creative commons

COMMONS DEED

저작자표시-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학석사학위논문

Parametric Study of Two Rotor Cycloidal Blade System Using CFD Analysis for Cyclocopter Forward Flight

사이클로이달 블레이드 시스템의 유동해석을 활용한

사이클로콥터의 전진비행연구

2013 년 2 월

서울대학교 대학원

기계항공공학부

김 윤 성

ABSTRACT

The cyclocopter is an aircraft that is capable of vertical take-off and landing using a cycloidal blade system. The cycloidal blade system consists of variable pitch blades that rotate around an axis which lies parallel to the spanwise direction of the blades. The model cyclocopter developed by Seoul National University has four rotors and maximum pitch angle and phase angle of the blades in each rotor are individually controlled. Each rotor has four blades and the controlled maximum pitch and phase angles are related to the magnitude and the direction of thrust force, respectively.

In the modeled cyclocopter, two rotors are placed next to each other as a tandem type and the distance between the two is very close. Therefore, there is flow interference between the rotor during the hovering or the forward flight. It is necessary to understand the flow structure between those rotors to establish a control algorithm, which includes determining proper maximum pitch angle and phase angle for each flight situation. In this paper, CFD analysis is used for parametric study of two rotor cycloidal system. It was conducted with a transient method because the blades change its pitch angle periodically during a rotation. A two-dimensional finite volume mesh system contains front and rear rotors which is half of the modeled cyclocopter. Note here that only two rotors are needed for the two-dimensional CFD study since the other two rotors are aligned repetitively.

Commercial softwares, ANSYS Fluent and ANSYS CFD-Post, were used for flow solving and post processing. To change pitch angle of the blades during a rotation that has a sinusoidal profile, a user defined function (UDF) was integrated into the Fluent as well as a sliding-mesh method. Each blade and each rotor are meshed with circular rotating zone

that has a sliding interface with larger surrounding mesh system.

Result from the two rotor system is different from a single rotor system in terms of force magnitude, efficiency of the system, and flow structure such as induced flow direction. In the forward flight case, rear rotor feels quite different incoming flow than the front rotor. This behavior can cause a failure due to imbalanced force of rotors. The degree of imbalance varies with velocity, posture, and flying direction. In order to overcome the imbalanced force of rotors, the flight posture should be tilted to forward to release the rear rotor from the downwash effect. In addition to this, the different pitch and phase conditions must be applied on those two rotors to maintain a stable flight.

The major purpose of current analysis is to study the flows of the two-rotor cycloidal system and to obtain the dynamic characteristics of the rotors on various conditions in which velocity, pitch angle, phase angle, and posture are different. Hence, it is important to understand flow structures to predict flight stability and to build up control algorithm. The case studies will be followed to attain the most effective and efficient pitch angles and phase degrees for given velocity and posture conditions.

Keywords : Cycloidal blade system, cycloidal pitch motion, cyclocopter forward flight, CFD analysis

Student number : 2011-20696

TABLE OF CONTENTS

ABSTRACT ii
TABLE OF CONTENTS iv
LIST OF FIGURES v
LIST OF TABLES viii
1. INTRODUCTION 1
1.1 Cycloidal Blade System 2
1.2 History of Cyclocopter in SNU 3
1.3 Recent Developments of Cycloidal Blade System UAV 6
1.4 Current Work 7
2. SYSTEM MODELING FOR ANALYSIS 9
2.1 2D Mesh Modeling with Patran 9
2.2 Pre-Processing with Gambit 12
2.3 Pre-Processing on Fluent 14
2.4 Configuration of Simulation 15
3. CFD ANALYSIS 17
3.1 Single Rotor Analysis 17
3.2 Two Rotor Analysis - Level Flight 24
3.3 Tilted Two Rotor Analysis 29
3.4 Tilted Two Rotor Analysis - C.G. Shifted 36
4. CONCLUSIONS 43
REFERENCES 45
APEENDIX A. APPLYING PITCHING CONTROL MECHANISM TO UDF 46
APEENDIX B. PITCHING MOTION UDF 50

LIST OF FIGURES

Figure 1.1 Cycloidal flying machine (Kirsten, 1934)

Figure 1.2 Six-bladed cycloidal rotor (Bosch Aerospace, 1998)

Figure 1.3 Cycloidal blade system

Figure 1.4 1st and 2nd version of cyclocopter (2003)

Figure 1.5 Mini cyclocopter 1st version (2005)

Figure 1.6 Mini cyclocopter 2nd version (2006)

Figure 1.7 Mini cyclocopter 3rd version (2007)

Figure 1.8 Mini cyclocopter 4th version (2009)

Figure 1.9 D-Dalus propulsion system

Figure 1.10 Hover flight of cyclocopter in UMD

Figure 1.11 Mini cyclocopter 5th version

Figure 2.1 Divided cell zone

Figure 2.2 2D meshes for CFD analysis

(a) Blade Cell Zone

(b) Hub Cell Zone

(c) Outer cell zone

Figure 2.3 Discordance of Boundary Nodes

Figure 2.4 Blade pre-positioning

Figure 2.5 Boundary condition

Figure 2.6 Blade pitch angle for 25degree of maximum pitch

Figure 2.7 Conceptual diagram of simulation

Figure 3.1 Single rotor thrust

Figure 3.2 Single rotor velocity contour – No inlet velocity

Figure 3.3 Phase correction result

Figure 3.4 Front rotor velocity contour

(a) TSR 0.01 (b) TSR 0.10 (c) TSR 0.25 (d) TSR 0.30 (e) TSR 0.35 (f) TSR 0.40

Figure 3.5 Rear rotor velocity contour

(a) TSR 0.01 (b) TSR 0.10 (c) TSR 0.25 (d) TSR 0.30 (e) TSR 0.35 (f) TSR 0.40

Figure 3.6 Single blade generating lift for rotation angle
Figure 3.7 Rotor Efficiency
Figure 3.8 Velocity contours of TSR 0.01(0.31m/s) case
(a) Initial Case (b) Final Case
Figure 3.9 Velocity contours of TSR 0.05(1.56m/s) case
(a) Initial Case (b) Final Case
Figure 3.10 Velocity contours of TSR 0.10(3.11m/s) case
(a) Initial Case (b) Final Case
Figure 3.11 Adjusted pitch and phase angle
Figure 3.12 Transition of flight posture
Figure 3.13 Finite element model of tilted rotor analysis
Figure 3.14 Velocity contours of TSR 0.15(4.67m/s) case – C.G -0.225m
(a) 10deg (b) 20deg (c) 30deg (d) 45deg
Figure 3.15 Velocity contours of TSR 0.20(6.22m/s) case – C.G -0.225m
(a) 10deg (b) 20deg (c) 30deg (d) 45deg
Figure 3.16 Velocity contours of TSR 0.25(7.78m/s) case – C.G -0.225m
(a) 10deg (b) 20deg (c) 30deg (d) 45deg
Figure 3.17 Velocity contours of TSR 0.30(9.33m/s) case – C.G -0.225m
(a) 10deg (b) 20deg (c) 30deg (d) 45deg
Figure 3.18 Velocity contours of TSR 0.35(10.89m/s) case – C.G -0.225m
(a) 10deg (b) 20deg (c) 30deg (d) 45deg
Figure 3.19 Velocity contours of TSR 0.40(12.44m/s) case – C.G -0.225m
(a) 10deg (b) 20deg (c) 30deg (d) 45deg
Figure 3.20 Thrust distribution on rotors – C.G -225m Thrust distribution
(a) TSR 0.15(4.67m/s) (b) TSR 0.20(6.22m/s)
(c) TSR 0.25(7.78m/s) (d) TSR 0.30(9.33m/s)
(e) TSR 0.35(10.89m/s) (f) TSR 0.40(12.44m/s)
Figure 3.21 Pitching moment for initial cases
Figure 3.22 Diagram of tilted flight posture
Figure 3.23 Required thrust difference between rotors for dissipation of pitching moment
Figure 3.24 Velocity contours of TSR 0.15(4.67m/s) case – C.G zero

(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.25 Velocity contours of TSR 0.20(6.22m/s) case – C.G zero

(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.26 Velocity contours of TSR 0.25(7.78m/s) case – C.G zero

(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.27 Velocity contours of TSR 0.30(9.33m/s) case – C.G zero

(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.28 Velocity contours of TSR 0.35(10.89m/s) case – C.G zero

(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.29 Velocity contours of TSR 0.40(12.44m/s) case – C.G zero

(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.30 Thrust distribution on rotors – C.G zero

(a) TSR 0.15(4.67m/s) (b) TSR 0.20(6.22m/s)

(c) TSR 0.25(7.78m/s) (d) TSR 0.30(9.33m/s)

(e) TSR 0.35(10.89m/s) (f) TSR 0.40(12.44m/s)

Figure 3.31 Rotor thrust efficiency difference

Figure 3.32 Phase and pitch angle correction result

Figure 3.33 Cyclocopter Power Required for Forward Flight

Figure A.1 Mechanical model of pitching mechanism

Figure A.2 Schematic diagram of pitching mechanism

(a) Zero maximum pitch angles

(b) Maximum pitch angle of θ_{max}

(c) Pitch angle of θ at rotation angle of φ on quadrant I or IV

(d) Pitch angle of θ at rotation angle of φ on quadrant II or III

LIST OF TABLES

Table 1.1 Mini cyclocopter fifth version design variables
Table 3.1 Initial TSR 0.01(0.31m/s) case condition and result
Table 3.2 Final TSR 0.01(0.31m/s) case condition and result
Table 3.3 Initial TSR 0.05(1.56m/s) case condition and result
Table 3.4 Final TSR 0.05(1.56m/s) case condition and result
Table 3.5 Initial TSR 0.10(3.11m/s) case condition and result
Table 3.6 Final TSR 0.10(3.11m/s) case condition and result
Table 3.7 Initial 10deg tilted TSR 0.15(4.67m/s) case condition and result
Table 3.8 Initial 45deg tilted TSR 0.15(4.67m/s) case condition and result

1. INTRODUCTION

The cycloidal blade system has various advantages for using on UAV. There were numerous studies about it, yet there is no remarkable result about cycloidal blade system UAV until these days. The UAV with cycloidal blade system was introduced by Kirsten at the University of Washington in 1920's [1]. However, Kirsten's works could not come to fruition due to the funding matters. Around the same time, Wheatley at the NACA Langley Research Center developed cyclogiro rotating wing [2]. He also couldn't reach the successful result due to the differences of theory and experiment. More recently, McNabb developed a computer model for cycloidal propeller and compared thrust and power on the rotor from experimental results constructed by Bosch Aerospace [3]. This work also could not deliver complete result.

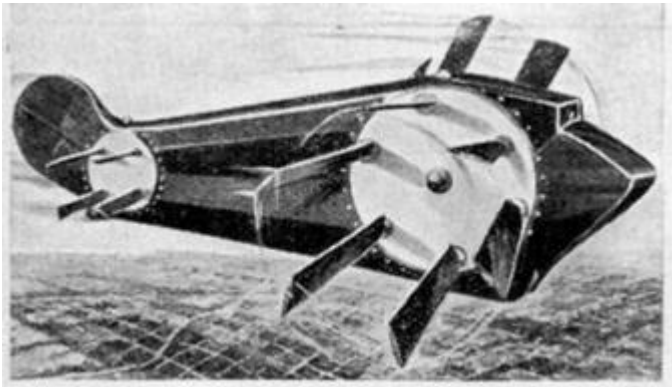


Figure 1.1 Cycloidal flying machine (Kirsten, 1934)

However, there are few noticeable results in these days including Maryland University and D-Dalus in Austria. Both groups have developed practical model working on stable hover

flight. The cycloidal blade system receives attention again in these days as an effective VTOL (vertical take-off and landing) propulsion technology. The research in this paper could show the possibility of a cyclocopter as an alternative of VTOL UAV.

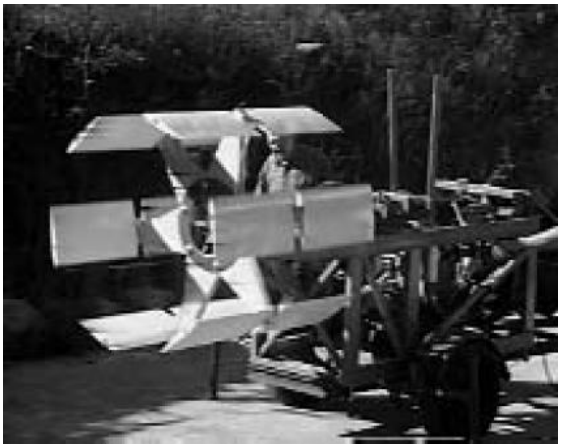


Figure 1.2 Six-bladed cycloidal rotor (Bosch Aerospace, 1998)

1.1 Cycloidal Blade System

The cycloidal rotor consists of several blades rotating about a horizontal axis. These blades vary its pitch angle periodically during the rotation. The cycloidal blade system changes the direction and magnitude of thrust by adjusting phase angle and maximum pitch angle. The process of changing phase angle and maximum pitch angle is can be conducted simply shifting the eccentricity of the control point. Therefore cycloidal blade system can provide high maneuverability by instantaneous change of thrust. Another advantage of a cycloidal blade system is low noise. Since the cycloidal rotors require lower rotation speed compare to conventional helicopter rotors and also the blade tip

speed is much lower than the other one.

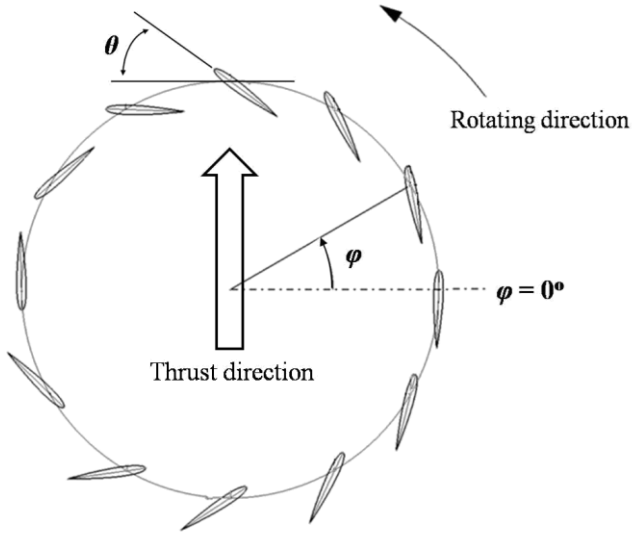


Figure 1.3 Cycloidal blade system

Figure 1.3 shows the cycloidal blade system phase angle(φ) of which is zero. The blade pitch angle(θ) varies to maximum value at +y(90°) and –y(270°) axis positions. This maximum pitch angle position is tilted as phase angle is applied. The thrust direction of a rotor is supposed to be vertical upward when the phase angle is zero. However the thrust vector is slightly tilted to the rotating direction due to the distribution of induced velocity [4].

1.2 History of Cyclocopter in SNU

The first two versions of cyclocopter were large which have 1.5m blade span and about 20 horsepower. The first mini version of cyclocopter is a small one with simple design shown on the Figure.1.5.

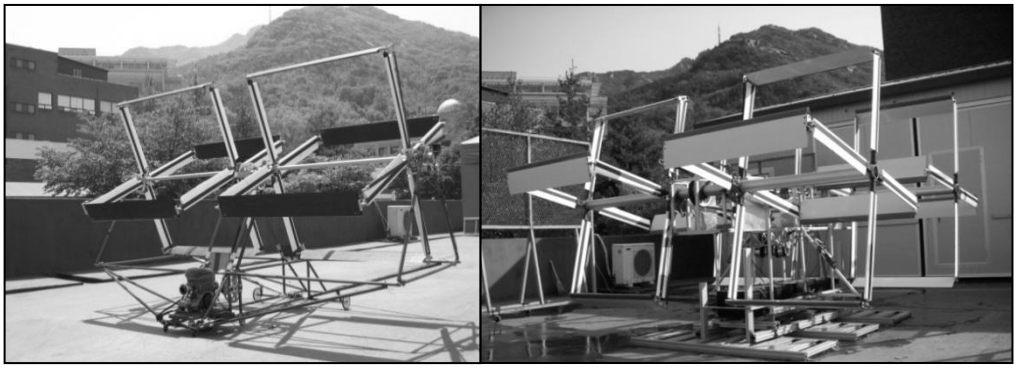


Figure 1.4 1st and 2nd version of cyclocopter (2003)

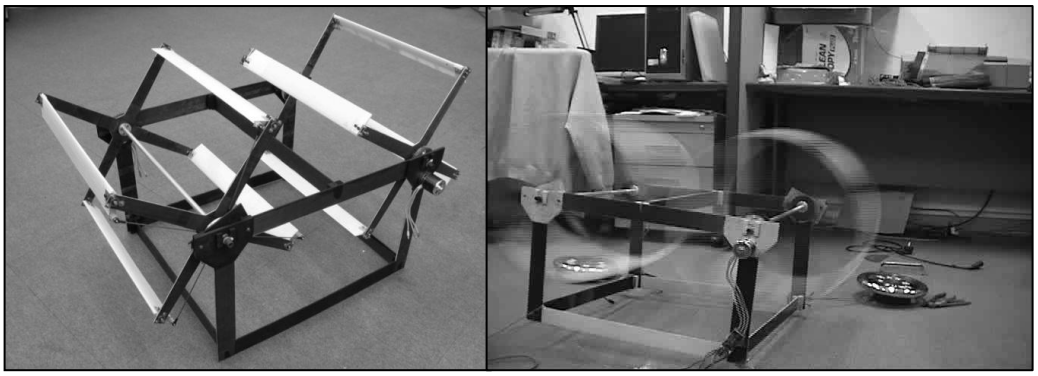


Figure 1.5 Mini cyclocopter 1st version (2005)

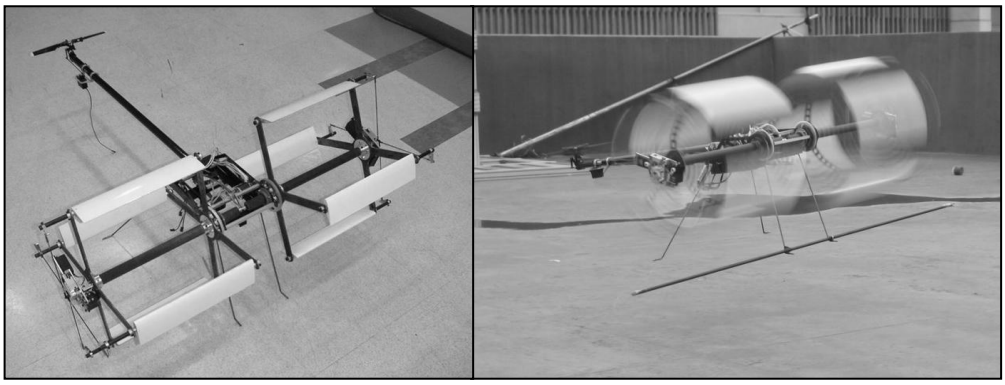


Figure 1.6 Mini cyclocopter 2nd version (2006)

The second version of mini cyclocopter adopted two rotor blade system with one tail rotor as shown on the Figure. 1.6. The third version is quadrotor cyclocopter which consists of four elliptic blades for each rotor with more complicated design. The fourth version of mini cyclocopter is an upgraded model of the previous one. It has two pairs of rotors which are powered by 4 brushless motors and it takes swash plate mechanism to control maximum pitch and phase angle.

