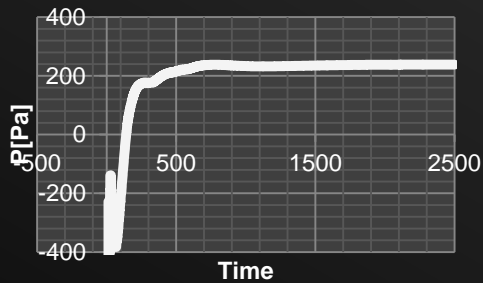
계산 모니터링

압력, 토크, 효율 모니터링 (w/ torquePressureFunc)

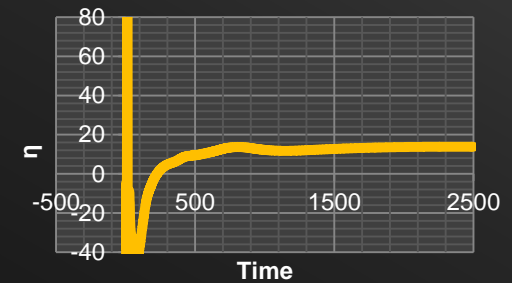
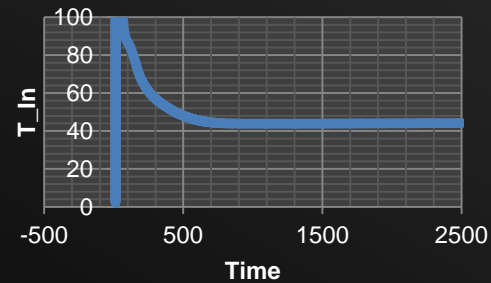
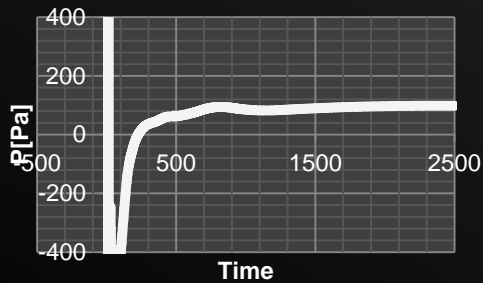
5 CMM



6 CMM



7 CMM



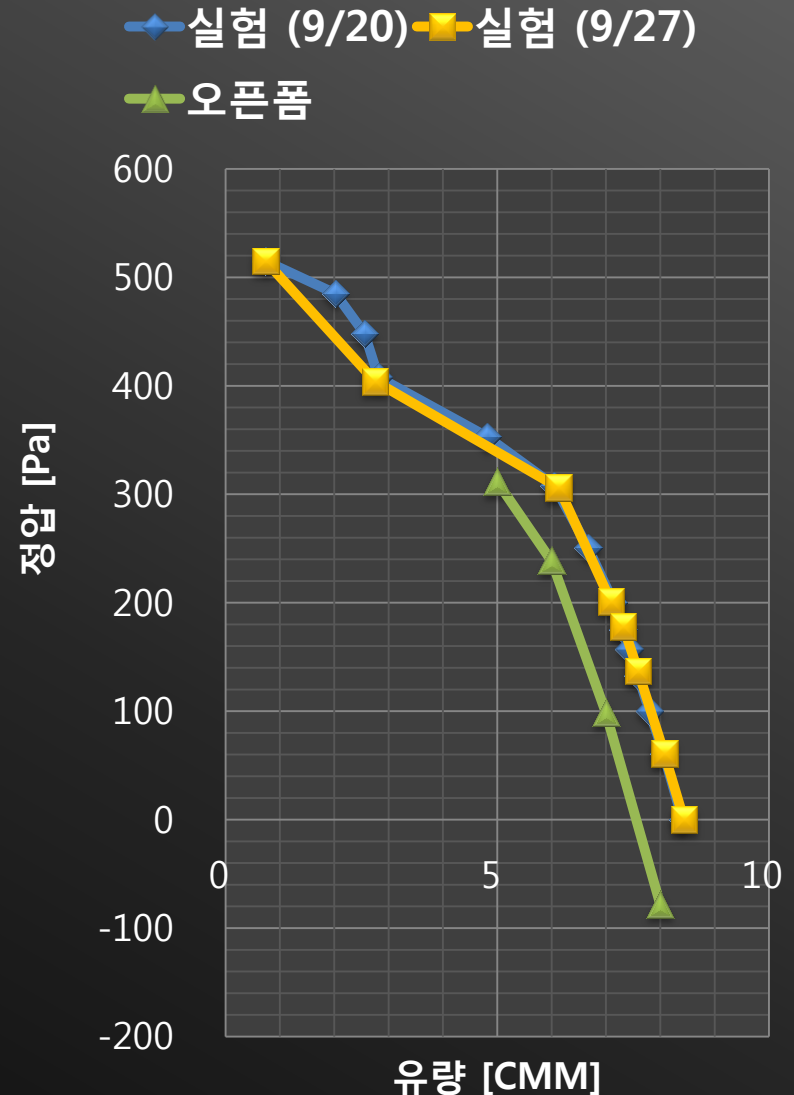
정상상태 해석자원 및 성능예측 결과

정상상태 해석에 필요한 해석 자원

- 한 케이스에 4일 (8 CPUs, Mem < 24 GB)

