Research into the effect of Counter-Rotating Propellers, for the propulsion of a Vertical Take-Off and Landing Ducted Fan UAV, on the flow pattern

Internship assignment, executed by M. Voerman at the Royal Melbourne Institute of Technology, as part of the master programme Mechanical Engineering at the University of Twente, The Netherlands

February – April 2012

Supervisors on behalf of RMIT University (SAMME and SMG):

Prof. C. Bil and Prof. Eddie, Ly

Supervisor on behalf of the School of Engineering Fluid Dynamics, Twente University:

Prof. Dr. In. H. Hoeijmakers

Chapter 0: Summary

Over the past couple of years, members of the School of Aerospace and Mechanical Engineering at the Royal Melbourne Institute of Technology, together with a number of exchange students, have worked on the design of a Vertical Take-Off and Landing Unmanned Aerial Vehicle. These efforts have resulted in a preliminary design of a VTOL UAV and a scale model that has been tested in a wind tunnel.

UAVs are increasingly being used for both military and non-military purposes, like for example reconnaissance missions, surveillance and fire fighting. The advantage of an airplane that can be controlled from a distant location is that no lives are being put at risk and that it is easier to maintain than manned aircraft. Because the UAV, developed at RMIT University, is able to take-off and land vertically it requires only little operating space while at the same time it has much better long range flight characteristics than an ordinary helicopter.

Figure 1 shows a picture of the design at the start of the internship project. Up to this point, the aerodynamics of the plane had only been modelled without taking into account the motion of the (counter rotating) propellers. Because the flight direction of the UAV is controlled by changing the position of the ailerons that are part of the control surfaces, it is important that the flow coming out of the duct is *as straight as possible*. The idea behind a propulsion system with counter rotating propellers is that by making the propellers spin in opposite directions, the outflow from the duct will more or less be straight.

The goal of the internship is to implement the swirl that results from the movement of the propeller blades into a numerical model. Hopefully this will provide a better idea of how the counter rotating propeller concept will turn out in practice.

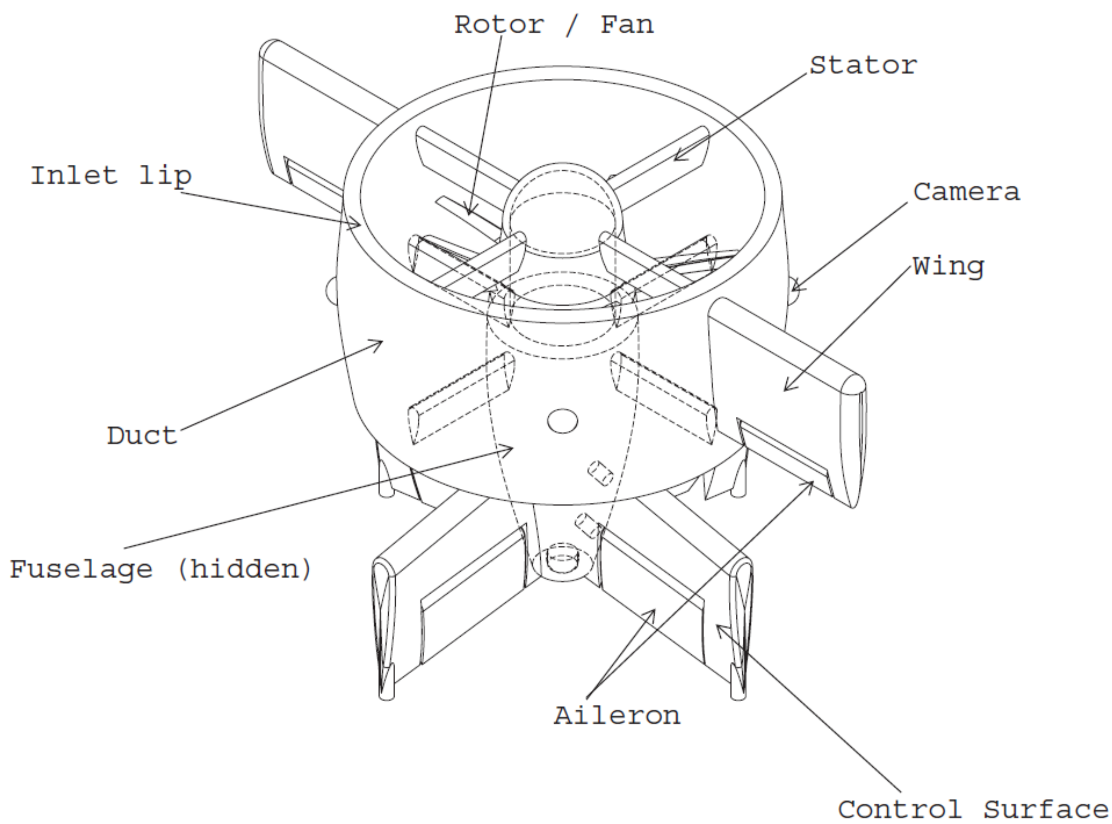

Figure 1: Overview of the VTOL-UAV configuration

Contents

Acknowledgements

I would like to thank Prof. Dr. In. H. Hoeijmakers of the University of Twente, for his extensive contacts in the field of aerospace that made it possible for me to go abroad and perform my internship project at RMIT University in Melbourne, Australia.

On behalf of RMIT, Associate Professor Cees Bil. and Dr. Eddie Ly, supervised my work and provided support in case I got stuck with my assignment. Their efforts have greatly added to my experience in Melbourne.

I'd also like to thank the School of Aerospace, Mechanical and Manufacturing Engineering for providing me with the assignment and the accompanying material in the form of the SolidWorks model of the geometry and the files and reports of my predecessors. This has been of great help to me.

I'd also like to thank the School of Mathematical and Geospatial Sciences and especially secretary Erin Schembri, for providing me with an office during the period of my stay and all the facilities that I could have possibly needed during that time.

Ibrahim Sargon, a SAMME-student, helped me out with setting up the first versions of my model in ANSYS CFX. Thanks to his knowledge of the software I was able to save some time which I used later on to improve the model and obtain better results.

LEAP Australia, a Melbourne based company specialised in ANSYS products, was available for questions whenever I couldn't find the source of an error generated by the software. Thanks to their helpdesk I was able to talk to experienced people, who are familiar with the kind of problems I had to deal with.

Chapter 1: Introduction

For the major part, this project is about creating a well-performing CFD-model capable of simulating propeller affected flow past an UAV in flight. After such a model has been created, the results can be used for *further design improvements in the future*. This chapter discusses the general approach that was used to obtain the results.

Analysis of the aerodynamic behaviour of the UAV was carried out by making use of CFD software. Whoever worked in the field of *computational flow dynamics* is aware of the fact that there are many different software packages available on the market nowadays, which makes it a lot harder to find out what package best suits the problem at hand. During the entire internship project, a lot of different programs were used to carry out different jobs, starting with the creation of the geometry that was subject to flow analysis.

In this particular case, the geometry was already available in the form of a model created in *SolidWorks*. Up to this point the design of the duct, the wings, the fuselage and the tail section had been optimized for *optimal flight performance within the limits of weight and dimensions*. So nothing much had to be changed here. Minor changes to the geometry that had to be made on behalf of mesh generation, will be discussed later on, when the different analysis cases will be looked upon in closer detail.

After the geometry had been changed to meet meshing requirements, it was exported to ANSYS Workbench (**Figure 2**). More specifically ANSYS DesignModeler.