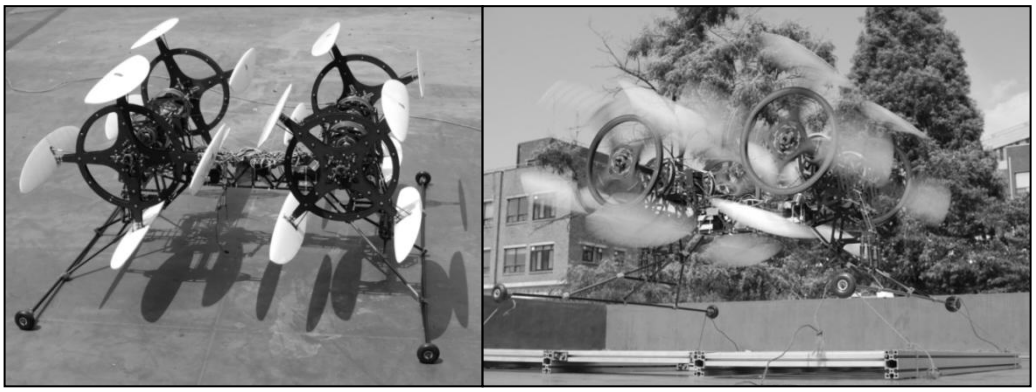


Figure 1.7 Mini cyclocopter 3rd version (2007)

Figure 1.8 Mini cyclocopter 4th version (2009)

1.3 Recent Developments of Cycloidal Blade System UAV

There are some recent achievements using cycloidal blade system. The D-Dalus is one of them. The current model is a prototype using 120bhp engine with 70kg payload. The propulsion consists of four disk-type rotors. The thrust is controlled by maximum pitch angle while rpm of each rotor are the same. [5].

Figure 1.9 D-Dalus propulsion system

Chopra at the University of Maryland has been working on a micro-scale cyclocopter and currently developed model weighs about 0.5kg [6]. The recent version of cyclocopter at UMD has two cycloidal rotors with single a conventional rotor for pitching control. The thrust is controlled solely by rotational speed of cycloidal rotors. Experimental and computational studies have been conducted for hovering and forward flights.

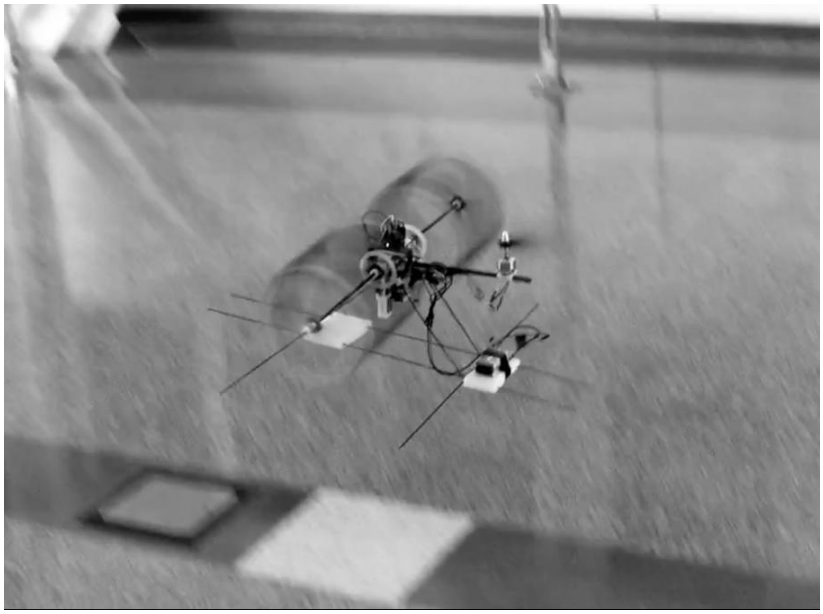


Figure 1.10 Hover flight of cyclocopter in UMD [7]

1.4 Current Work

The fifth version of mini cyclocopter is mainly referred in this paper which has a few improvements from the fourth one. The pitch angle control system is replaced with a new mechanical device and the rotor hub is made by semi monocoque method. Also, the four brushless motors are replaced by a single motor that is four times stronger. The fifth version performs a stable hover flight and forward flight experiments are currently being conducted.

The actual model weighs 12.8kg and additional 1.5kg can be added for any mission equipment such as a camera. Therefore total required thrust is determined as 14.3kg, or 140N. Since the analysis model is half of the actual model, the required thrust is determined as 70N in this paper. The following analysis is based on this specification.

Figure 1.11 Mini cyclocopter fifth version

Table 1. 1 Mini cyclocopter fifth version design variables

Rotating speed	1100 rpm
Radius of rotor	0.27 m
Distance between rotors	0.7 m
Span of blade	0.5 m
Airfoil	NACA 0018
Weight	12.8 kg
Total thrust	16.32 kgf

2. SYSTEM MODELING FOR ANALYSIS

The purpose of this analysis is to understand flow features during forward flight. It would be useful to generate a control algorithm for forward maneuver based on the analysis results. Also the goal of this analysis is to find stable flight conditions for different vehicle velocity conditions by controlling maximum pitch and phase angles. The stable flight condition determined in this paper is that horizontal net force and pitching moment of the entire model is close to zero. Also vertical net force of both rotors is around 70N assuming the cyclocopter maintains a constant speed. The inlet speed as boundary condition is set to be identical to the flight speed which is defined by TSR(tip speed ratio). The analysis is conducted with a transient method since the rotor blade pitch angle varies periodically during each rotation. The sliding mesh method and user defined function of ANSYS Fluent is used to realize the varying pitch motion. The finite element model consists of two-dimensional and the k-ε viscous model is used as a turbulence model.

2.1 2D Mesh Modeling with Patran

The pre-processing is required before performing a CFD analysis. The process of creating geometry and element is conducted with Patran which provides various and convenient methods for creating mesh. The entire domain is divided with three different cell zone as shown on the Figure 2.1; outer, hub, and blade.

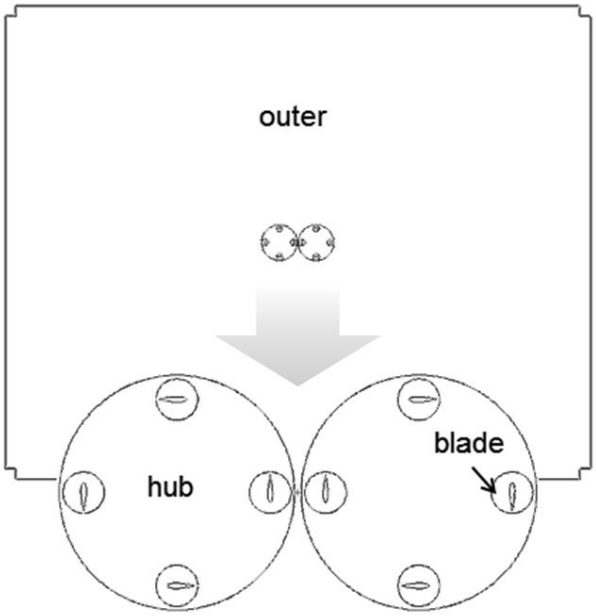
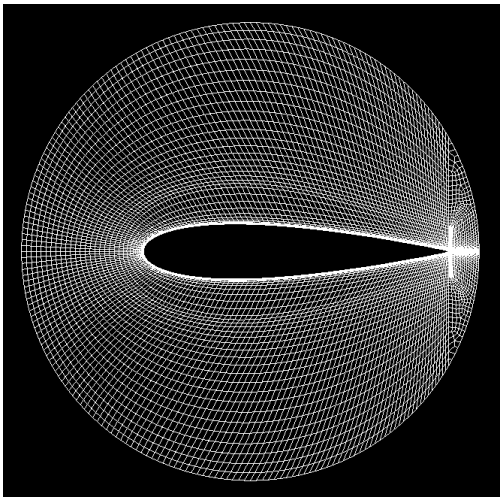
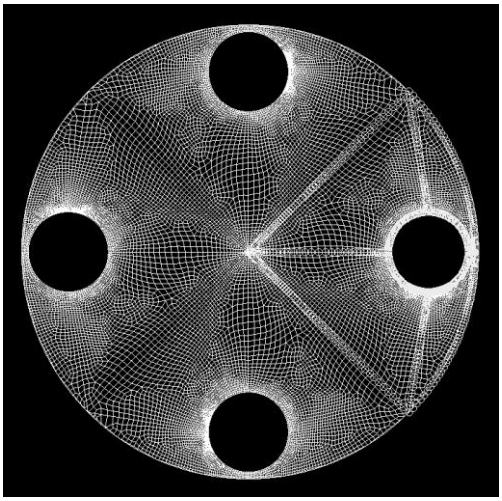


Figure 2.1 Divided cell zone

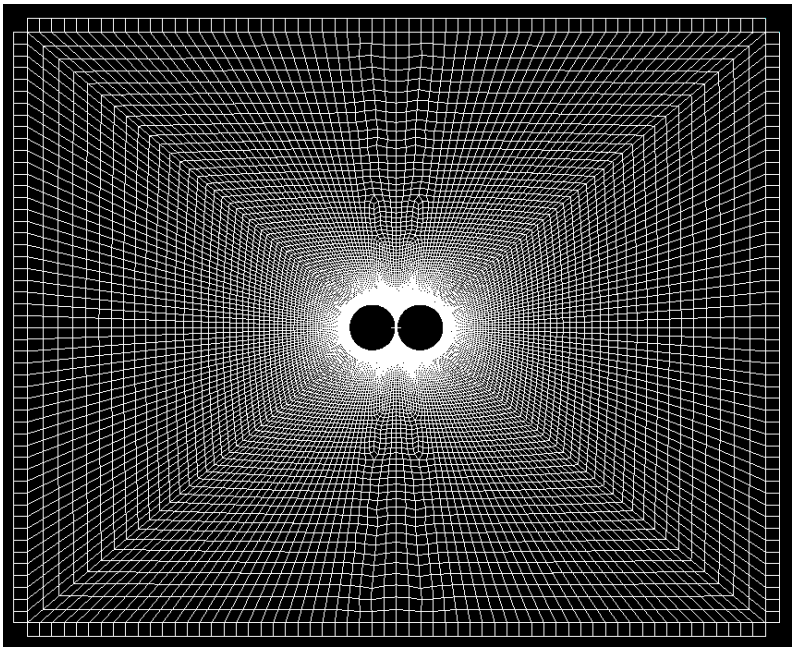
There are four circular cell zones for blade domain inside the hub cell zone. The area near the blade wall and the trailing edge consists of relatively denser meshes. The hub cell zone also has dense mesh near the blade cell zone especially near leading and trailing edge. The outer cell zone is the rest of domain and there are coarse meshes since the velocity gradient is lower. The number of elements for the blade, hub, outer cell zones are 9,068, 16,656, and 26,072, respectively. The entire model consists of 131,928 elements. All boundary nodes of the cell zones do not correspond to adjacent cell zone. Because each cell zone has its own boundary which will be defined as the interface zone for sliding mesh method. The sliding mesh method does not require aligned element nodes since the interface zones exchange flow data of the adjacent elements by itself.



(a) Blade Cell Zone



(b) Hub Cell Zone



(c) Outer cell zone

Figure 2.2 2D meshes for CFD analysis

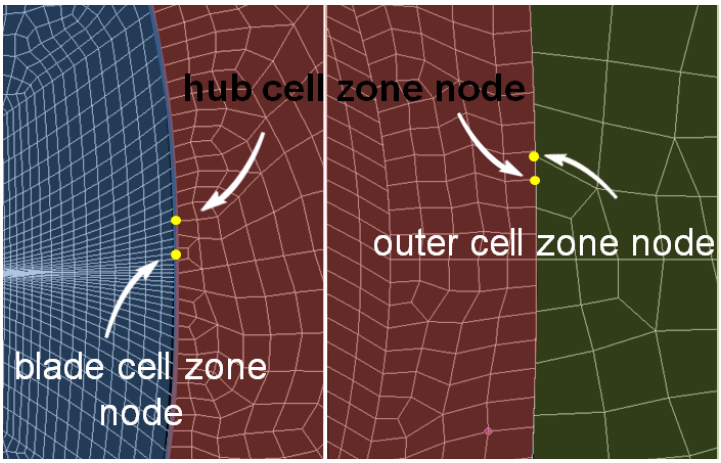


Figure 2.3 Discordance of Boundary Nodes

2.2 Pre-Processing with Gambit

Gambit provides convenient methods for defining zones and creating mesh files for ANSYS Fluent. In this step, inlet, outlet boundaries, cell zones and interfaces are defined. Adjacent zone interfaces are named similarly to define a coupled interface for sliding mesh method. For instance, outside boundary of the hub cell zone and inside boundary of the outer cell zone are defined as a coupled interface.

Blades are positioned in this step. The blade and hub meshes are generated with a mirroring method, and therefore, both front and rear rotors share the same position. In the experiment, however, the two rotors rotate with 45 degrees different blade position angle as shown in Figure 2.4. Operating conditions of phase and pitch angles for the considered cases are applied simultaneously.

The inlet boundary on front rotor side is defined as velocity inlet for the forward flight condition. The upper boundary is defined as pressure inlet and rest boundaries are defined

as pressure outlet. Those boundary condition options (velocity inlet, pressure inlet/outlet) are available in ANSYS Fluent. All of these geometrical mesh data including boundary and cell zone information are exported as a mesh file for ANSYS Fluent.

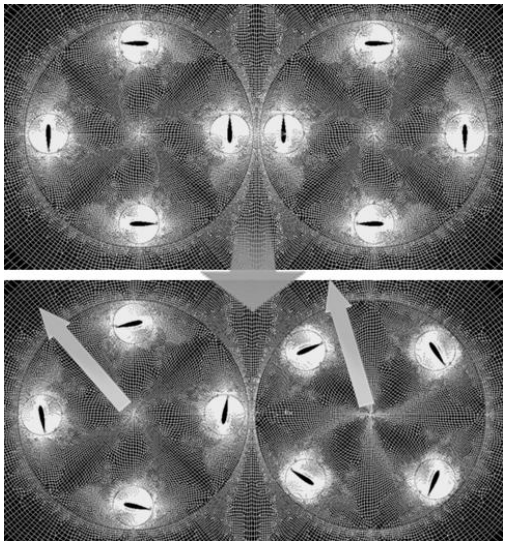


Figure 2.4 Blade pre-positioning

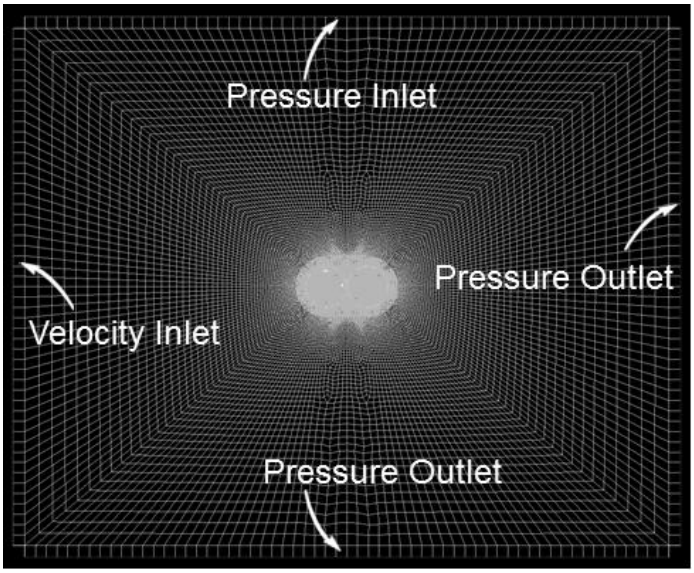


Figure 2.5 Boundary condition

2.3 Pre-Processing on Fluent

Before conducting the analysis, additional conditions should be applied. Since pitch angle changes during the cyclic rotation, it is necessary to use UDF(user defined function) to simulate a cycloidal pitch motion. Theoretically, the cycloidal motion is based on a sinusoidal function. However, the blade motion of actual model is slightly deviated from the sinusoidal profile due to the mechanism of pitching control device.