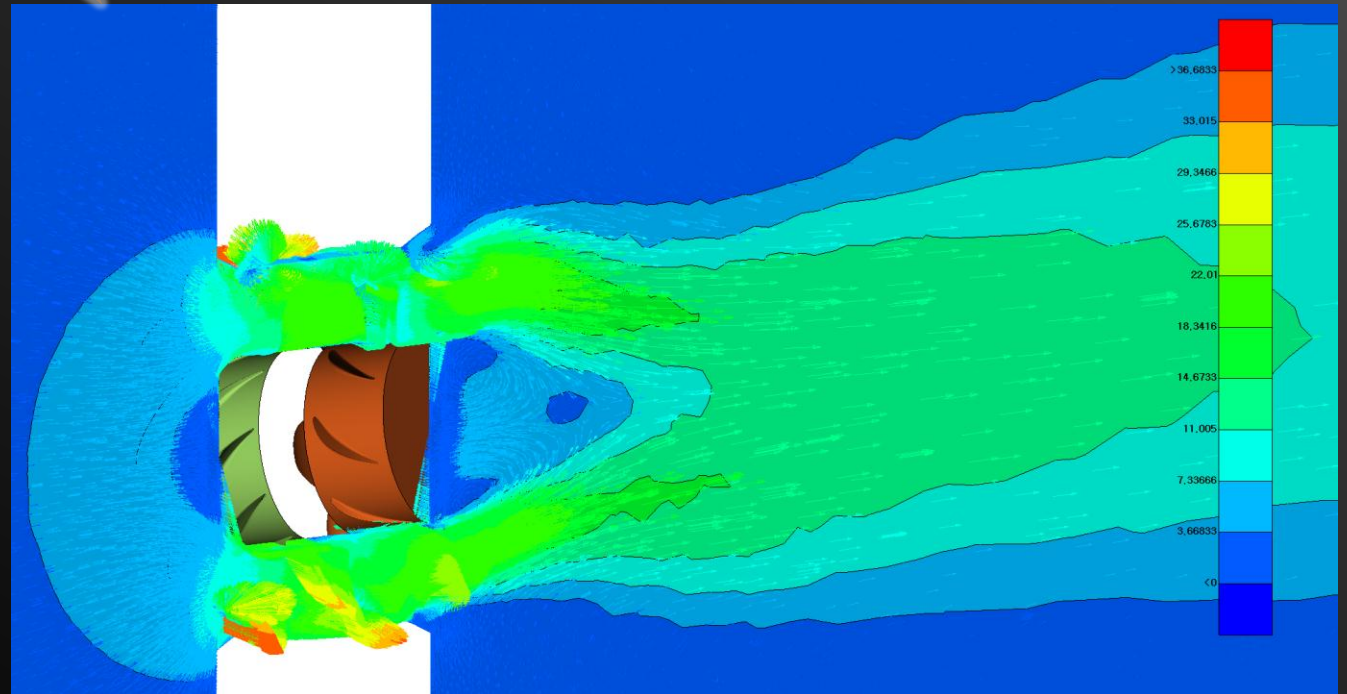
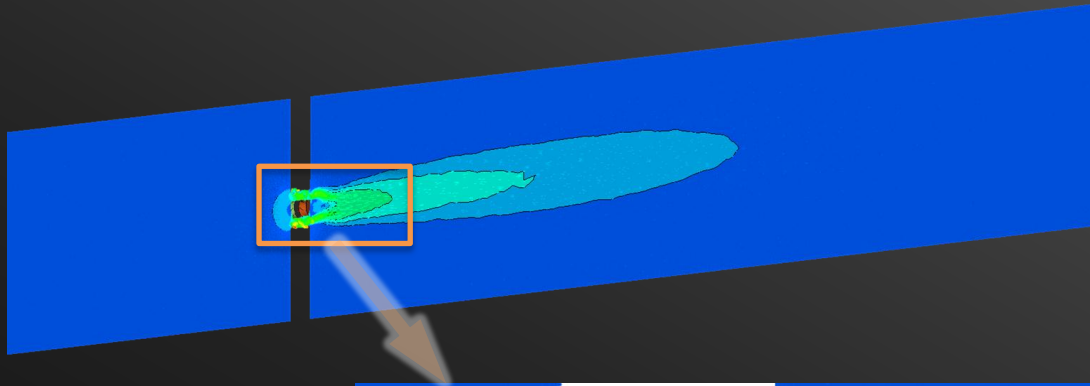
해석 자원 상세

- 운영체제 : Ubuntu 14.04 LTS
- 오픈폼버전 : **simpleNFOam**
OpenFOAM 2.4.0 .. 넥스트폼 개선 solver
(오픈폼 설치는 openfoamwiki 홈페이지 참고)
- 워크스테이션 :
 - Intel Xeon E5-2650 V2@ 2.6 GHz
- 계산 시간 :
 - 병렬 해석 8 CPUs (Hyper threading)
 - 메모리 사용량 < 24 GB
 - 수렴 도달 스텝 ~ 2500 step
 - 계산 시간 < 4일



정상상태 예측 유동장

유동 박리 예측 위한 상대속도 결과 처리 요구됨



요약 및 다음연구

- 오픈폼을 이용하여 동축반전 축류팬의 유동해석 수행함.
- 오픈폼으로 예측된 성능값은 풍동 실험과 유사한 경향성을 보임.
- 조금 불편한 점은 있지만, 근본적으로 상용코드에서 사용하는 전반적인 기능이 많이 구현됨.
- 상대 속도 결과 등 후처리를 보강하여 블레이드 형상 변경에 따른 효율 영향도 분석 예정임.