ANSYS is a software package that can be used for many different kinds of analyses, like for example mechanical, thermal, or in this case aerodynamic analysis. ANSYS is not able to handle the .SLDPRT and .SLDASM type of files that are standard when creating SolidWorks models, so there was a need to *export* the files in a different format. For this purpose, there are different options to choose from: .STEP, .IGS, .x_t and many more. Literature is not unambiguous about what kind of file is preferred, but .x_t, better known as *Parasolid*, seemed to work fine.

When no model is available at all, ANSYS DesignModeler can be used to create a model from scratch, but in this case the program was only used to *extend* the existing geometry by creating the *control volume* around the UAV and to replace the actual propellers by disks of the same size.

From ANSYS DesignModeler the geometry was plugged into a *mesh generator*. Meshing is a delicate operation that requires a lot of attention from the user. It is a vital part of practising CFD though. Meshing makes it possible to go from a continuous domain towards a domain that is *discretised*, containing a finite amount of elements filling up the space. Without discretization, the computer model would never be able to calculate a solution.

There are different computer programs on the market to take on this task, of which ANSYS Workbench' default program, *ANSYS Mesher*, has been used. In the past, meshing for this project had always been carried out using the more powerful and easy to use software package *GAMBIT*, but this software was not available at the time of the internship project. This made the files of my predecessors that I was provided with, quite useless.

When, at first, creating a good mesh with ANSYS Mesher didn't seem to be successful, for what now seems to be caused by an error in the SolidWorks-model, ICEM was tried instead. ICEM allows the user to have a lot of influence on the mesh design by making use of a method called 'blocking'. Blocking divides the space to be meshed into different areas, or 'blocks', to which the user can assign mesh properties, mainly related to the size of the elements in that particular region. Mastering the technique of *manually making a mesh*, results in *better meshes* and *better trouble-shooting* in case the CFD-Solver doesn't accept the mesh. Like has been said before it requires a *lot more time* to manually create a mesh, so automatic mesh generation is the preferred method.

After the mesh had been defined, but before the start of any calculations, *pre-processing* of the case took care of *boundary condition* application, solver-process control and the definition of output parameters. Again, there were different software packages to choose from, although all of the considered options were able to operate within the ANSYS Workbench environment.

The first couple of times, FLUENT was used to pre-process the case and in later stages to run the calculation and display the results. For this purpose there is nothing wrong with using FLUENT, which is the pre-processor designed and built by the ANSYS Corporation, but it turned out to be that colleagues of the School of Aerospace and Mechanical Engineering were more familiar with CFX. CFX does practically the same, but lets you choose from a range of boundary conditions that is slightly different from the one that is provided by FLUENT. One example is the very useful boundary condition 'free slip wall' for the wall of the surrounding cylinder. Although in reality the surroundings of the UAV during flight wouldn't be anywhere near the plane, modelling of the control volume implies creating a bounding surface. By assigning the 'free-slip wall' boundary condition to the wall of the cylinder, the flow will not be decelerated when it comes close to this surface. So that is one of the reasons why CFX was chosen.

When the boundary conditions were defined, the *solution* was initiated. Dependent on the complexity of the model, calculations can either take a few minutes or a couple of days(!). During the calculation the *residual* provides a good indicator for convergence. The residual is a measure for the local imbalance of each conservative control volume equation. It is the most important *measure of convergence* as it relates directly to whether the equations have been solved. CFX presents the normalized residuals to judge convergence. By normalizing the residuals, you are presented with a relatively consistent means of judging convergence. The Normalized Residual is used to automatically stop the CFX-Solver run when a specified level has been obtained. There are two types of residuals, of which *Root Mean Square (RMS)* has been used. Playing around with the *residual target* and the maximum amount of iterations is useful, because sometimes the calculation will be terminated on behalf of the limit to the amount of iterations, while the residual target has not been reached yet. This will lead to a solution that has not converged, and provides false information to the user. When the results in the post-processor do not seem to make any sense, the solution probably has not converged yet, so either the maximum amount of iterations needs to be raised, or the value for the Residual Mean Square needs to be lowered.

When the calculation has ended, either by reaching the residual target or by interference of the user, the results can be plugged into a *post-processor*. Post-processing lets the user choose between varieties of methods for *displaying the results*. Some very useful ways to get a quick understanding of the quality of the calculation is plotting the *streamline pattern* or making a plot of the *velocity vectors*. Another useful tool is making a *contour plot of the pressure distribution* over the surface of the UAV. A streamline pattern very different from expectations should make the user suspicious about the quality of the calculation. Bad calculation quality can be tracked all the way back to poor meshing, but might as well have to do with settings in the pre-processor, which makes problem-solving a time consuming process for which a lot of experience is required.

It soon became clear that simplifying the model by replacing the propellers by disks of the same size and located in the same place wasn't as easy as it looked. Subtracting the volume of the disks from the control volume, just like had been done with the UAV geometry, would create a boundary for the air flowing through the duct, instead of the driving force sucking air into the duct that the propellers would be in real life. Because there are different approaches to deal with this problem, the report treats two cases, each containing a different strategy for modelling the propellers. Chapter 2 deals with the first case, while chapter 3 deals with the second case. Those chapters will incorporate the approach that was explained in this chapter.

An overview of the activities that have been performed during the internship period can be found in **Attachment B**.