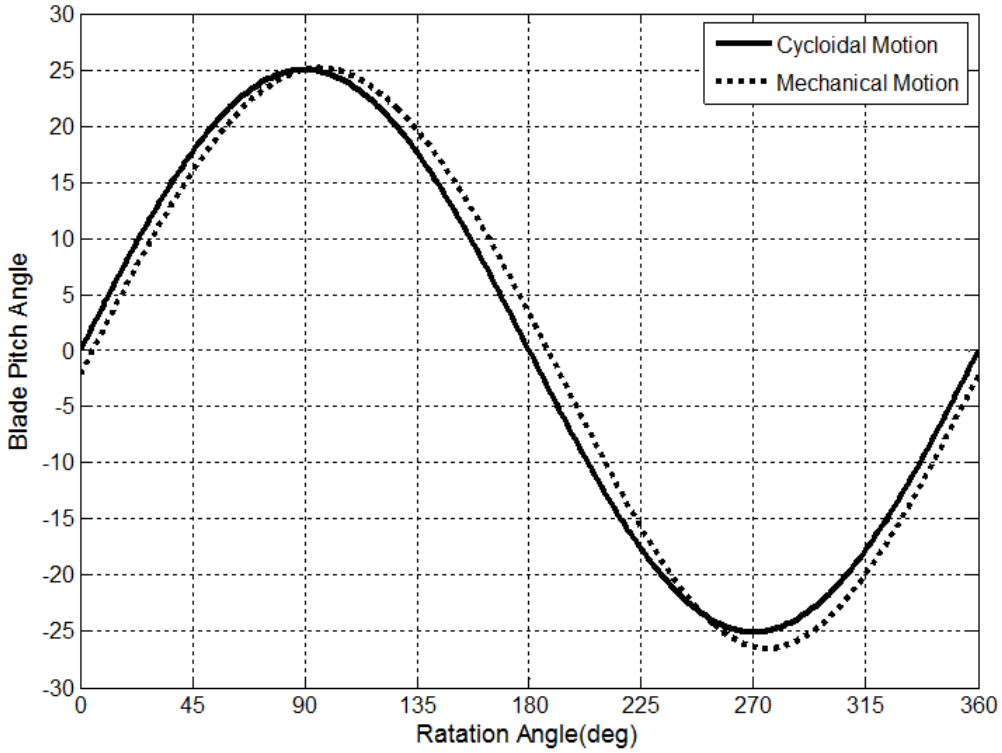


Figure 2.6 Blade pitch angle for 25degree of maximum pitch

Figure 2.6 shows difference between mechanical motion and sinusoidal function when the maximum pitch angle is 25°. The UDF for pitching motion is written base on this

mechanical motion instead sinusoidal function in order to minimize errors. Pitch angle of the blade is roughly -26° at 270°(-y axis) for the resultant mechanical motion while the sinusoidal one is -25°. This differential is not negligible since substantial lift force is generated at around 270° location, which is the lowest part of hub [4]. The UDF source code is longer than that of the sinusoidal function case since the pitch angle should be calculated from geometrical relationship of control rod and control point. Although the source code becomes longer and complicated, the affection on calculation time is negligible.

There are various inlet velocity conditions considered in this paper based on TSR 0.01 to 0.4 which is 0.311m/s to 12.44m/s since TSR is the convenient parameter to classify incoming flow [8].

2.4 Configuration of Simulation

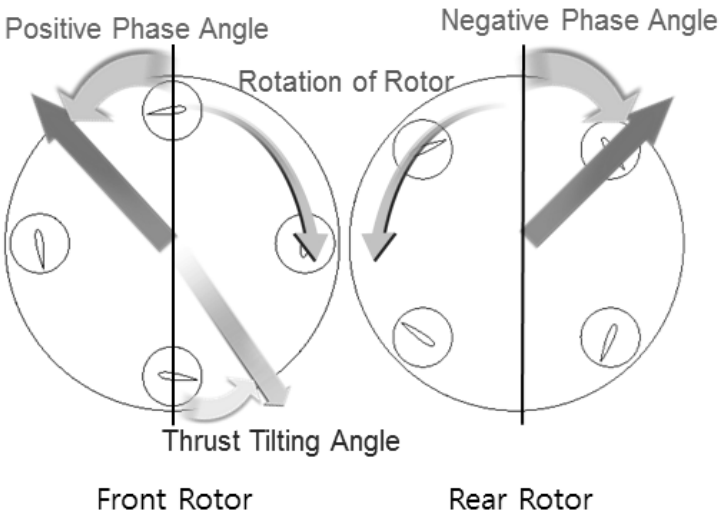


Figure 2.7 Conceptual diagram of simulation

The front rotor rotates in clock-wise while the rear rotor rotates in the other way. Theoretically when the phase angle is zero, it indicates that there is no horizontal component of a thrust vector while vertical force may exist. When the thrust vector is needed to tilt to positive angle for forward maneuver, the overall phase angle becomes positive. Also, the positive thrust angle means here that thrust angle is tilted in counter clock-wise, i.e. from vertical to right lower direction in Figure 2.7.

3. CFD ANALYSIS

The analysis process consists of the number of steps. Before conducting a two rotor case, a single rotor case is analyzed to understand flow physics of a rotating blade for different phase and pitch angles. This process is important to establish phase and pitch angle condition of initial case for two rotor analysis. Phase condition of initial case for two rotor analysis is based on the single rotor result. Two rotor analysis is conducted for low velocities; TSR 0.01, 0.05 and 0.1. Tilted two rotor cases and C.G(center of gravity) shifted cases will be followed. These cases are conducted for higher velocity; TSR 0.15 to 0.4.

All cases are calculated at least 20 rotations to guarantee numerical convergence. It takes about 15 hours for each case using a single CPU. Results averaged over five revolves are presented in this section.

3.1 Single Rotor Analysis

Single Rotor Analysis is conducted separately for the front and the rear rotor. Fig. 3.1 shows how thrust changes with maximum pitch angle. There is a remarkable thrust variation when the maximum pitch angle increases from 20 to 25 degrees. Figure 3.2 is the velocity contour of 25 degree pitch angle case. The induced flow and the thrust vector are slightly tilted as shown on the Figure 3.2 even though there is no inlet velocity. The goal of this analysis is to find a stable flight condition. As stated above, horizontal component of the thrust and pitching moments should be close to zero. Therefore phase angle is

adjusted to generate truly vertical thrust for each velocity condition.

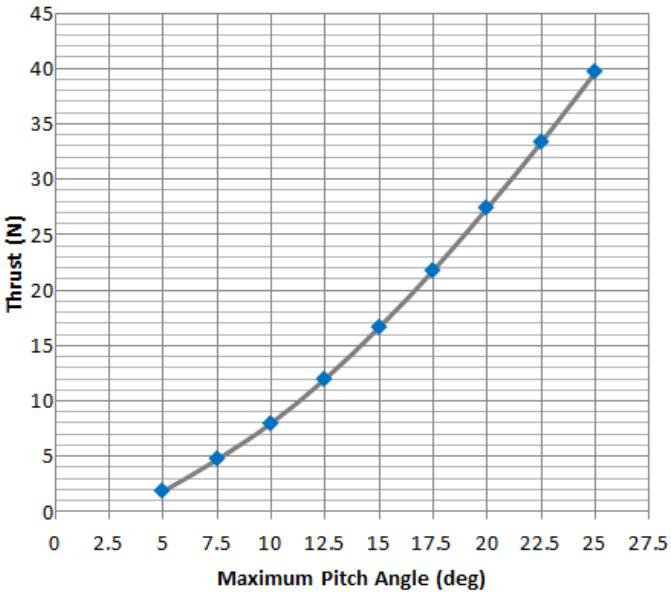


Figure 3.1 Single rotor thrust

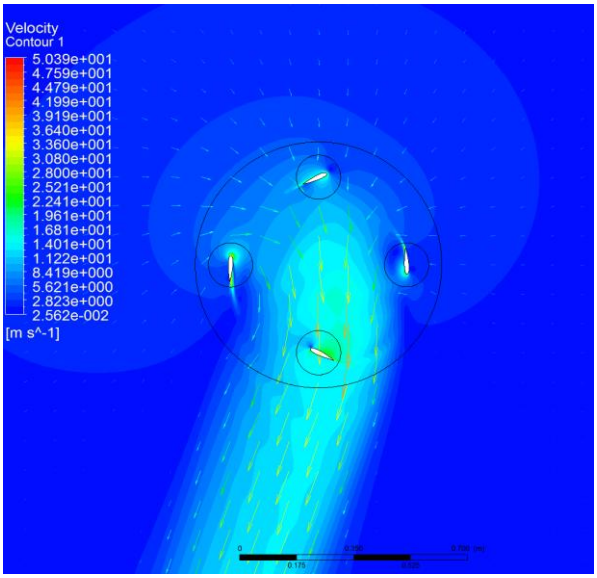


Figure 3.2 Single rotor velocity contour – No inlet velocity

The Figure 3.3 shows phase correction results for single rotor analysis. Following velocity contours on the Figure 3.4 and 3.5 are the result of this phase correction value. Those case results create thrust to solely vertical direction although the flows are tilted. This result is useful to set initial case condition for two rotor analysis.

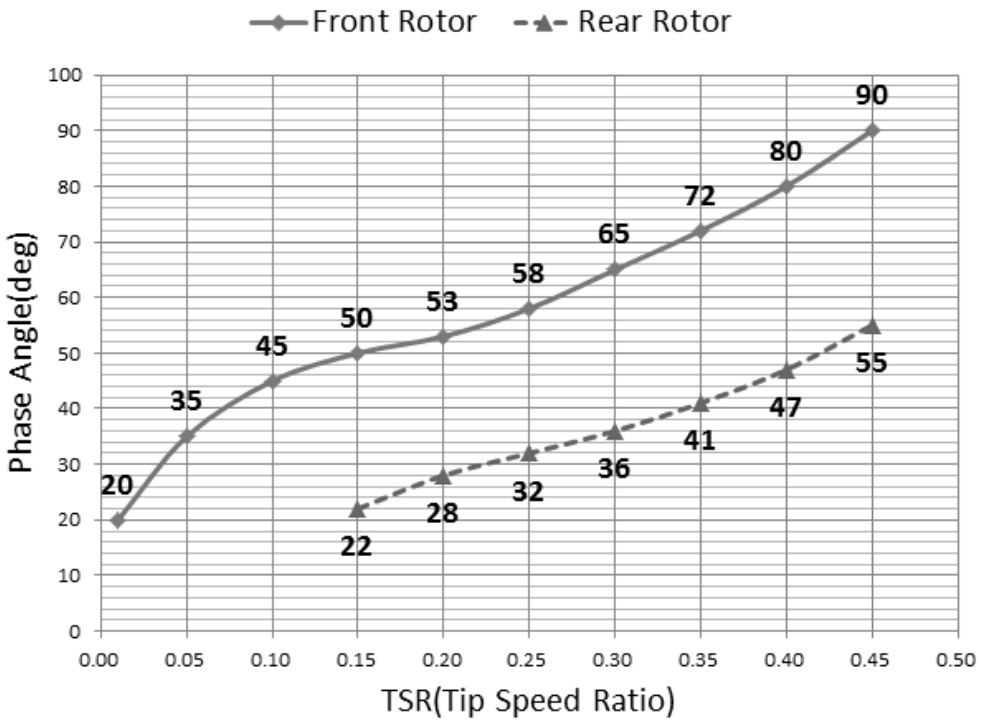
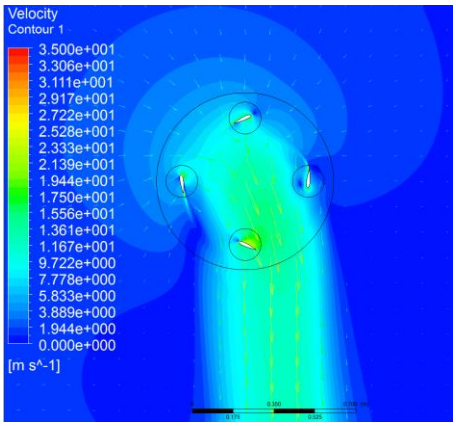
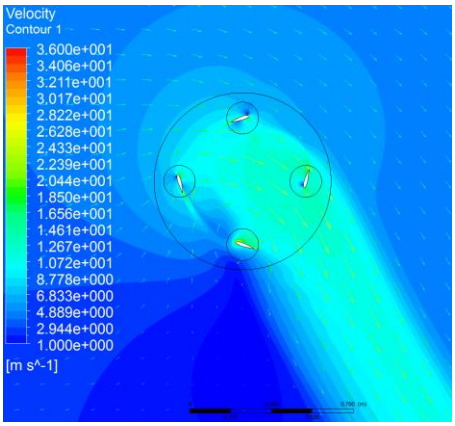


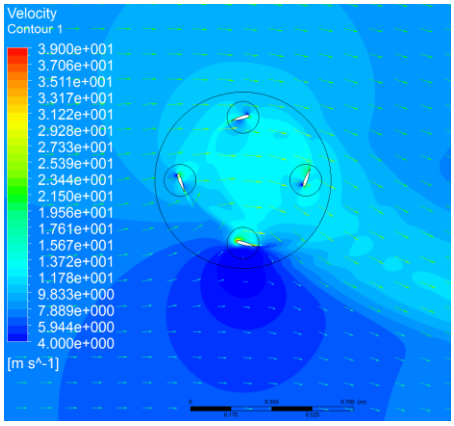
Figure 3.3 Phase correction result



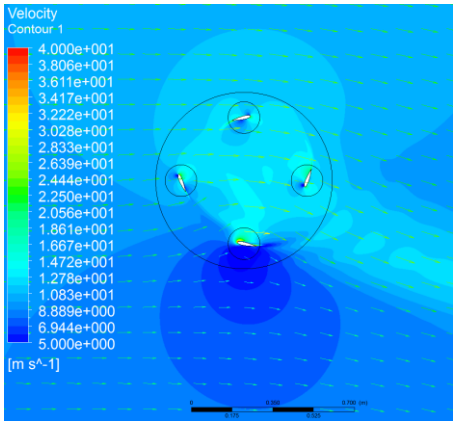
(a) TSR 0.01



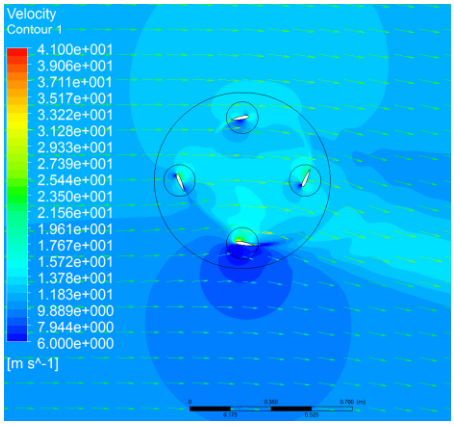
(b) TSR 0.10



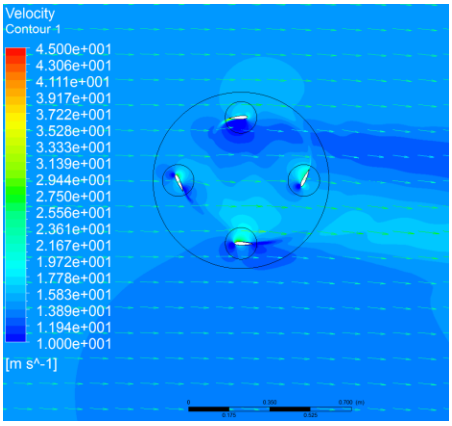
(c) TSR 0.25



(d) TSR 0.30

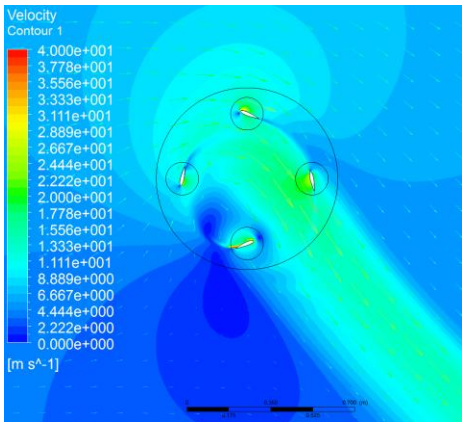


(e) TSR 0.35

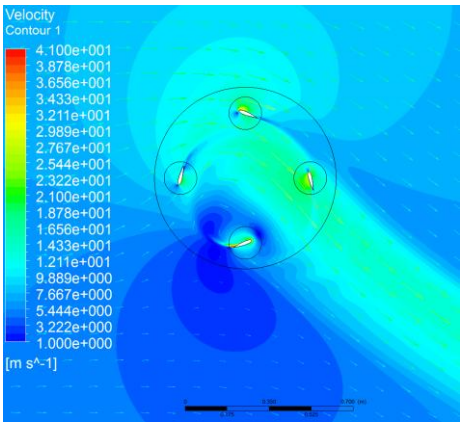


(f) TSR 0.45

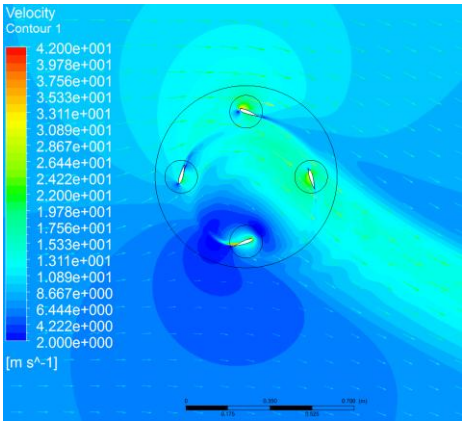
Figure 3.4 Front rotor velocity contour



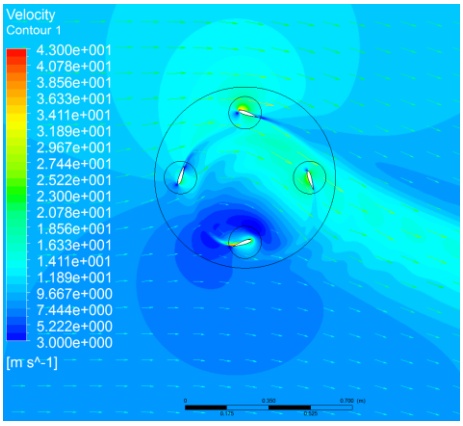
(a) TSR 0.15



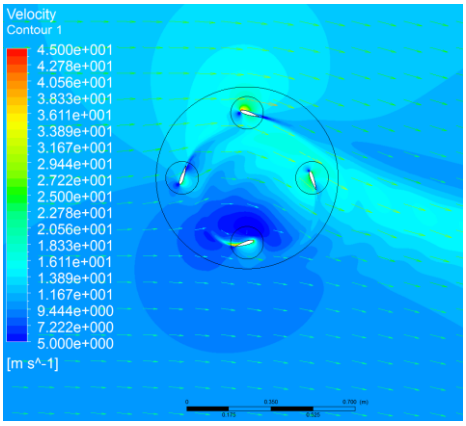
(b) TSR 0.20



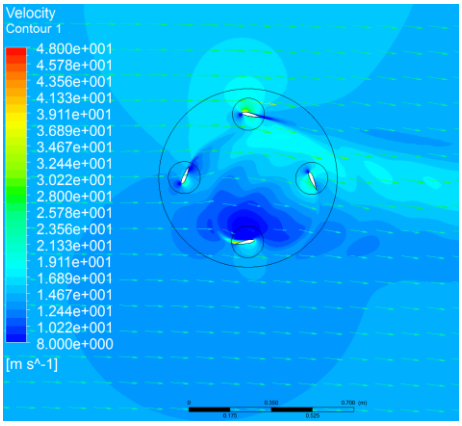
(c) TSR 0.25



(d) TSR 0.30



(e) TSR 0.35



(f) TSR 0.45

Figure 3.5 Rear rotor velocity contour

Major difference between front and rear rotor is the flow structure. In case of front rotor, the recirculation zone is generated right below the rotor. It indicates that the pressure difference is large in that area and thus, large portion of the lift force would be created when the blade passes through the lower point, or 270° degrees of rotation angle(-y axis). In fact, the front rotor generates most of the lift when the blade passes through that area as shown on the Figure 3.6. This result also can be explained by virtual camber effect [9]. The figure 3.6 shows lift generated from each of the two rotors for TSR 0.25 case. The rear rotor creates more lift when the blade is in the upper side, on the contrary to the front rotor. That is because the rear rotor creates a recirculation zone inside the rotor track due to the sense of rotation. Since the speed of flow heading blade is relatively faster on the upper side of the rear rotor, it creates more lift when the blade is located at there. This fact indicates that the rear rotor should take enough flow on the upper side of the rotor to create sufficient lift.

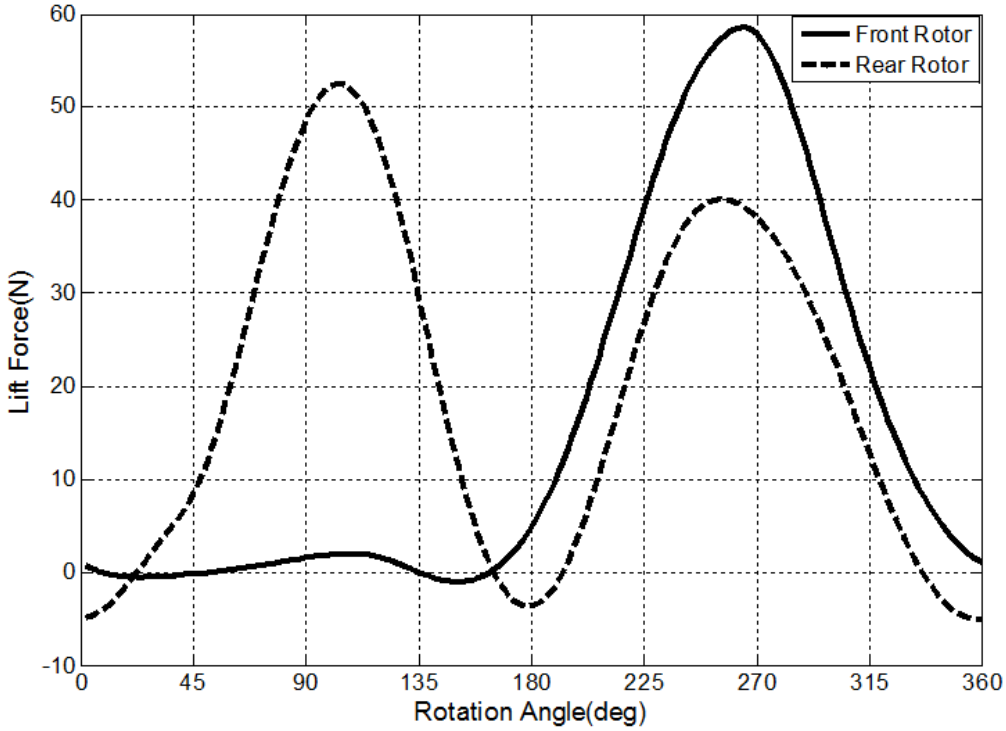


Figure 3.6 Single blade generating lift for rotation angle

In addition, it begins to show more complicated flow features as velocity increases for both rotors. In the TSR 0.45 cases, induced flow becomes highly fluctuative and shows chopped velocity contours while lower velocity cases show nearly uniform and smooth flow structures. Figure 3.7 shows generated thrust per required power for both rotors and the efficiency begins to decrease as the velocity goes up to TSR range of 0.3~0.4. It is also noticeable that the efficiency of the front rotor is higher than the rear rotor. Therefore it is important to maximize the performance of the front rotor.

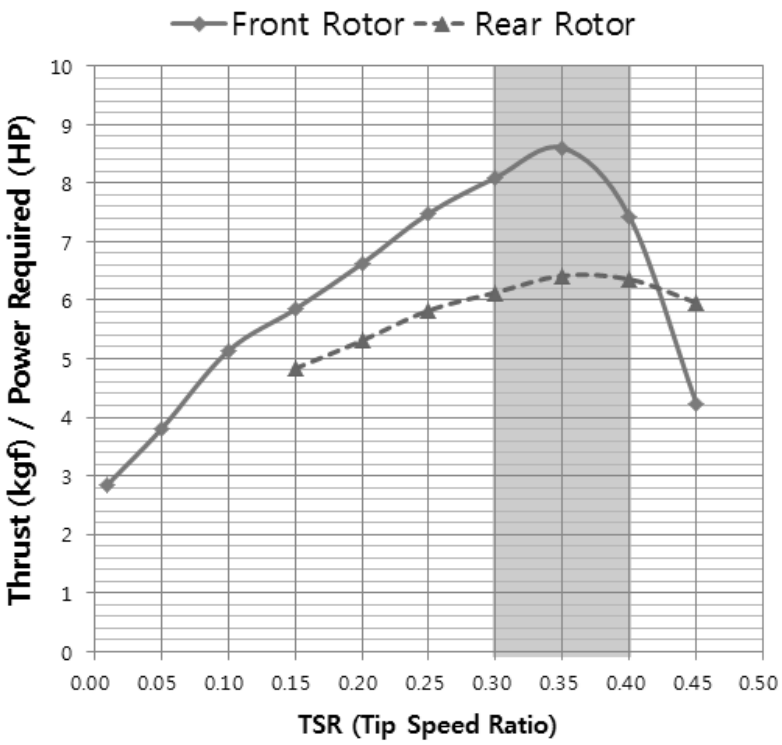


Figure 3.7 Rotor Efficiency

3.2 Two Rotor Analysis – Level Flight

The phase angle condition of initial case is obtained from single rotor analysis. The maximum pitch angle of all initial cases is set to 25 degrees. The phase angle for the rear rotor at TSR 0.01, 0.05 and 0.1 is set to zero since it would differ from the single rotor analysis due to the front rotor effect. The flow came from the front rotor is titled sharply at lower velocity cases as shown on the Figure 3.4. Therefore the flow feature heading for rear rotor is quite different from the single rotor case.

Table 3.1 Initial TSR 0.01(0.31m/s) case condition and result

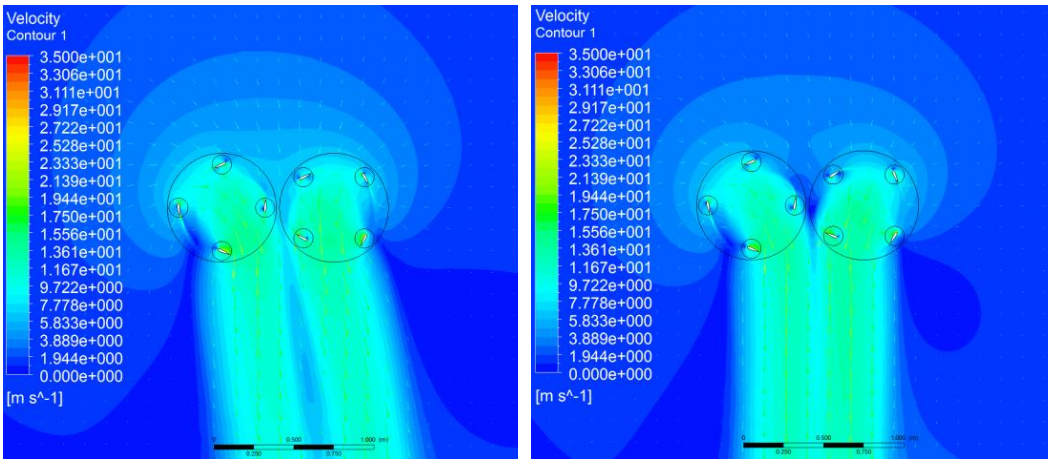
	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	20°	25°	17.55 N	39.01 N	
Rear	0°	25°	-27.78 N	29.59 N	
Total			**-10.23 N**	**68.60 N**	**-1.23 N·m**

Table 3.1 is pitch and phase angle condition and force results for TSR 0.01 case. The force is calculated by summing pressure and viscous forces on the blade wall. This force is calculated at every time step and averaged over one rotation. The presented results are averaged for five rotations. The pitching moment is calculated on the center of gravity which is 0.225m below the mid-point of two rotors. In addition, the power consumption of vehicle is calculated using the moments calculated at the center of each rotor.

The result shows the horizontal component of thrust -10.23N which would cause the acceleration to forward direction. The thrust should be minimized to find steady and stable condition. Following table is the result of final case after several case studies which minimize the horizontal component and pitching moment.

Table 3.2 Final TSR 0.01(0.31m/s) case condition and result

	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	31°	24°	3.14 N	36.12 N	
Rear	19°	25°	-2.57 N	36.05 N	
Total			**0.56 N**	**72.47 N**	**-0.69 N·m**



(a) Initial case (b) Final Case

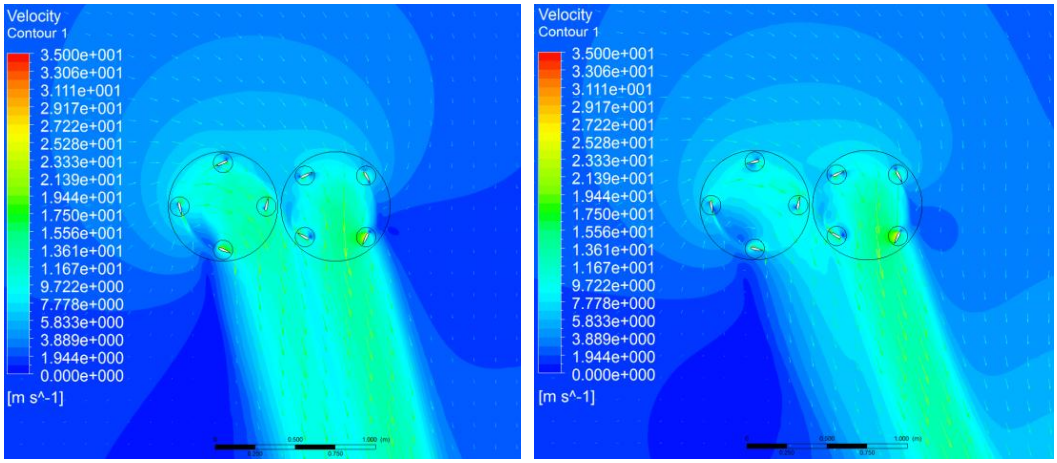
Figure 3.8 Velocity contours of TSR 0.01(0.31m/s) case

Table 3.3 Initial TSR 0.05(1.56m/s) case condition and result

	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	35°	25°	9.57 N	47.15 N	
Rear	0°	25°	-10.67 N	32.36 N	
Total			**-1.10 N**	**79.51 N**	**-6.20 N·m**

Table 3. 4 Final TSR 0.05(1.56m/s) case condition and result

	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	46°	20°	4.84 N	32.14 N	
Rear	2°	25°	-5.22 N	34.89 N	
Total			**-0.37 N**	**67.04 N**	**-1.81 N·m**



(a) Initial case (b) Final Case

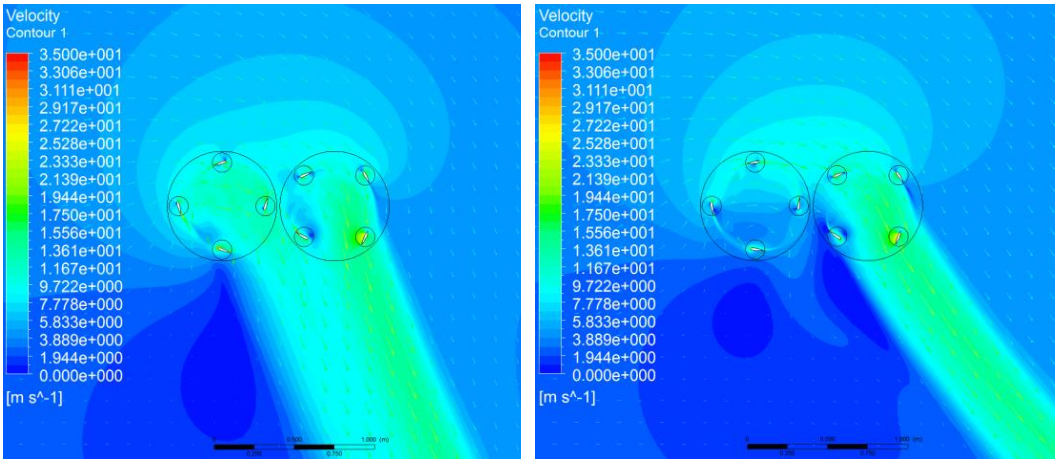
Figure 3.9 Velocity contours of TSR 0.05(1.56m/s) case

Table 3.5 Initial TSR 0.10(3.11m/s) case condition and result

	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	45°	25°	1.03 N	53.94 N	
Rear	0°	25°	-9.83 N	40.67 N	
Total			**10.86 N**	**94.60 N**	**-8.71 N·m**

Table 3. 6 Final TSR 0.10(3.11m/s) case condition and result

	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	51°	12°	3.63 N	31.82 N	
Rear	15°	25°	-5.38 N	40.09 N	
Total			**-1.75 N**	**71.91 N**	**-0.61 N·m**



(a) Initial case (b) Final Case

Figure 3.10 Velocity contours of TSR 0.10(3.11m/s) case

The front rotor provides more thrust than the rear rotor as flight speed goes up since the airstream velocity heading front rotor also increases. At the same time, upper side of the rear rotor could not take enough airstream since flow came from the front rotor is tilted to downward. This affect to rear rotor cannot generate sufficient lift. Hence, there is inevitable thrust loss on the front rotor by reducing maximum pitch angle to dissipate this unintentional pitching moment due to the force imbalance. As a result, maximum pitch angle of front rotor for TSR 0.10 case is reduced to 12 degrees. The Figure 3.10 (b) also shows the imbalance of flow over two rotors. However this is not an efficient flight condition since the thrust efficiency of front rotor is higher. Thus transition of flight posture is required for higher velocity flight.