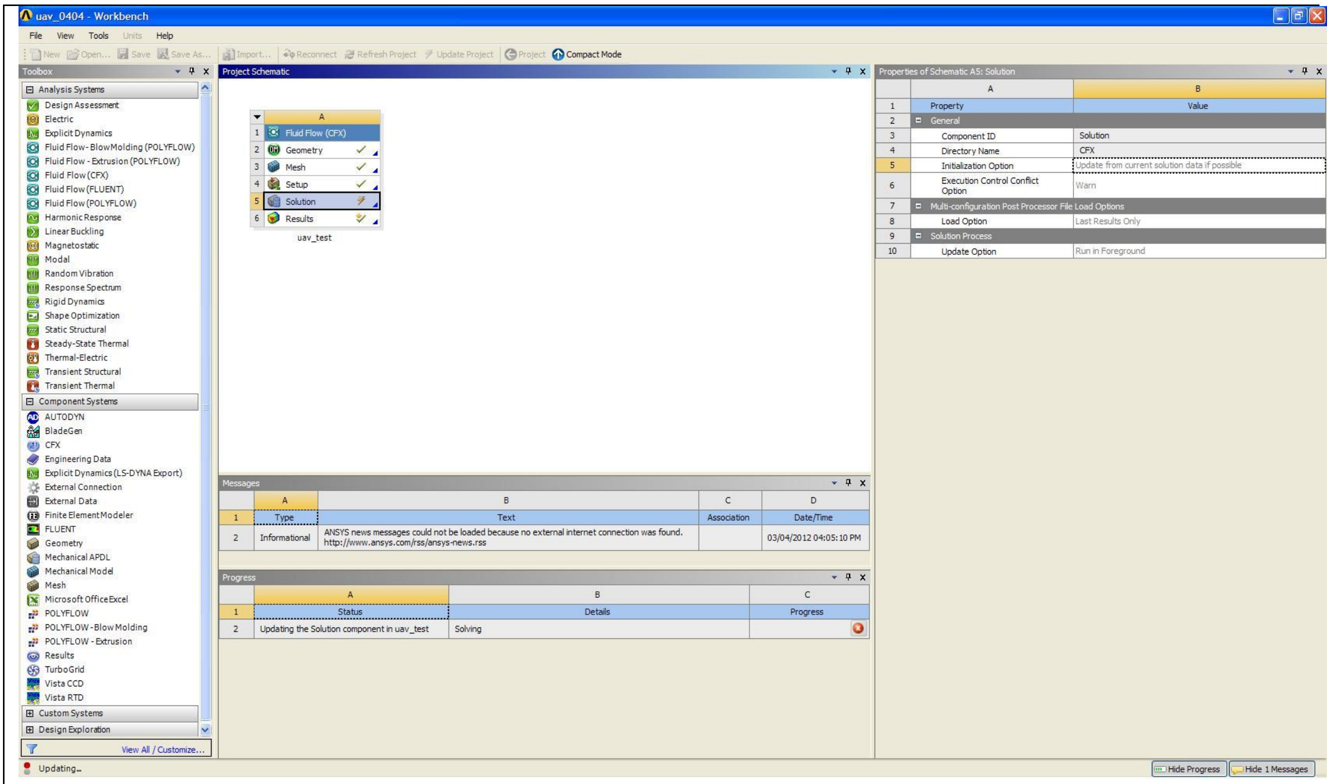

Figure 2: ANSYS Workbench

Chapter 2: Analysis of Case I, Replacement Propeller by Rotating Fluid Domain

This chapter gives the implementation of the approach sketched in Chapter 1 for the case of modelling the two propellers by two rotating flow domains.

2.1.1. Creating the Geometry - Simplification

Because the original SolidWorks geometry was causing problems during mesh generation due to complicated surfaces, it needed modification. Especially the winglets, ailerons and the thin shape of the rotor blades kept generating meshing errors. Probably the surface angles were too large. In the end the winglets where left out, just as the ailerons. The space that was left after deleting these entities was filled with material. The rotors where replaced by disks inside the duct, the size of the original rotors. Because the propeller blades, together with the hub, form one part, deleting the entire part would cause the fuselage to have a gap, like is displayed in **Figure 3**.

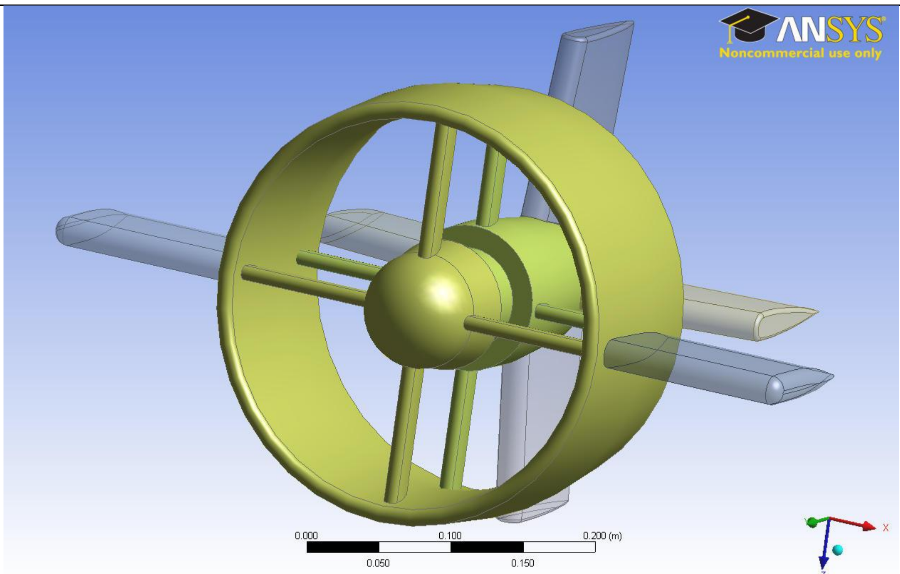

Figure 3: Gap as a result of leaving out the propeller parts

This has been taken care of by changing the propeller parts in SolidWorks, deleting only the propeller blades, without making adjustments to the hub. The result of this operation can be seen in **Figure 4**.

The drawback of leaving out the ailerons is that only the situation of horizontal or vertical flight can be analysed, so analysis of flow past the UAV while performing turns is excluded. This is something to worry about at a later stage (if at all), because for now modelling of the propellers is challenging enough and should be the main focus of the project.

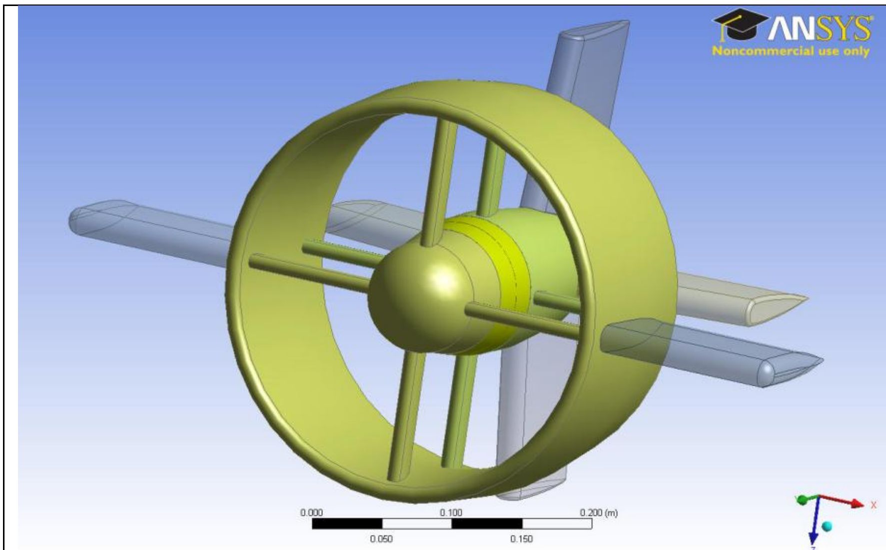

Figure 4: Gap filled by new propeller components

2.1.2. Creating the Geometry - Control Volume and Propeller Disks

Now that the SolidWorks model has been exported to ANSYS, it is time to create the control volume that contains the air surrounding the UAV.

Remark - *All imported and newly created geometry needs to be added as 'Frozen Solid' which makes it possible to have overlapping geometry in the model. This is useful when Boolean operators are being used. Unchecking the option 'Process Surface Bodies' while importing geometry, prevented a problem in the meshing section from happening for reasons that are not really clear. Probably it has got something to do with different surfaces of the UAV not stitching together nicely.*

Because the shape of the UAV is roughly cylindrical, the same shape has been chosen for the control volume. As a general rule, the dimensions of the control volume should be 10 times larger than the dimensions of the object. Because the diameter of the Duct, that forms the main component of the UAV, equals 30 cm, the diameter of the surrounding cylinder should be 3 m. In order to keep computation times short and because of the fact that the computations need to be performed with an educational version of the software that is limited to a maximum number of elements, an outside diameter of 0.7 m has been chosen. The length of the cylinder totals 2.5 m, of which 1/5 is located in front of the UAV and 4/5 behind. The reason for this is that the flow will be disturbed by the object, so flow patterns that are of interest are to be expected at the back of the UAV.

After the cylinder has been created, overlapping volumes need to be deleted by using a Boolean operator. By selecting the surrounding cylinder and the geometry of the UAV at the same time and subtracting the shape of the UAV from the cylinder, the air surrounding the vehicle is all that remains (**Figure 5**).