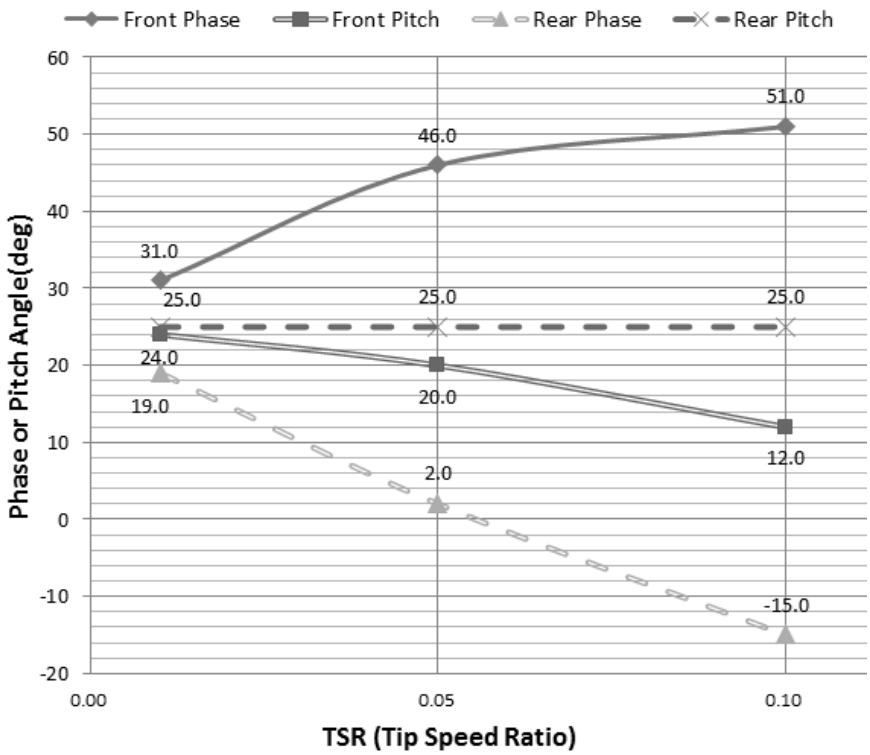


Figure 3.11 Adjusted pitch and phase angle

3.3 Titled Two Rotor Analysis

The posture that is tilted to forward as shown in the Figure 3.11 would help rear rotor take more flows without losing thrust of front rotor. Therefore following analysis is conducted with tilted flight posture for higher velocity, over TSR 0.10

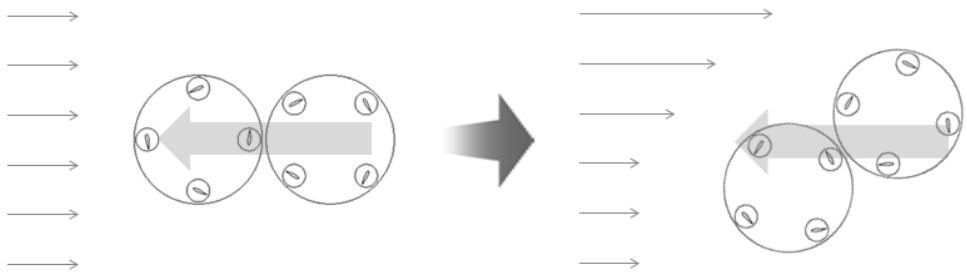


Figure 3.12 Transition of flight posture

There is an additional interface zone for tilted two rotor analysis. The tilting zone is added between hub cell zone and outer cell zone. The rest part of computational model is same with previous non-tilted model. The basic configuration for tilted case is same with former one. The definition of zero phase angles also set to vertical direction although the upper side of vehicle is tilted to forward. The tilting degrees considered are 10, 20, 30 and 45 degrees and each case is calculated for different velocities from TSR 0.15 to 0.40. As aforementioned, the phase angle of both rotors for initial case is set to the value that is obtained from the single rotor analysis.

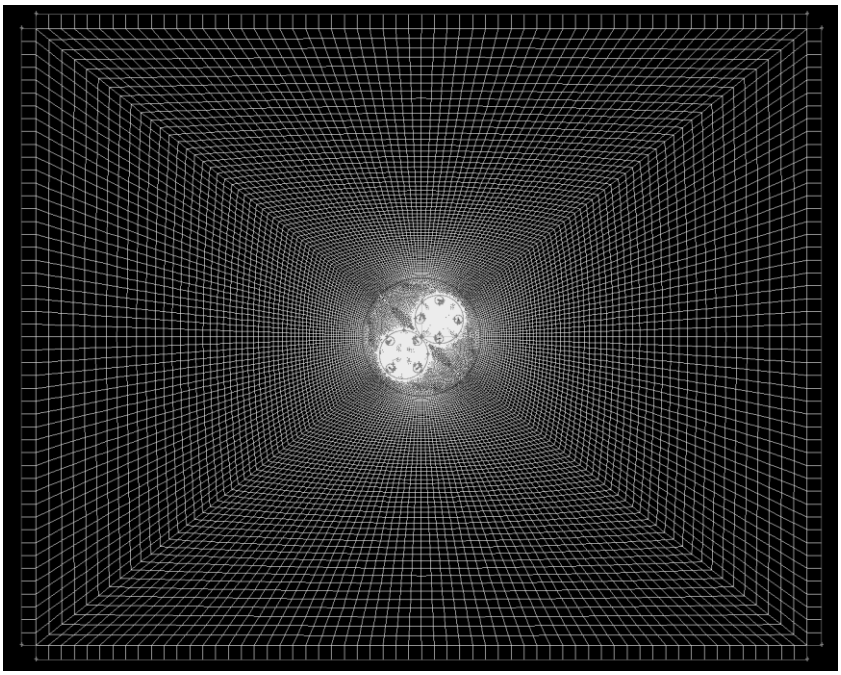
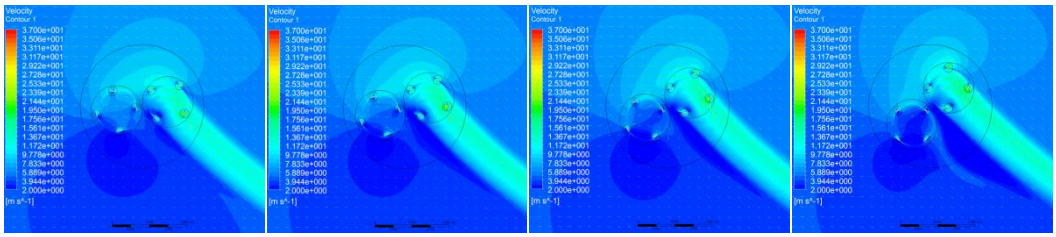


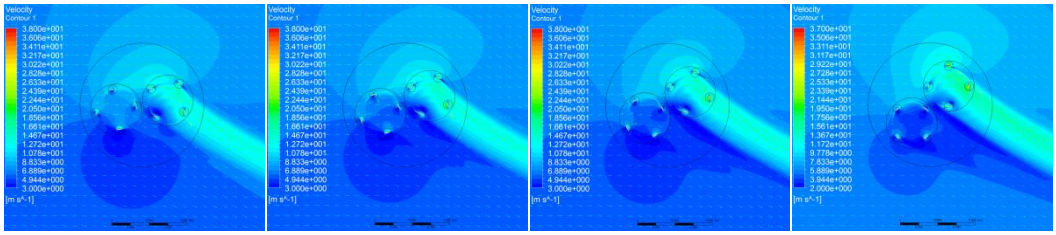
Figure 3.13 Finite element model of tilted rotor analysis

The flow interference to rear rotor due to the downwash effect from front rotor is reduced as flight posture is tilted. The Figures 3.13 to 3.18 are the result of phase and pitch correction cases. The velocity contour shows that the rear rotor is released from the flow came out from front rotor as tilting degree increases. However, the difference of flow features over rotors shows that the imbalance of thrust still exists. In fact, the distribution of thrust in the Figure 3.19 shows the differential of thrust on the rotors.



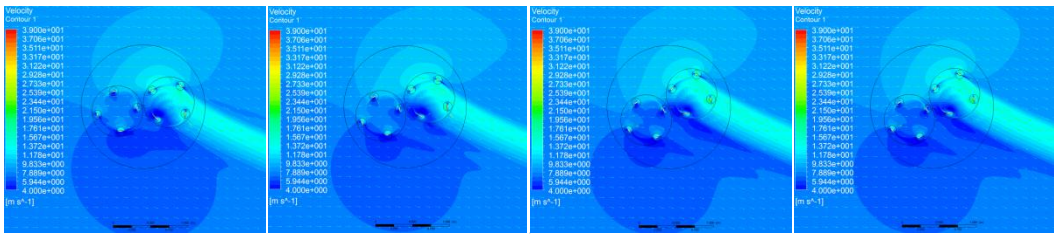
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.14 Velocity contours of TSR 0.15(4.67m/s) case – C.G -0.225m



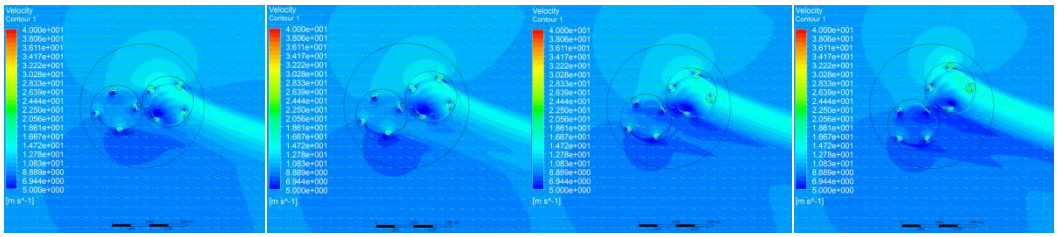
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.15 Velocity contours of TSR 0.20(6.22m/s) case – C.G -0.225m



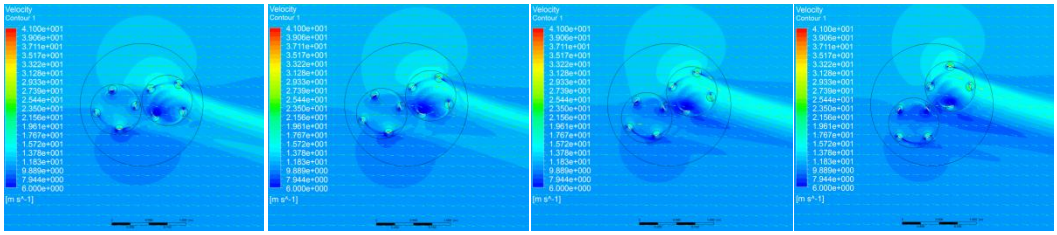
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.16 Velocity contours of TSR 0.25(7.78m/s) case – C.G -0.225m



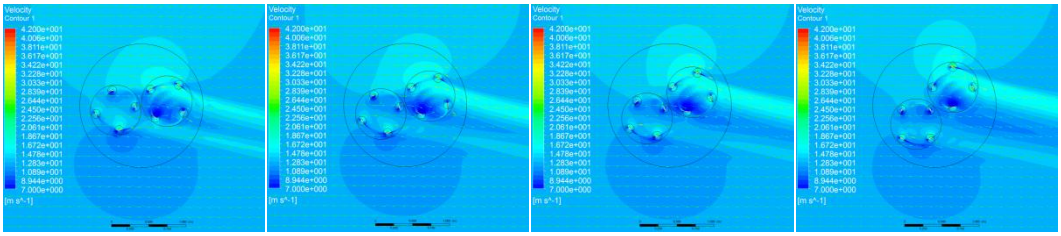
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.17 Velocity contours of TSR 0.30(9.33m/s) case – C.G -0.225m



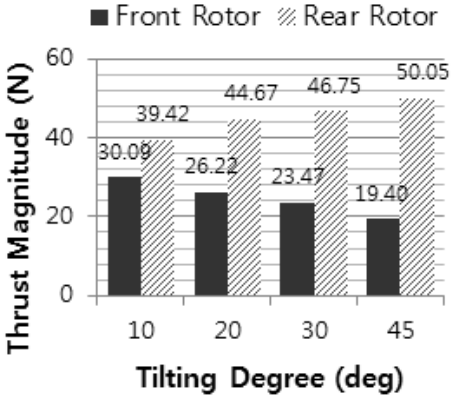
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.18 Velocity contours of TSR 0.35(10.89m/s) case – C.G -0.225m

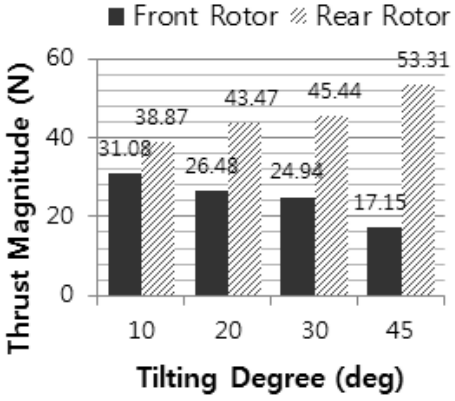


(a) 10deg (b) 20deg (c) 30deg (d) 45deg

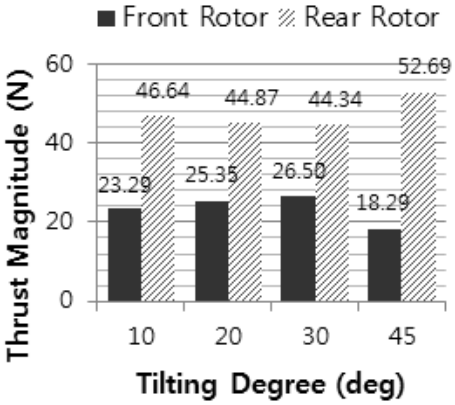
Figure 3.19 Velocity contours of TSR 0.40(12.44m/s) case – C.G -0.225m



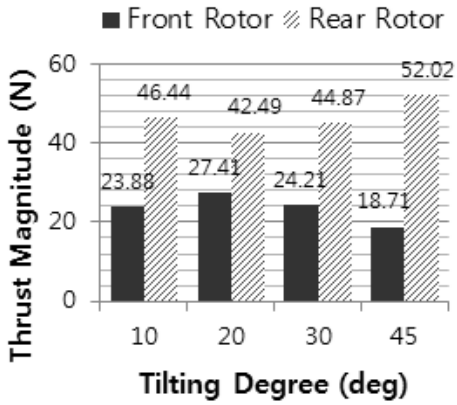
(a) TSR 0.15(4.67m/s)



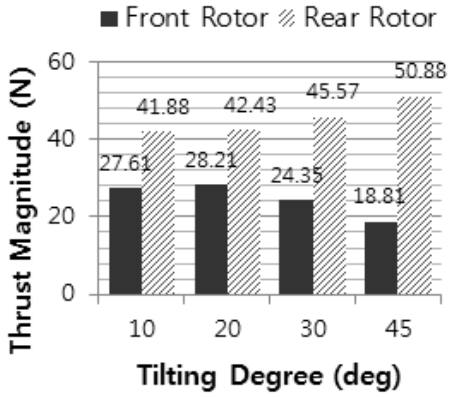
(b) TSR 0.20(6.22m/s)



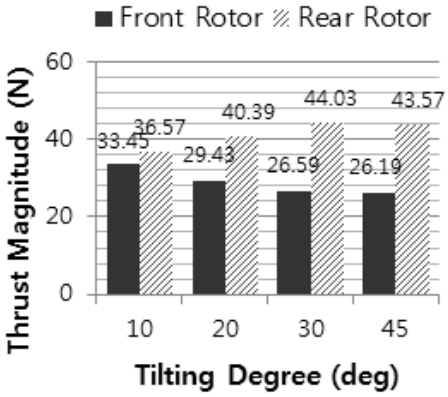
(c) TSR 0.25(7.78m/s)



(d) TSR 0.30(9.33m/s)



(e) TSR 0.35(10.89m/s)



(f) TSR 0.40(12.44m/s)

Figure 3.20 Thrust distribution on rotors – C.G -225m Thrust distribution

The reason that still the thrust is concentrated on the rear rotor is because of the pitching moment. The Table 3.7 shows that relatively large pitching moment is created compare to level flight cases. Furthermore, moment increases more as tilting degree increases as shown on the next table. One reason might be that the velocity is increased compare to non-tilted case. However, the next figure indicates that the tilting degree is a dominant factor of large pitching moment, not the velocity. The reason for large pitching moment is that the center of gravity is below the rotors. The Figure 3.21 explains this. The horizontal distance between the front rotor and the center of gravity increases as tilting degree increases. In case of rear rotor, it is the opposite. The distance creates unintended moment while there should be only vertical thrust which equals to weight for steady state flight. Therefore this leads to increased pitching moment.

Table 3.7 Initial 10deg tilted TSR 0.15(4.67m/s) case condition and result

	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	50°	25°	-1.44 N	56.73 N	
Rear	22°	25°	-4.65 N	37.43 N	
Total			-6.09 N	94.15 N	**-10.48 N·m**

Table 3.8 Initial 45deg tilted TSR 0.15(4.67m/s) case condition and result

	Phase Angle	Maximum Pitch Angle	Force-X	Force-Y	Pitching Moment
Front	50°	25°	-1.63 N	54.66 N	
Rear	22°	25°	0.51 N	49.34 N	
Total			-1.12 N	104.00 N	**-20.28 N·m**

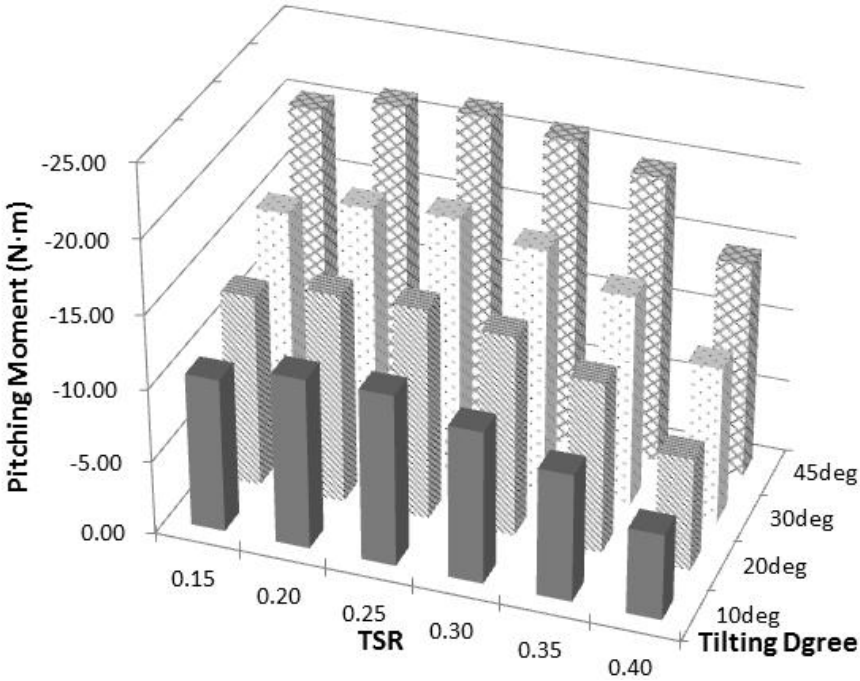


Figure 3.21 Pitching moment for initial cases

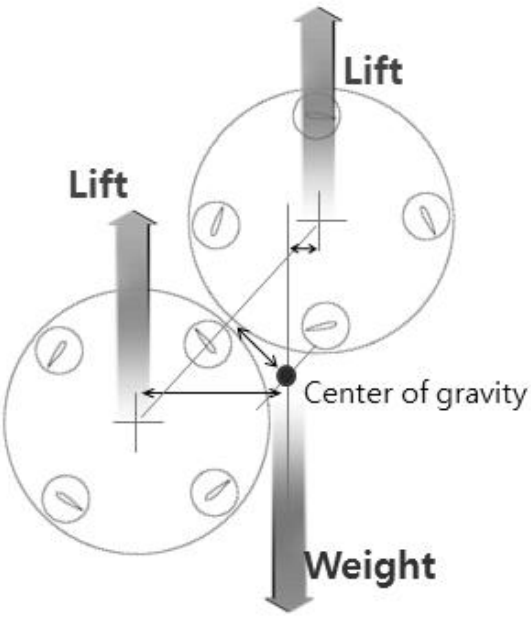


Figure 3.22 Diagram of tilted flight posture

The Figure 3.22 shows how much thrust difference between rotors is required in order to dissipate unintended pitching moment for locations of the center of gravity. In case of 0.225m below center of gravity, the thrust ratio of the front rotor to the rear rotor over 2:1 as tilting degree is over 45 degrees. As a result, the tilted posture maneuver would help the rear rotor to take enough flow but still bad for the efficiency.

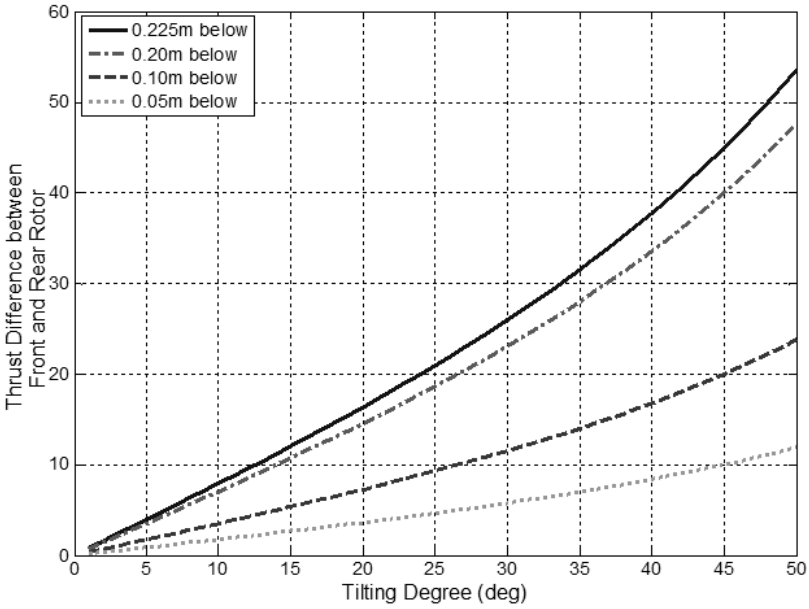
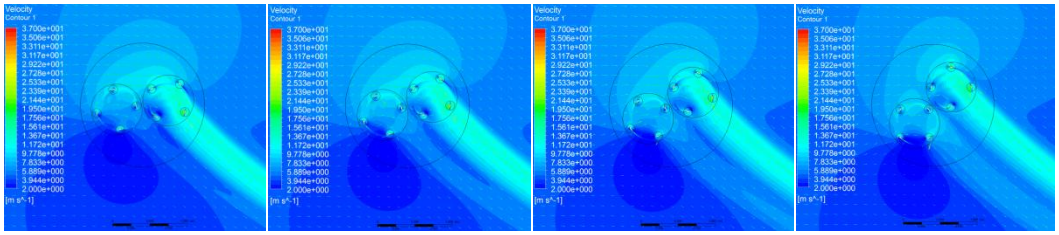


Figure 3.23 Required thrust difference between rotors for dissipation of pitching moment

3.4 C.G. Shifted Tilted Two Rotor Case

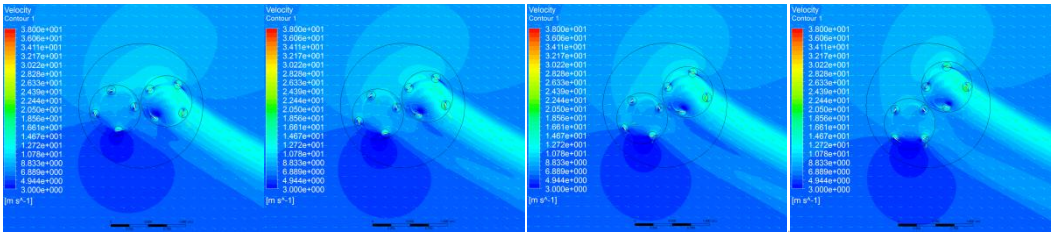
Eventually, the center of gravity should be shifted up to mid-point of rotors to reduce imbalance of thrust from rotors. As mentioned earlier, it is important to consider the efficiency of rotors.

Following figures are the results with corrected phase and pitch angles when the center of gravity is shifted. Following results are obtained after several cases of parametric study adjusting phase and pitch angles. The thrust is adjusted to 70N with the errors allowed to be less than one percent. The horizontal thrust is also allowed to be less than 1N which is negligible, and similarly, the pitching moment is permitted to be less than 1N·m.



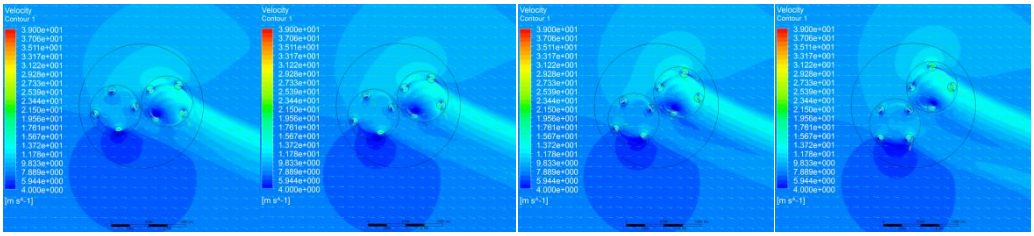
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.24 Velocity contours of TSR 0.15(4.67m/s) case – C.G zero



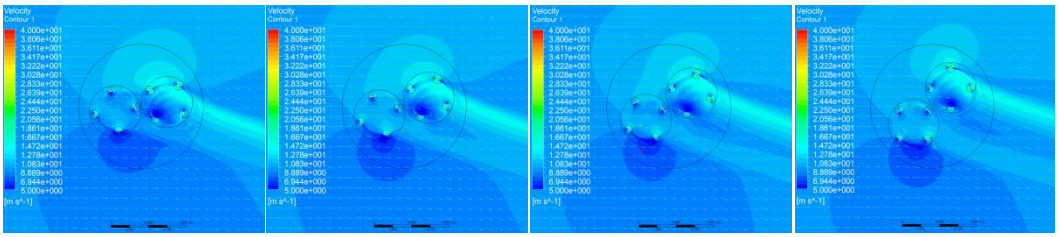
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.25 Velocity contours of TSR 0.20(6.22m/s) case – C.G zero



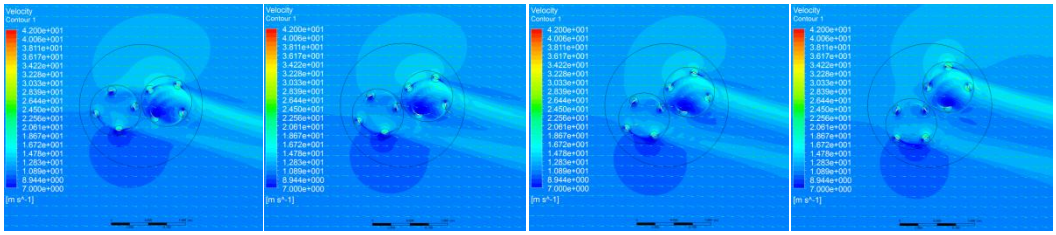
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.26 Velocity contours of TSR 0.25(7.78m/s) case – C.G zero



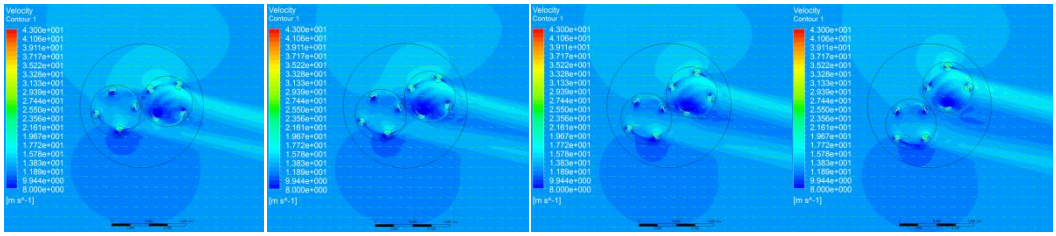
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.27 Velocity contours of TSR 0.30(9.33m/s) case – C.G zero



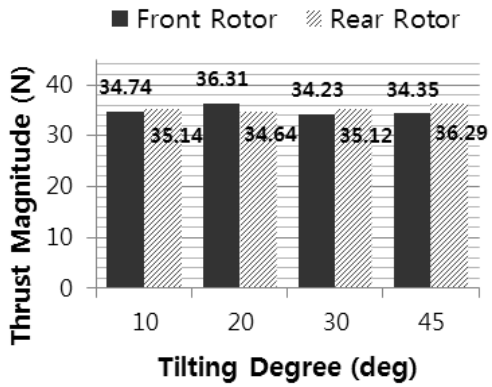
(a) 10deg (b) 20deg (c) 30deg (d) 45deg

Figure 3.28 Velocity contours of TSR 0.35(10.89m/s) case – C.G zero

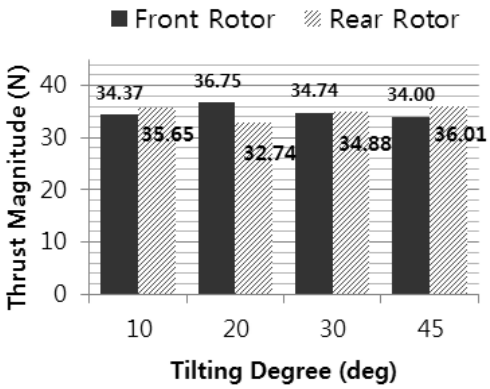


(a) 10deg (b) 20deg (c) 30deg (d) 45deg

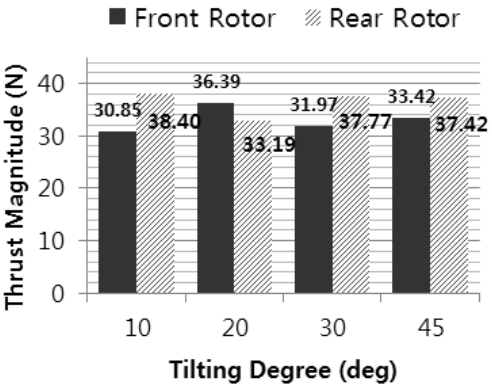
Figure 3.29 Velocity contours of TSR 0.40(12.44m/s) case – C.G zero



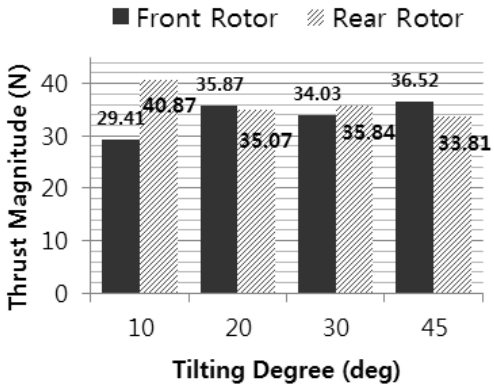
(a) TSR 0.15(4.67m/s)



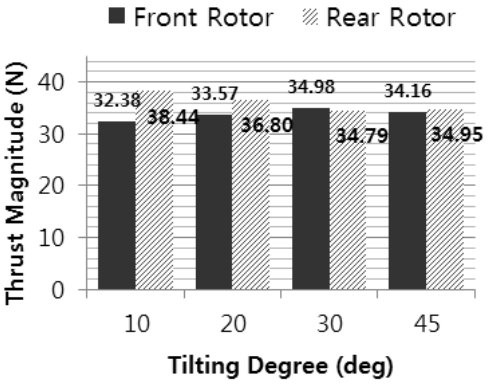
(b) TSR 0.20(6.22m/s)



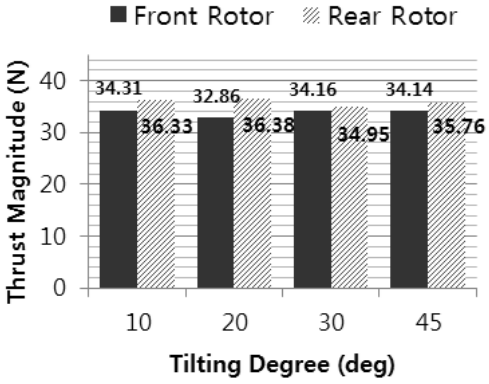
(c) TSR 0.25(7.78m/s)



(d) TSR 0.30(9.33m/s)



(e) TSR 0.35(10.89m/s)



(f) TSR 0.40(12.44m/s)

Figure 3.30 Thrust distribution on rotors – C.G. zero

The thrust is evenly distributed to the two rotors when compared to the former case which has different center of gravity location. The velocity contours show that the front rotor creates lager thrust compare to the former cases since the flow came from the front rotor is clearly visible. Equalizing rotor thrust leads to increase flight efficiency. The thrust efficiency of the front rotor is higher than that of the rear rotor, hence, increasing thrust portion of front rotor leads to improvement of the total efficiency. As shown in Fig 3.30, thrust per input power is increased by over 1 when the center of gravity location is shifted.

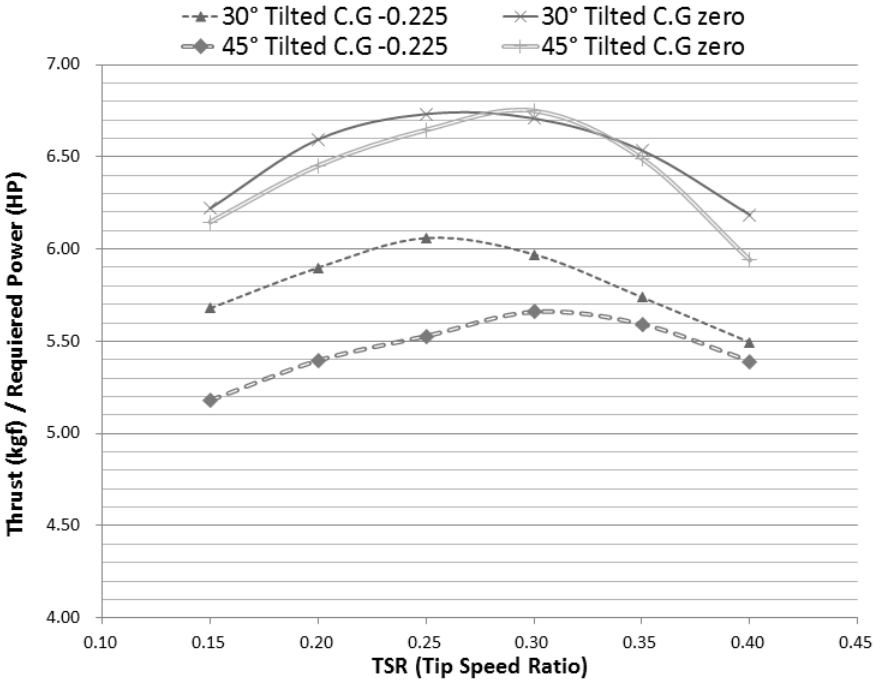


Figure 3.31 Rotor thrust efficiency difference

Both phase and pitch angles increase as velocity increases. The trends of variation of phase and pitch angles relate to the free stream velocity and there is weak relationship with tilting degree. The maximum pitch angle and phase angle are slightly changed as tilting degree changes. The phase angle of the front rotor slightly decreases as tilting

degree increases while the phase angle of the rear rotor increases. The flow heading to the rear rotor increased as tilting degree increases therefore the phase angle of the rear rotor should be increased to increase the thrust for forward direction.