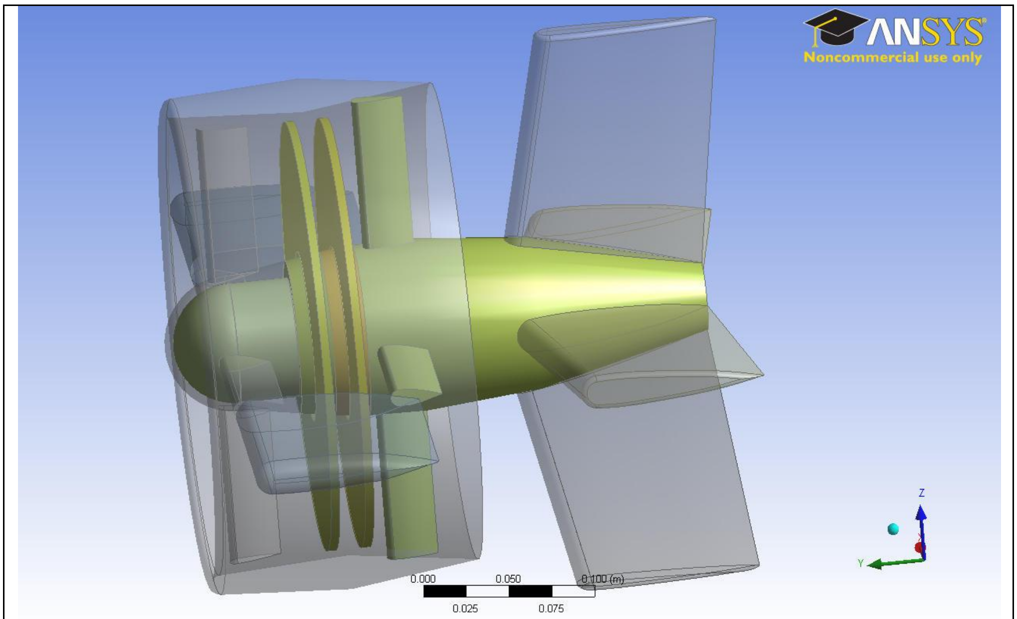

Figure 5: Hollow space that is left after subtracting the UAV geometry from the control volume. Highlighted are the Fuselage and the Propeller disks

A second and third domain needs to be created for the propellers, because the original propeller geometry was too complicated. The propellers will be replaced by two disks with holes in the middle, to make them fit around the Fuselage of the UAV. The outside diameter of the disks equals 0.12 m and the inner radius is just large enough to *prevent interference* with the Fuselage. The thickness of the disks equals 0.005 m and they are right in the centre between the front and back end of the hub to which the propellers used to be attached. The geometry created this way also needs to be subtracted from the control volume, just as the geometry of the UAV. Because this zone should not be considered as a solid object, the tool body of this Boolean operation should be kept and assigned the fluid property. At this point 3 fluid zones can be distinguished: the fluid zone surrounding the UAV and part of the inside of the duct, and the (much smaller) cylindrical zones at the location where the propellers used to be. At the interfaces between those three zones, this generates adjacent surfaces that are actually in the same place, but by ANSYS CFX are considered to be two different boundaries. This creates a challenge during problem set-up that will be solved later on.

2.2.1. Meshing - Introduction

The geometry has been created in SolidWorks, and was modified and expanded in ANSYS DesignModeller. Now it's time to create a mesh in ANSYS Mesher.

2.2.2. Meshing - Method

During meshing, the volume of the geometry will be subdivided into many different parts, or elements. Those elements can take on different shapes, of which *tetra- and hexahedral shapes* are the most common ones. A hexahedral element structure requires a lot of input from the user, because it needs a skilled person to make sure the elements fit into the geometry. A tetrahedral mesh structure is more forgiving when it comes to complicated shapes because it just fills up the space using randomly shaped (with respect to element size) elements. This is the faster method of the two and can be carried out *automatically*, but the mesh might be less efficient and may contain more elements than would be strictly necessary. In this case automatic generation of a tetrahedral mesh has been used, in combination with a *patch conforming algorithm*. Creation of a handmade hexahedral mesh alone would take a few weeks, which would leave no time for running analyses and interpreting results.

2.2.3. Meshing - Sizing

What the user *can* do is have influence on the element size in different regions of the geometry. This is useful, because areas of special interest can be given a finer mesh with smaller element sizes, to make the solution at these points describe the real situation more accurately. Especially regions in the proximity of the UAV Surface and in the slip stream behind the vehicle require special attention, because the fluid-solid interaction at these points leads to *high gradients in both velocity and pressure*.

In order to achieve mesh refinement near the surface, *sizing control* has been applied. Parameters that need to be set by the user are the *element size at the scoping surface* and the *growth ratio*. By selecting a value for the growth ratio, one can select the speed of growth of the elements to or from a surface. A higher number for the growth ratio will result in faster growing element sizes and a lower total number of elements. This can be useful when the mesh contains too many elements for the solver to terminate correctly.

An element size of 0,008 meters has been chosen, starting at the surface of the UAV and expanding towards the boundary of the control volume with a growth rate of 1.2 which is the default value.

2.2.3. Meshing – Statistics

Table 3 contains some parameters that are part of the *mesh statistics*. Those are the minimal element size, the maximum element size, the number of nodes and the number of elements.

Parameter	**Value**
Minimal element size	1.5127e-003 m
Maximal element size	0.3025e 0 m
Nr. of nodes	718523

Nr. of elements	513115

Table 3: Mesh Statistics, Case I

2.2.4. Meshing - Remark

Before the mesh generator successfully created the mesh, errors of all sorts occurred and were solved later on, of which the most persistent one had to do with the amount of elements in the model. An understanding that pretty much solved this issue was the *redundancy of a volume mesh for the UAV*. After all, this research was about the flow *around* the UAV and not about the mechanical behaviour *of* the UAV, like vibrations for example. This way of thinking saved a lot of computation time, because the geometry of the UAV contained a lot of elements due to its wavy surfaces. Now, the surface resulting from cutting out the UAV geometry from the surrounding cylinder, acts as the UAV Surface.

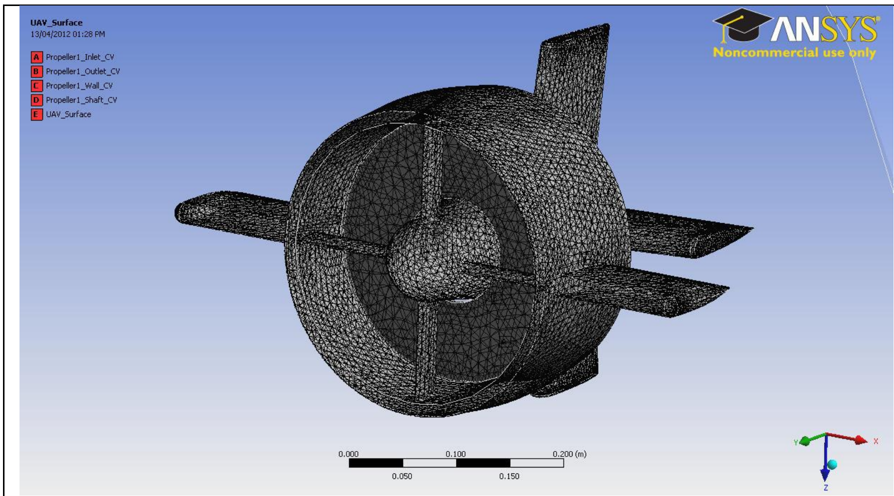

Figure 6: Control volume mesh; Only elements on the domain boundary are displayed. Propeller disks are a boundary to the control volume.

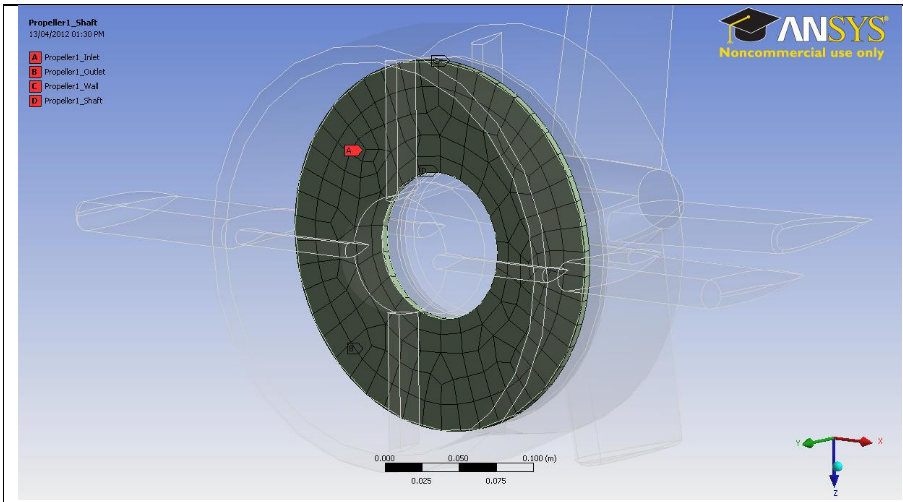

Figure 7: Same propeller surfaces are a boundary to the propeller domain too.

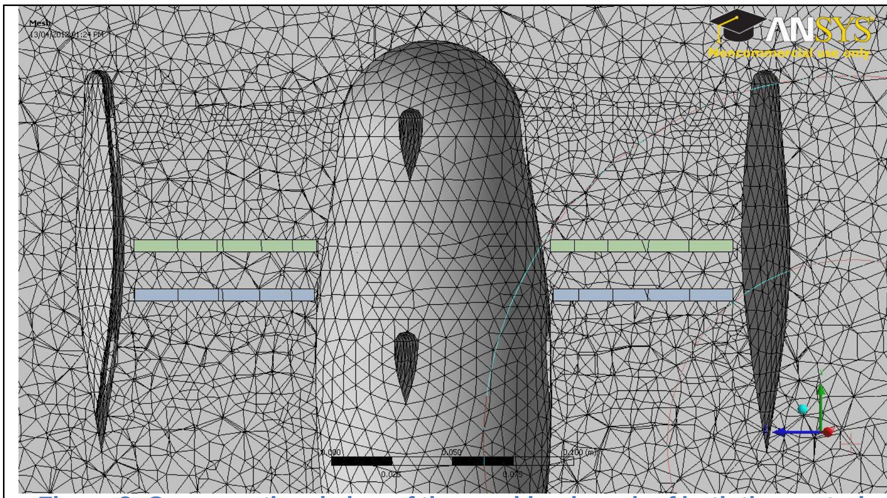

Figure 8: Cross-sectional view of the combined mesh of both the control volume and the Propeller disks

2.2.5. Meshing - Creating Named Selections

In order to more easily apply the boundary conditions during pre-processing, it's to be advised to give names to the different surfaces that are present in the model. For this reason, the circular surface on the front end of the surrounding cylinder is defined as 'Inlet', the cylinder wall as 'Wall' and the circular face on the back end as 'Outlet'. Between inlet and outlet, the flow goes past the 'UAV_surface'. Because, at the location where the propellers used to be, two disks where cut out, the control volume has another 8 faces. Each cut-out disk leaves 4 surfaces, named 'PropellerX_Inlet_CV', 'PropellerX_Wall_CV', 'PropellerX_Shaft_CV' and 'PropellerX_Outlet_CV'. The addition _CV is present because the disks themselves also have 4 surfaces: 'PropellerX_Inlet', 'PropellerX_Wall', 'PropellerX_Shaft' and 'PropellerX_Outlet'. Now that all faces have been given names, the solution can be set up in the ANSYS pre-processor.

2.3.1. Pre-processing - Introduction

The pre-processor is used to create the problem setup. This paragraph will explain which settings were used for the calculations of Case I.

2.3.2. Pre-processing - Steady State or Transient

One of the first steps in setting up an analysis case is to determine whether the flow is *steady state or transient*. A steady state flow does not take start-up behaviour into account and can be used for an analysis in which the UAV is considered to be in steady flight. When, during the calculation the solution does not seem to converge but convergence indicators like the Root Mean Square are oscillating, this can be an indicator for transient behaviour. When the time step is being changed, but the period of the oscillatory movement stays the same, it is a transient effect. This effect was not detected during the calculations, from which it can be concluded that the steady state option works just fine.

2.3.3. Pre-processing - Thermal Model

Another characteristic of the model that needs to be specified is the *thermal property*. There are different options: *none, isothermal, thermal energy and total energy*. Leaving out the heat transfer calculation completely from the governing equations will make the time required by the CFX Solver a lot less, but seems to be a bit drastic. In the future, heat transfer might be subject to research, so it is better to include this in the model. For now the *isothermal* heat transfer option seems to be sufficient. It requires a *uniform temperature* for the whole domain. A value of 293 K has been chosen, equal to room temperature. For flight at high altitude, the temperature needs to be changed to a lower value. The isothermal model doesn't require as much computational power as the thermal energy model and the total energy model do, but does take into account changes of fluid properties as a result of temperature fluctuation like for example density.

2.3.4. Pre-processing - Domain Specification

The GUI in ANSYS CFX shows the geometry that was defined in the DesignModeler. In the tree-outline, an overview is given of the different surfaces and volumes. Although volume information is passed on from earlier stages, *domains* still need to be defined at this point. The control volume domain and the propeller domains are defined successively. The propeller domains are a bit special: those are rotating. The speed of revolution equals the rotational speed of the propeller blades, of which the value is not exactly clear from earlier reports. For this reason different speeds have been tried in order to see the effect on the streamline pattern. In theory the second propeller should be rotating in the opposite direction compared to the first one and at the same time should have a higher rotational speed in order to make sure the outflow from the first propeller doesn't encounter a barrier at the inlet of the second propeller.

After a domain has been defined, the properties for this domain need to be picked. **Attachment A** contains tables with the properties of the three domains present in this model and an explanation for why those values have been picked.

Next, all *domain boundaries* have to be specified, an overview of which is provided in the **Table 1**. The Greek alphabet

Domain: Control volume	Domain: Propeller1	Domain: Propeller2
Boundary: Inlet	*Boundary*: Propeller 1_Inlet	*Boundary*: Propeller 2_Inlet
B: Outlet	B: Propeller1_Outlet	B: Propeller2 _Outlet
B: Wall	B: Propeller 1_Wall	B: Propeller2 _Wall
B: UAV_surface	B: Propeller 1_Shaft	B: Propeller2 _Shaft
B: Propeller1_Inlet_CV		
B: Propeller1_Outlet_CV		
B: Propeller1_Wall_CV		
B: Propeller1_Shaft_CV		
B: Propeller2_Inlet_CV		
B: Propeller2_Outlet_CV		
B: Propeller2_Wall_CV		
B: Propeller2_Shaft_CV		

Table 1: Boundary Names

2.3.5. Pre-processing - Domain Interfaces

As can be seen, the boundaries of the propellers are counted twice, so *domain interfaces* are needed at these points, merging two boundaries to one domain interface. Eight domain interfaces are created in this way. It is obvious that the Fluid-Fluid type of interface needs to be selected. Next, the interface model needs to be chosen. This model defines the way the solver models flow physics across the interface. There are three options: *translational periodicity, rotational periodicity and general connection*.