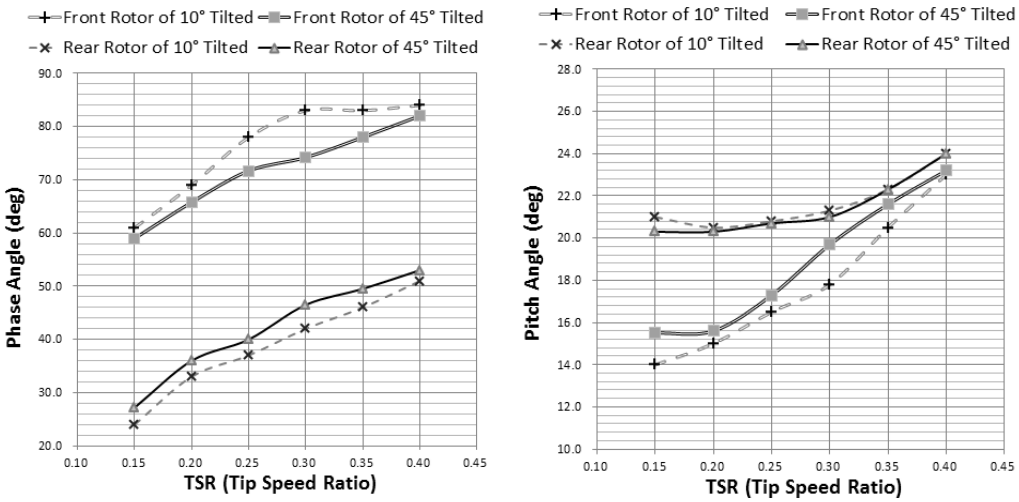


Figure 3.32 Phase and pitch angle correction result

However, although the control parameters rarely depend on the tilting degree, flight performance does. The Figures 3.23 to 28 show that the flow came out from the front rotor is more parallelized to the free stream as velocity increases. It means the rear rotor is affected by the wake flow from the front rotor. Therefore the tilting degree should be increased to reduce this interference. The following Figure 3.31 shows the required power for several inflow velocity of each tilting degree condition. The graph shows that tilting by 10, 20, and 45 degrees are optimum for TSR of 0.2, 0.25, and 0.3 cases, respectively. Also, it can be concluded that 30 degrees of tilting will result in the best performance when TSR lies between 0.2 and 0.3. Here, a weak tendency is found that for the higher velocity, the steeper tilting angle is required for better efficiency. The values written under the graph are the optimal values of thrust per required power. Although it is the result of rotor blades only, in which flow interference caused by rotor hub or structures and mechanical loss from power transfer unit such as gear or belt are not considered, these values still have higher competitiveness compare to the conventional rotorcraft analysis.

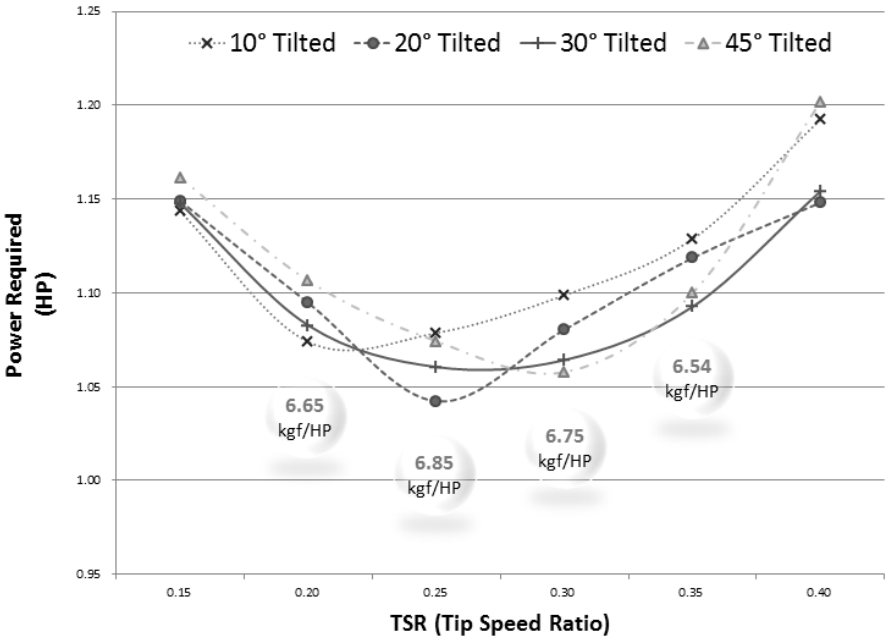


Figure 3.33 Cyclocopter Power Required for Forward Flight

4. CONCLUSIONS

This goal of this study is to find a stable condition of the forward flight of a quad-rotor cyclocopter, however, it is also important to understand flight behaviors and flow characteristics to predict flight mechanism and to build up control algorithm. Certain tendencies are found about the forward flight of the cyclocopter with several different phase and pitch angle conditions through a set of case studies. One of the tendencies is that reducing maximum pitch angle of the front rotor induces weakened downwash, and consequently, the rear rotor creates larger thrust that is caused by increased flow heading to the rear rotor. Therefore, only vertical component of thrust is increased. When a thrust angle is defined as an angle between vertical and thrust directions, the thrust angle of the rear rotor is decreased. In other words, the thrust direction of the rear rotor is tilted toward vertical. On the contrary, the thrust angle of the front rotor is increased since the vertical thrust is decreased. Thus it is noted that the front rotor phase angle should be added whereas the rear rotor phase angle should be lower when the maximum pitch angle of front rotor decreases.

Adjusting maximum pitch angle of the rear rotor also affects on the thrust of the front rotor. Thrust of the front rotor increases as thrust of the rear rotor increases and vice versa. This is because of the induced flow of the rear rotor. The induced flow becomes stronger as thrust of rear rotor increases and this leads to additional airflow into the front rotor.

Another notable point is that the magnitude of the thrust increases as the thrust angle approaches to zero. When the thrust angle lies between -5 and 5 degrees, the magnitude of the thrust is larger by approximately 10% than when the angle is less than -10 or larger

than 10. It indicates that in certain degree, the thrust is controlled by adjusting phase angle, not through changing maximum pitch angle.

These results thus prove that the dynamic characteristics of two rotors are interrelated and the flight behavior is very complicated. In that sense, this study is valuable since it deals with forward flight of two rotor cycloidal system while most former studies consider single rotor or hovering state only.

In summary of the analysis performed, in a level flight case, the rear rotor can not take enough flow due to the downwash of the front rotor. The tilted flight posture cases are followed to overcome this problem. However, thrust imbalance problem still exists to dissipate unintentional pitching moment caused by the position of the center of gravity. Eventually, efficient and stable forward flight condition is found when the center of gravity is shifted on the tilted posture flight. The tilting degree should be increased as the velocity increases to improve efficient. In addition to this, there are optimized amount of tilting degrees for corresponding flight velocity.

In conclusion, this study is meaningful since the forward flight characteristics of the cylocopter obtained from numerous flight conditions would be a useful source to establish a control algorithm. Furthermore the cyclocopter shows competitiveness for forward flight compare to the conventional rotorcrafts through this study. Therefore the cyclocopter would be considered as an efficient and effective flight approach in the near future.

REFERENCES

1. Kirsten, F.K., “Cycloidal Propulsion Applied to Aircraft,” Transactions of the American Society of Mechanical Engineers, Vol. 50, No. AER-50-12, pp.25-47, 1928.
2. Wheatley, J.B., “Simplified Aerodynamic Analysis of Cyclogiro Rotating-wing System,” Technical Notes NACA No.467, August, 1933
3. McNabb, M.L., “Development of a Cycloidal Propulsion Computer Model and Comparison with Experiment,” Master’s Thesis, Mississippi State University, 2001.
4. Hwang, I.S., “Numerical and Experimental Study on Design and Development of Quadrotor Cyclocopter,” Ph.D. Thesis, Seoul National University, 2009
5. Miller, Kaitlin, “Popular Science Feb 2012,” Vol. 280 Issue 2, p30-30 February, 2012
6. Benedict, M., Jarugumilli, T., Lakshminarayan, V., Chopra, I, “Experimental and Computational Studies to Understand the Role of Flow Curvature Effects on the Aerodynamic Performance of a MAV-Scale Cycloidal Rotor in Forward Flight,” 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, Hawaii, April 2012
7. Benedict, M., “500 gram Cyclocopter - University of Maryland,” http://www.youtube.com/watch?v=4j67-tUzJ0Y, 2012
8. Park, C.J., “Optimal Design and Verification of Cycloidal Vertical Axis Wind Turbine” Master’ Thesis, Seoul National University, 2011
9. Jeong, J.S., “An Experimental Study of UAV Cyclocopter with Cycloidal Blades System” Master’ Thesis, Seoul National University, 2004

APPENDIX A. APPLYING PITCHING CONTROL MECHANISM TO UDF

The understanding of pitching motion mechanism is required before applying a cyclodial pitching motion of cyclocopter to UDF. The maximum pitch angle is controlled by eccentricity of control point.

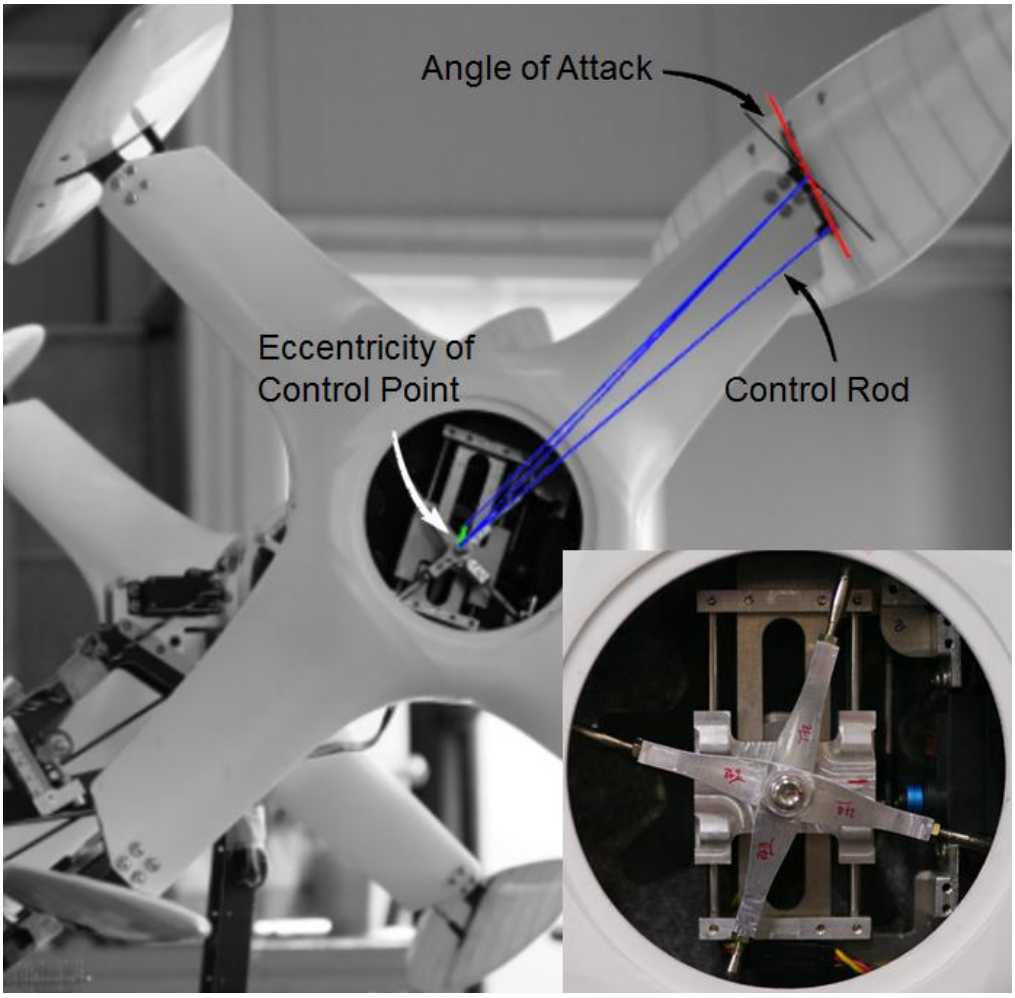
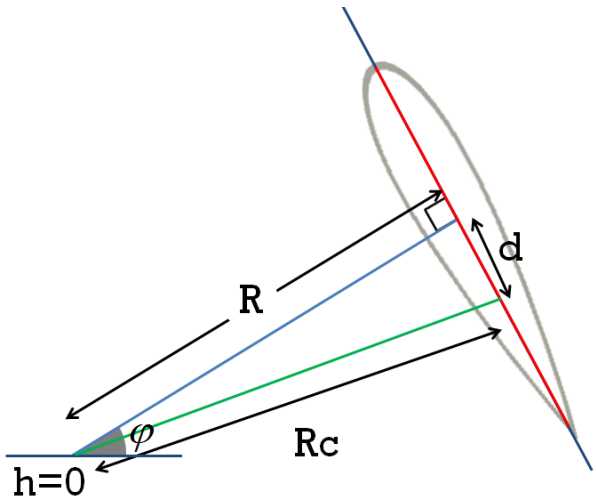
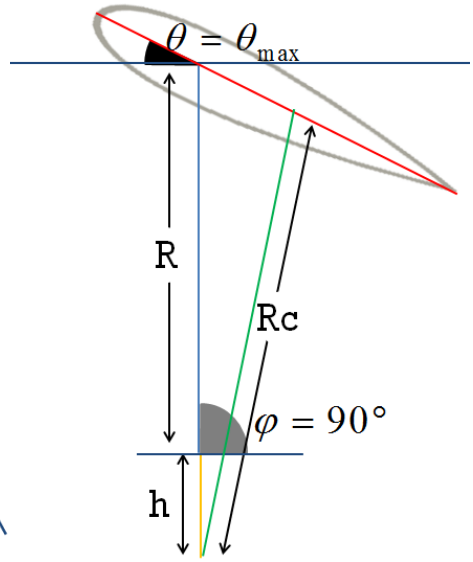


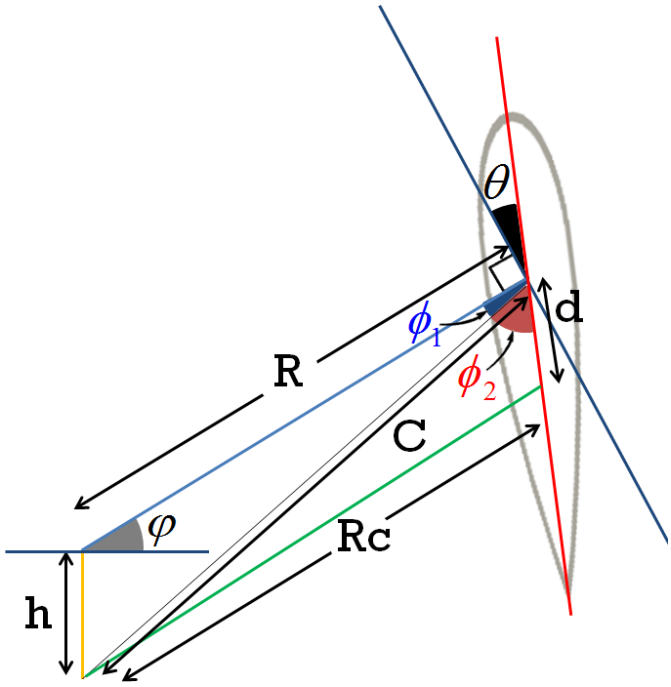
Figure A.1 Mechanical model of pitching mechanism



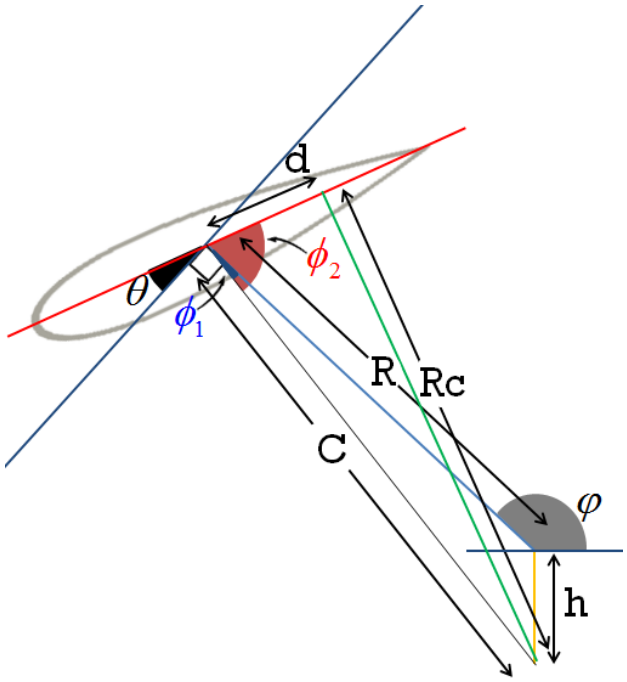
(a) Zero maximum pitch angles



(b) Maximum pitch angle of θ_{max}



(c) Pitch angle of θ at rotation angle of φ on quadrant I or IV



(d) Pitch angle of θ at rotation angle of φ on quadrant II or III

Figure A. 2 Schematic diagram of pitching mechanism

R is the radius of rotor and **h** is the eccentricity of control point. The length of control rod **Rc** is fixed and determined by the distance between pivot point and control rod joint **d**. When the maximum pitch angle $_{max}$ is applied, the eccentricity of control point **h** is adjusted. Since the pitch angle or the angle of attack θ varies with rotation, it should be described with the function of rotation angle φ to apply the mechanism to UDF. These can be calculated by geometrical relationship as shown on the Figure A.2.

$$Rc = \sqrt{R^2 + d^2} \tag{A.1}$$

$$h = d\sin\theta_m - R + \sqrt{\left(R - d\sin\theta_m\right)^2 + 2Rd\sin\theta_m} \tag{A.2}$$

$$C = \sqrt{h^2 + R^2 + 2Rh\sin\varphi} \tag{A.3}$$

$$\phi_1 = \cos^{-1}\left(\frac{R^2 + C^2 - h^2}{2RC}\right) \tag{A.4}$$

$$\phi_2 = \cos^{-1}\left(\frac{d^2 + C^2 - Rc^2}{2dC}\right) \tag{A.5}$$

$$\theta = \frac{\pi}{2} - \phi_1 - \phi_2 \qquad \text{for quadrant I or IV} \tag{A.6}$$

$$\theta = \frac{\pi}{2} + \phi_1 - \phi_2 \qquad \text{for quadrant II or III} \tag{A.7}$$

Unfortunately, the UDF of Fluent related to rotation is based on the angular velocity, not the angle. Therefore the angular velocity at each rotation angle should be obtained to implement UDF. However it is very difficult to formularize with differential because of the inverse cosine term. Thus the angular velocity for UDF is calculated by subtracting angles of each time step.