The periodic options are used when just one section is being analysed that is part of a structure with multiple identical regions, like for example one rotor and one stator in a turbine. Although modelling of the propellers looks a lot like modelling a turbine, the propellers are being analysed as part of the UAV, which changes the scale of the problem resulting in the absence of repetitive elements.
In the description of the general connection model it is said that this model can be used for: “Connecting non-matching grids” and “Applying fully transient sliding interfaces between domains”.

The first property is useful, because the grid on the control volume side of the propeller disks is much more refined than the mesh on the propeller side. The usefulness of the second property seems obvious. As an example for applying the general connection type of domain interface is given: "Two sides, of which one is in a stationary frame of reference and the other side is in a rotating frame of reference." This makes perfect sense.

2.3.6. Pre-processing - Change/Mixing Model

After the interface model has been chosen, the *frame change/mixing model* has to be picked. Again, there are different options to choose from: *Frozen Rotor, Stage* and *Transient Rotor Stator*.

When *Frozen Rotor* is selected, the frame of reference and/or the pitch is changed but the relative orientation of the components across the interface is fixed. This seems to be the case, because although the disks are rotating the relative position of the disks with respect to the UAV stays the same. The modelling guide of ANSYS CFX describes the Frozen Rotor model as the model requiring the least amount of computational effort of the three methods. Disadvantages of the model are that errors occur when the quasi-steady assumption doesn't apply (steady conditions are assumed) and that losses incurred in the real (transient) situation as the flow is mixed between stationary and rotating components is not modelled.

Those losses are, in fact, included in the *Stage Model*. A one-time mixing loss is included for every stage. This loss is equivalent to assuming that the physical mixing supplied by the relative motion between components is sufficiently large to cause any upstream velocity profile to mix out prior to entering the downstream machine component. The stage model seems to be appropriate when a repeating geometry in analysed, like the rotor and stator of a turbine wheel. This case is very different, because the propeller is modelled as a whole, which is a major simplification compared to the real world situation. Accepting such a high degree of simplification takes away the need for a frame change/mixing model that is more advanced than the standard Frozen Rotor model. It consumes the least computation time and it looks like it can deal with sliding surfaces and changing reference frames.

In this perspective, the *Transient Rotor-Stator Model* is even more advanced, which can be derived from the description in the CFX modelling guide: *"This model should be used anytime, it is important to account for transient interaction effects at a sliding (frame change) interface. It predicts the true transient interaction of the flow between a stator and rotor passage. It ultimately accounts for all interaction effects between components that are in relative motion to each other."* and *"…if you are interested in simulating a periodic-in-time quasi-steady state, then it may be helpful to first obtain a steady state solution using Frozen Rotor interfaces between components. This solution will contain most of the overall flow features, and should converge to the desired transient simulation in the fewest transient cycles."* For this reason Frozen Rotor has been chosen as the frame change model.

2.3.7. Pre-processing - Boundary Conditions

For each boundary, a B.C. needs to be defined, of which an overview is given in **Attachment A**. The following boundary types were used: inlet, free-slip wall, no-slip wall, interface and outlet. At the domain interfaces, pressure jumps have been applied adding up to a total of -300 Pa for each propeller. This value has been chosen based on the force that should be generated by the propellers, but can be changed to another value at any time. Normally the speed of the rotor will cause the pressure difference, so when information is available about the correlation between rotor speed and pressure build-up, the pressure-jump can be changed to the right value.

2.3.8. Pre-processing - Remark

At this point a shortcoming of the software has to be discussed, that has to do with the need for creating physical, 3-dimensional geometry, in order to define boundary conditions.
Simplifying a rotor by a disk with finite thickness (so no 2D within a 3D model), results in two boundaries when only in- and outflow areas are considered, while a pressure jump would require only one boundary. This can be solved by splitting the total pressure jump in two parts: 1 at the inlet and 1 at the outlet, each of them half the size of the total pressure jump for the propeller. At first sight this seems to have the exact same effect as defining just one pressure jump over a 2D surface, but it is an elaborate way of modelling.
In order to avoid this, another configuration has been tried, in which the two separate disks have been swopped for one rotating cylinder, occupying the volume of the two disks and the space in between them (**Figure 9**). This way the rotor domain now consists out of just one geometrical shape, with just one inlet and one outlet. One could now apply a pressure jump on the front end of the domain (first propeller) and another pressure jump on the back end of the domain (second propeller). A drawback of this model is that there is only one rotor domain, so counter rotation is not an option anymore. For this reason no further effort was put into this concept, but *investigating a way to define 2D surfaces within a 3D domain could be an interesting subject for future research*.

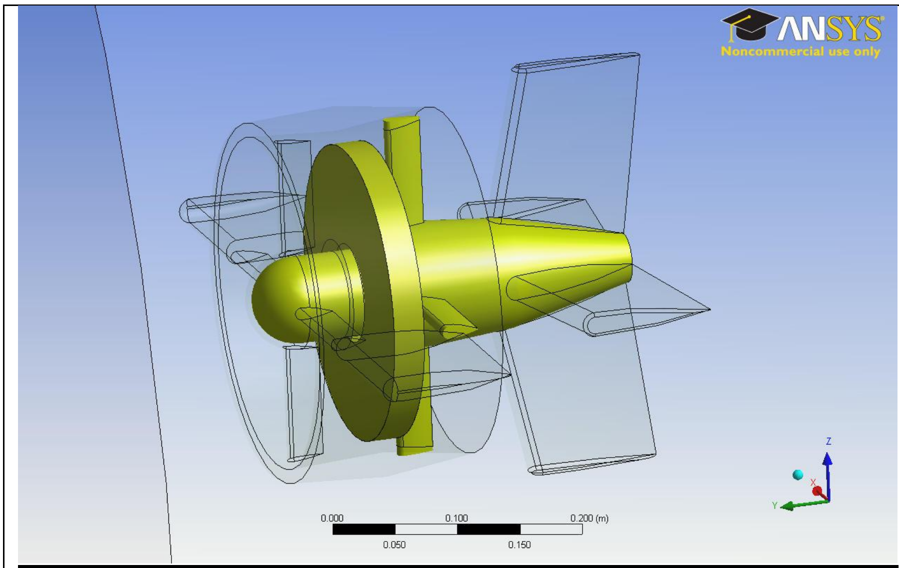

Figure 9: Single Rotor Domain

2.3.9. Pre-processing - Solver Control

Solver control allows the user to set convergence criteria in order to make sure the solution will terminate, whether this is due to convergence or due to user-defined limits. A list of parameters that have been fixed is shown in **Table 2**.

Solver Control Parameter	Value
Advection Scheme	High Resolution
Turbulence Numerics	High Resolution
Convergence Control	Min. Iterations: 1; Max Iterations: 15000
Timescale Control	Auto Timescale
Length Scale Option	Conservative
Timescale Factor	1.0
Convergence Criteria	RMS Residual Target: 1e-06
Domain Interface Target	0.01 (Recommended when working with domain interfaces)

Table 2: Solver Control Parameters

2.4.1. Results - Introduction

During the *final stage* of the project, the approach in which a *rotating flow domain* was used seemed to be giving the *most realistic* results. As a consequence, more data was obtained using this method than using the method described in Chapter 3. This paragraph will show results that were obtained using different settings, in order to get a better understanding of how the model works. Each configuration, using different settings, has been assigned a character ranging from **A** to **E** as shown in the **Table 3**.

	Turbulence Model	***Inlet B.C.***	***Ω Rotor 1***	***Ω Rotor 2***	***Δp Rotor 1***	***Δp Rotor 2***
Config. A	Shear Stress Transport	Inlet, Velocity Normal to Boundary: 1 ms^{-1}	16000 rpm	18000 rpm	-2.5 bar	-2.5 bar
Config. B	Shear Stress Transport	Inlet, Velocity Normal to Boundary: 1 ms^{-1}	16000 rpm	18000 rpm	-2.5e-3 bar	-2.5e-3 bar
Config. C	SSG Reynolds	Inlet, Velocity Normal to Boundary: 1 ms^{-1}	16000 rpm	-18000 rpm	None	None
Config. D	SSG Reynolds	Opening, Relative Pressure = 0 Pa	17000 rpm	-18000 rpm	-3.0e-3 bar	-3.0e-3 bar
Config. E	SSG Reynolds	Opening, Relative Pressure = 0 Pa	17000 rpm	-17000 rpm	-3.0e-3 bar	-3.0e-3 bar

Table 3: Configurations

2.4.1. Results - Configuration A

Figure 10 shows values of the Root Mean Square for the properties Mass and Momentum as a function of time. The y-axis, displaying the RMS, runs from 1.0e-6 towards 1.0e00. The x-axis, displaying the Accumulated Time Step or the amount of iterations, runs from 0 to just over 5000. As the calculation proceeds, this axis is being extended towards the amount of iterations completed at that time.

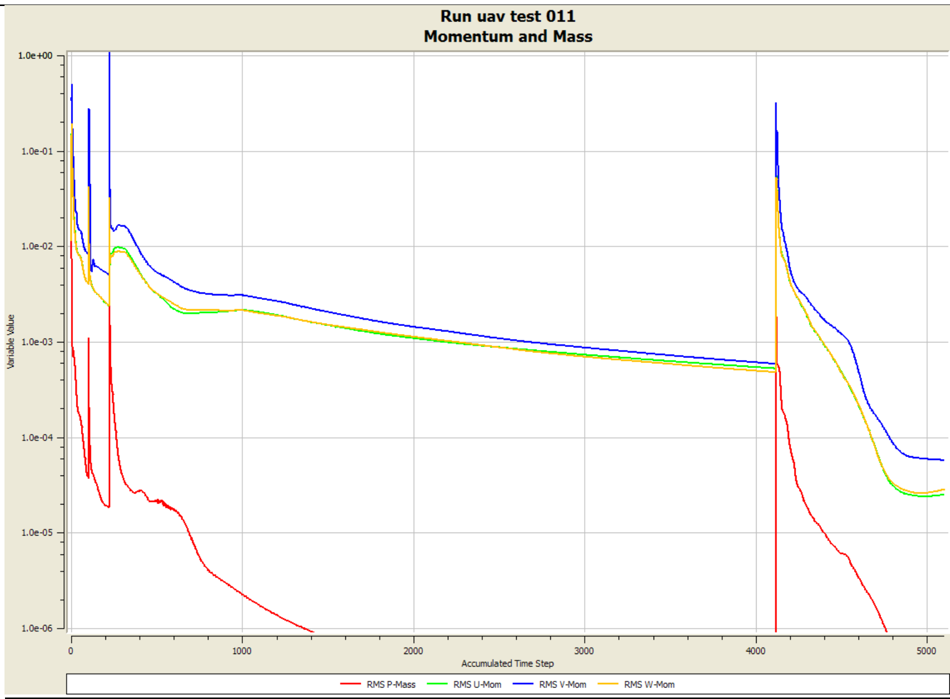

Figure 10: Root Mean Square vs. Time Step, Config. A

The graph is composed out of 4 runs, as can be seen from the steep peaks characterizing the beginning of a new calculation. This happens each time when parameters in the pre-processor have been changed. At this point it is repeated that the value for the RMS is an indicator for the *degree of convergence* of the solution. All solutions had to be terminated *before* the convergence criterion was reached, which was set at 1.0e-6. This value was never reached, as can be seen from the third run in the first picture. The part between 1000 and 4000 iterations took one night of calculating, performed on an ordinary desktop computer. Waiting for the RMS to eventually get below 1.0e-6 would take an *unreasonable amount of time*. Especially in a stage in which it is not even certain whether or not the model is correct.

Figure 11 shows the streamline pattern emerging from the Outlet surface of Propeller 1. Plotting *streamline patterns* is always a good way to get an idea of the *quality* of the model. In case the streamline pattern differs a lot from the results expected, it is likely that there is a problem with the boundary conditions or some of the other settings.

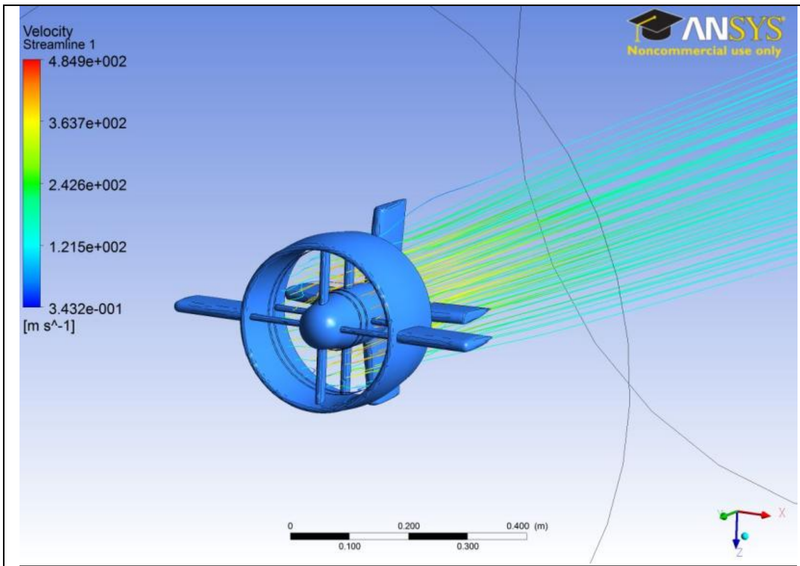

Figure 11: Streamline Pattern, Emerging from propeller 1, Config. A

Local flow velocities as low as 0 ms^{-1} and as high as 485 ms^{-1}(!) are observed. Because of the no-slip boundary condition on the surface of the UAV, air particles at this location will have zero velocity. The same is true for air at the stagnation points on the nose of the UAV and the front surfaces of the duct, the wings and the control surfaces. The highest velocities present in the flow are located inside the Duct, close to the Propeller disks. When the air has passed the UAV it quickly decelerates towards speeds varying from 240 ms^{-1} at the centre of the outflow, towards 120 ms^{-1} on the sides.

A remarkable thing is the *absence of swirl* in the flow passing through the Propeller area. Although the first Propeller is rotating with 16000 rpm and the second propeller with a speed of 18000 rpm in the same direction, the flow seems to be unaffected.