APPENDIX B. PITCHING MOTION UDF

>> vari.h

```
 real phase_angle = 51*M_PI/180; /*phase angle correction setting (rad)*/
 real max_pitch = 12*M_PI/180; /*max pitch setting (rad)*/
 real ang_vel_rotor = 1100*2*M_PI/60;    /*speed of fluid-rotating-core*/
 real R = 0.27; /*rotor hub radius*/
 real Rc = 0.271662; /*control rod length*/
 real h = 0.00627159; /*eccentricity of control point*/
 real d= 0.03; /*distance between pivot point and control rod joint*/
```

>> cyclo_motion.c

```
/* UDF for cyclocopter CFD */
#include "udf.h"
#include "vari.h"
DEFINE_ZONE_MOTION(motion_f_ypos,omega,axis,origin,velocity,time,dtime){
 real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2;
 real offset_angle = M_PI/2.0; /*Blade position, in radians, anticlockwise from +x axis*/
 current_angle = -ang_vel_rotor * (CURRENT_TIME ) + offset_angle ;
 next_angle =- ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
 cal_angle_1 = -current_angle+M_PI+phase_angle;
 cal_angle_2 = -next_angle+M_PI+phase_angle;
 x1= 0.27*cos(current_angle)-0.35;
 y1= 0.27*sin(current_angle);
 x2= 0.27*cos(next_angle)-0.35;
 y2= 0.27*sin(next_angle);
 C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
 C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
 quadrant=((int)(cal_angle_2/M_PI*2))%4;
 if(quadrant==0||quadrant==3)
 {          theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
            theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
 else
 {          theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
            theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
 velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
 velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
 origin[0] = x2;
 origin[1] = y2;
            *omega = -ang_vel_rotor-(theta_2-theta_1)/CURRENT_TIMESTEP;
 return;}

DEFINE_ZONE_MOTION(motion_f_xpos,omega,axis,origin,velocity,time,dtime){
 real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2;
 real offset_angle = 0.0; /*Blade position, in radians, anticlockwise from +x axis*/
 current_angle = -ang_vel_rotor * (CURRENT_TIME ) + offset_angle ;
 next_angle = -ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
 cal_angle_1 = -current_angle+M_PI+phase_angle;
 cal_angle_2 = -next_angle+M_PI+phase_angle;
 x1= 0.27*cos(current_angle)-0.35;
 y1= 0.27*sin(current_angle);
 x2= 0.27*cos(next_angle)-0.35;
```

```
  y2= 0.27*sin(next_angle);
  C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
  C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
  quadrant=((int)(cal_angle_2/M_PI*2))%4;
  if(quadrant==0||quadrant==3)
  {             theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  else
  {             theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
  velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
  origin[0] = x2;
  origin[1] = y2;
                *omega = -ang_vel_rotor-(theta_2-theta_1)/CURRENT_TIMESTEP;
  return;}

DEFINE_ZONE_MOTION(motion_f_yneg,omega,axis,origin,velocity,time,dtime){
  real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2;
  real offset_angle = -M_PI/2.0; /*Blade position, in radians, anticlockwise from +x axis*/
  current_angle = -ang_vel_rotor * (CURRENT_TIME ) + offset_angle ;
  next_angle = -ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
  cal_angle_1 = -current_angle+M_PI+phase_angle;
  cal_angle_2 = -next_angle+M_PI+phase_angle;
  x1= 0.27*cos(current_angle)-0.35;
  y1= 0.27*sin(current_angle);
  x2= 0.27*cos(next_angle)-0.35;
  y2= 0.27*sin(next_angle);
  C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
  C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
  quadrant=((int)(cal_angle_2/M_PI*2))%4;
  if(quadrant==0||quadrant==3)
  {             theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  else
  {             theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
  velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
  origin[0] = x2;
  origin[1] = y2;
                *omega = -ang_vel_rotor-(theta_2-theta_1)/CURRENT_TIMESTEP;
  return;}

DEFINE_ZONE_MOTION(motion_f_xneg,omega,axis,origin,velocity,time,dtime){
  real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2 ;
  real offset_angle = M_PI; /*Blade position, in radians, anticlockwise from +x axis*/
  current_angle = -ang_vel_rotor * (CURRENT_TIME ) + offset_angle ;
  next_angle = -ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
  cal_angle_1 = -current_angle+M_PI+phase_angle;
  cal_angle_2 = -next_angle+M_PI+phase_angle;
  x1= 0.27*cos(current_angle)-0.35;
  y1= 0.27*sin(current_angle);
  x2= 0.27*cos(next_angle)-0.35;
  y2= 0.27*sin(next_angle);
```

```
  C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
  C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
  quadrant=((int)(cal_angle_2/M_PI*2))%4;
  if(quadrant==0||quadrant==3)
  {             theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  else
  {             theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
  velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
  origin[0] = x2;
  origin[1] = y2;
                *omega = -ang_vel_rotor-(theta_2-theta_1)/CURRENT_TIMESTEP;
  return;}

DEFINE_ZONE_MOTION(motion_f_hub,omega,axis,origin,velocity,time,dtime){
  origin[0] = -0.35;
  origin[1] = 0;
  *omega = -ang_vel_rotor;
  return;}

DEFINE_ZONE_MOTION(motion_r_ypos,omega,axis,origin,velocity,time,dtime){
  real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2;
  real offset_angle = 3*M_PI/4.0; /*Blade position, in radians, anticlockwise from +x axis*/
  current_angle = ang_vel_rotor * (CURRENT_TIME ) + offset_angle;
  next_angle = ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
  cal_angle_1 = current_angle-phase_angle;
  cal_angle_2 = next_angle-phase_angle;
  x1= 0.27*cos(current_angle)+0.35;
  y1= 0.27*sin(current_angle);
  x2= 0.27*cos(next_angle)+0.35;
  y2= 0.27*sin(next_angle);
  C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
  C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
  quadrant=((int)(cal_angle_2/M_PI*2))%4;
  if(quadrant==0||quadrant==3)
  {             theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  else
  {             theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
  velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
  origin[0] = x2;
  origin[1] = y2;
                *omega = ang_vel_rotor+(theta_2-theta_1)/CURRENT_TIMESTEP;
  return;}

DEFINE_ZONE_MOTION(motion_r_xpos,omega,axis,origin,velocity,time,dtime){
  real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2;
  real offset_angle = M_PI/4.0; /*Blade position, in radians, anticlockwise from +x axis*/
  current_angle = ang_vel_rotor * (CURRENT_TIME ) + offset_angle;
  next_angle = ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
  cal_angle_1 = current_angle-phase_angle;
```

```
  cal_angle_2 = next_angle-phase_angle;
  x1= 0.27*cos(current_angle)+0.35;
  y1= 0.27*sin(current_angle);
  x2= 0.27*cos(next_angle)+0.35;
  y2= 0.27*sin(next_angle);
  C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
  C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
  quadrant=((int)(cal_angle_2/M_PI*2))%4;
  if(quadrant==0||quadrant==3)
{                theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                 theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  else
  {              theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                 theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
  velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
  origin[0] = x2;
  origin[1] = y2;
                 *omega = ang_vel_rotor+(theta_2-theta_1)/CURRENT_TIMESTEP;
  return;}

DEFINE_ZONE_MOTION(motion_r_yneg,omega,axis,origin,velocity,time,dtime){
  real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2;
  real offset_angle = -M_PI/4.0; /*Blade position, in radians, anticlockwise from +x axis*/
  current_angle = ang_vel_rotor * (CURRENT_TIME ) + offset_angle;
  next_angle = ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
  cal_angle_1 = current_angle-phase_angle;
  cal_angle_2 = next_angle-phase_angle;
  x1= 0.27*cos(current_angle)+0.35;
  y1= 0.27*sin(current_angle);
  x2= 0.27*cos(next_angle)+0.35;
  y2= 0.27*sin(next_angle);
  C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
  C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
  quadrant=((int)(cal_angle_2/M_PI*2))%4;
  if(quadrant==0||quadrant==3)
  {              theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                 theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  else
  {              theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                 theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
  velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
  origin[0] = x2;
  origin[1] = y2;
                 *omega = ang_vel_rotor+(theta_2-theta_1)/CURRENT_TIMESTEP;
  return;}

DEFINE_ZONE_MOTION(motion_r_xneg,omega,axis,origin,velocity,time,dtime){
  real x1,y1,x2,y2,current_angle,next_angle,cal_angle_1,cal_angle_2,C1,C2,quadrant,theta_1,theta_2;
  real offset_angle = -3*M_PI/4.0; /*Blade position, in radians, anticlockwise from +x axis*/
  current_angle = ang_vel_rotor * (CURRENT_TIME ) + offset_angle;
  next_angle = ang_vel_rotor * (CURRENT_TIME + CURRENT_TIMESTEP) + offset_angle;
  cal_angle_1 = current_angle-phase_angle;
  cal_angle_2 = next_angle-phase_angle;
```

```
  x1= 0.27*cos(current_angle)+0.35;
  y1= 0.27*sin(current_angle);
  x2= 0.27*cos(next_angle)+0.35;
  y2= 0.27*sin(next_angle);
  C1 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_1));
  C2 = sqrt(h*h+R*R+2*R*h*sin(cal_angle_2));
  quadrant=((int)(cal_angle_2/M_PI*2))%4;
  if(quadrant==0||quadrant==3)
  {             theta_1=acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  else
  {             theta_1=-acos((R*R+C1*C1-h*h)/(2*R*C1))+acos((d*d+C1*C1-Rc*Rc)/(2*d*C1));
                theta_2=-acos((R*R+C2*C2-h*h)/(2*R*C2))+acos((d*d+C2*C2-Rc*Rc)/(2*d*C2));}
  velocity[0] = (x2 - x1) / CURRENT_TIMESTEP;
  velocity[1] = (y2 - y1) / CURRENT_TIMESTEP;
  origin[0] = x2;
  origin[1] = y2;
                *omega = ang_vel_rotor+(theta_2-theta_1)/CURRENT_TIMESTEP;
  return;}

DEFINE_ZONE_MOTION(motion_r_hub,omega,axis,origin,velocity,time,dtime){
  origin[0] = 0.35;
  origin[1] = 0;
  *omega = ang_vel_outer;
  return;}
```

초 록

사이클로콥터는 사이클로이달 블레이드 시스템을 활용한 수직이착륙 비행체이다. 사이클로이달 블레이드 시스템은 블레이드 스팬 방향과 평행한 축을 기준으로 피치각을 가변하며 회전하는 블레이드들로 구성되어있다. 서울대학교 항공우주구조연구실에서 개발된 사이클로콥터는 최대 피치각과 위상각이 각각 따로 제어되는 4개의 로터로 이루어져있다. 각각의 로터는 4개의 블레이드로 이루어져있으며 추력과 추력방향은 각각 최대 피치각과 위상각을 통해 제어된다.

해석에 사용된 사이클로콥터는 탠덤방식의 두개의 로터가 서로 가까이 배치되어 있기 때문에 호버링이나 전진비행시 두 로터의 유동에 간섭이 생길 수 있다. 그러므로 각각의 비행여건에 맞는 피치각이나 위상각에 대한 제어조건을 결정하기 위해선 두 로터사이의 유동에 대한 이해가 반드시 필요하다. 본 논문에서는 전산유동해석을 통해 두개의 로터로 이루어진 사이클로이달 시스템에 대한 연구를 수행하였다. 또한 회전시 가변하는 블레이드 피치각을 고려하기위해 비정상 해석기법을 사용하였으며 해석모델은 전체 기체의 절반에 해당되는 앞쪽과 뒤쪽의 두개의 로터를 이차원 유한요소 모델로 구성하였다. 해석에는 상용해석 프로그램인 ANSYS Fluent가 사용되었으며 후처리에는 ANSYS CFD-Post가 사용되었다. 회전중에 가변하는 블레이드의 피치각을 구현하기 위해 ANSYS Fluent의 사용자 정의함수(UDF)와 Sliding Mesh 기법을 사용하였다. 각각의 블레이드와 로터는 슬라이딩 인터페이스을 가진 원형의 영역으로 모델링 되었다.

두개의 로터로 구성된 시스템은 한 개의 로터로 구성된 시스템과 비교해

해석결과에 있어 추력이나 효율, 유동의 형태 등에서 차이를 가지게 된다. 전진 비행시 뒤쪽 로터로 향하는 유동의 형태는 앞쪽의 로터와 다르며 이로 인해 앞뒤 로터 간의 추력불균형이 생기게 될 수 있다. 이러한 추력불균형의 정도는 비행속도와 자세, 방향 등에 따라 달라질 수 있다. 전진 비행중 앞뒤 로터 간의 추력불균형을 해소하기 위해서는 뒤쪽의 로터가 앞쪽 로터에서 생긴 유동에 영향을 덜 받을 수 있도록 기체를 앞쪽으로 기울일 필요가 있다. 또한 안정적인 비행자세의 유지를 위해 각 로터의 피치각과 위상각이 각각의 비행조건에 맞게 적절히 정해져야 한다.

본 연구의 주된 목적은 비행속도와 피치각, 위상각, 그리고 비행자세 등의 조건을 다양하게 적용시켜가며 두개의 로터로 이루어진 사이클로이달 시스템의 유동해석을 통해 사이클로콥터가 전진비행시 가질 수 있는 거동특성을 파악해보고 이를 추후 제어 알고리즘에 적용하고 활용하는데 있다. 따라서 이번 해석은 다양한 피치각과 위상각 조건들의 해석을 통해 각각의 비행속도와 자세에 맞는 안정적이고 효율적인 비행조건을 찾는데 그 목표를 가지고 수행되었다. 결과적으로 이번 해석을 통해 사이클로콥터의 전진비행시 거동적 특성을 확인할 수 있었고 로터의 추력 및 효율을 통해 차세대 비행체로서의 운용가능성을 확인 할 수 있었다.

주요어: 사이클로콥터 전진비행, 사이클로이달 블레이드 시스템, 사이클로이달 가변피치, CFD

학번: 2011-20696

감사의 글

2010년 여름방학 김승조 교수님께서 출연하신 방송을 보고 무작정 교수님께 찾아와 연구실에 들어오고 싶다고 하였던 것이 생각납니다. 기대와 설렘으로 시작하였던 연구실 생활이 어느덧 마지막을 앞두게 되었습니다. 사이클로콥터가 비행에 성공하였을 때의 감격이 바로 어제 같은데 어느새 이렇게 시간이 흐른 것인지 지나간 시간이 야속하기만 합니다. 이제와 좀 더 많은 것을 이루지 못한 것에 대한 아쉬움과 연구실 사람들과 함께한 시간들에 대한 그리움에 졸업의 기쁨이 그리 크지만은 않은 듯싶습니다.

일단 무엇보다 많이 부족하기만 저를 받아주시고 많은 것을 가르쳐주시고 좋은 기회를 주셨던 김승조 교수님께 감사 드립니다. 또한 지도교수님 자리를 대신해주신 김용협 교수님과 논문심사를 맡아주신 정인석 교수님께도 감사의 말을 전해드립니다.

그 동안 함께해준 연구실 선후배님들께 감사드립니다. 덕분에 정말 많은 것을 배우고 많은 도움을 받았습니다. 먼저 44동의 승용이형, 충희형 44동을 이끌어 주시고 저를 이끌어 주셔서 너무나도 감사했습니다. 그리고 궂은 일도 항상 마다하지 않으며 각종 제작에 열의를 불태웠던 명훈이형, 항상 뒤에서 저를 도와주신 창무형 정말 감사하고 잊지 않겠습니다. 묵묵히 연구실 뒷바라지 해주신 윤희누나도 감사합니다. 영어로 논문을 쓰는데 누구보다 많은 도움을 주신 희석이형 이 은혜 잊지 않겠습니다. 연구하는 내내 귀찮게 도움을 구해도 웃으며 받아주었던 국진,용진 형제와 희진, 정말 고마웠습니다. 이 밖에도 연구실 생활을 함께하며 많은 도움을 주셨던 민기형, 완일이형, 연희누나, 니테쉬, 서혜원씨까지... 정말 감사하다는 말 드리고 싶습니다. 그리고 무엇보다 항상 옆에 있어주며 든든한 힘이 되었던 윤정.. 고마워요.

마지막으로 이제 사회에 첫걸음을 내딛기까지 부족하고 못난 자식을 키워주시고 이끌어주신 부모님께 이 글을 바치고 싶습니다. 앞으로 많이 효도하고 행복하게 해드리겠습니다. 감사합니다!!