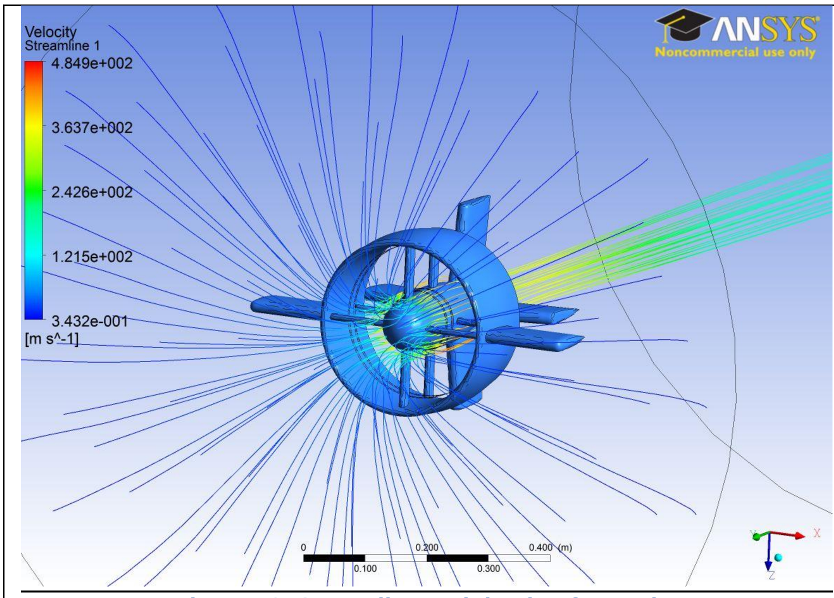

Figure 12: Streamlines originating from Inlet

When the streamlines are plotted that originate at the inlet face of the control volume (**Figure 12**), it soon becomes clear what is going on. Air particles way out of reach for the Duct entrance to be able to catch them, are sucked in anyway pointing out the fact that the pressure jump over the propellers is probably way too high. In order to prove this hypothesis the pressure jump will be lowered in Configuration B, which will be discussed next.

A pressure jump of -5 bars in total caused the flow to accelerate very fast after which it collides with the surface of the UAV. **Figure 13** shows the distribution of relative pressure over the surface of the UAV. Pressures as high as 7.882e004 Pa and as low as -1.970e005 Pa are being observed.

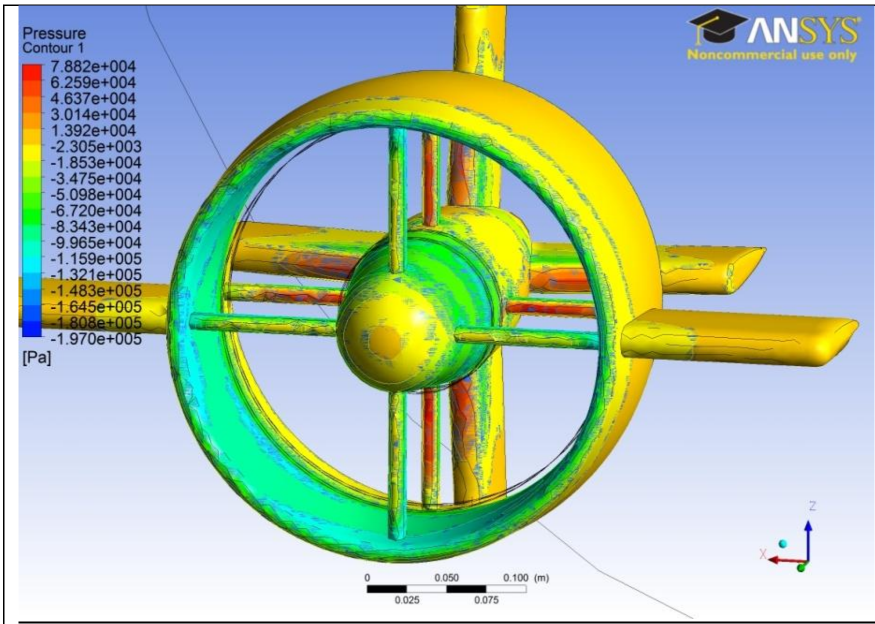

Figure 13: Relative Pressure Distribution over the UAV Surface, Config. A

Surfaces in the outflow area of the propellers are subject to high pressures because they act as stagnation points to the flow which is both accelerated by the pressure difference and by the rotation of the propellers.

At the front of the UAV one would expect the surface pressure to be a little higher than is the case here. The reason for this is that under normal flight conditions, the vehicle itself would have a speed compared to the stationary air surrounding it, instead of the air being forced towards the UAV. This causes the streamlines to bend towards the propeller inlet surface. As a result the air will not collide with for example the nose of the UAV head-on and the surface pressure at this point will be lower than in reality.

Figure 14 shows a vector plot of the airflow. A remarkable detail that was not visible in the streamline pattern is that the velocity vectors close to the outside surface area of the duct are pointing towards the front. This, of course, cannot be true in a real situation and probably has to do with the pressure boundary condition.

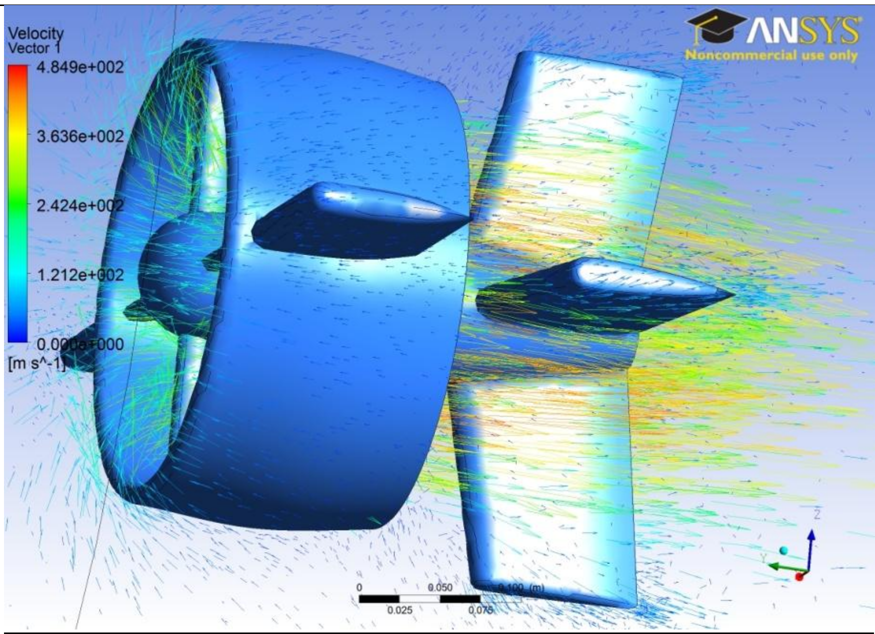

Figure 14: Velocity Vector Plot of the Air Flowing Around the UAV, Config. A

It is obvious that the pressure jump over the propellers has to be changed. In case this would solve most of the problems with the unrealistic streamline pattern, other variables might be changed too.

2.4.2. Results - Configuration B

Configuration B uses the same boundary conditions as Configuration A does, except for the fact that a lower pressure jump is applied to the propeller areas. **Figure 16** shows the solver run in terms of the Root Mean Square values as a function of time. 5 Consecutive runs are displayed, of which the last one is the run using Configuration B. Coarsely 4500 iterations were completed, before the calculation was terminated on user's request. Although the values for the RMS are still slightly decreasing, in-between checks of the results showed little change between current results and results obtained a couple of hundreds of iterations before. The RMS target of 1.0e-06 was out of reach anyway, within a reasonable time-span.

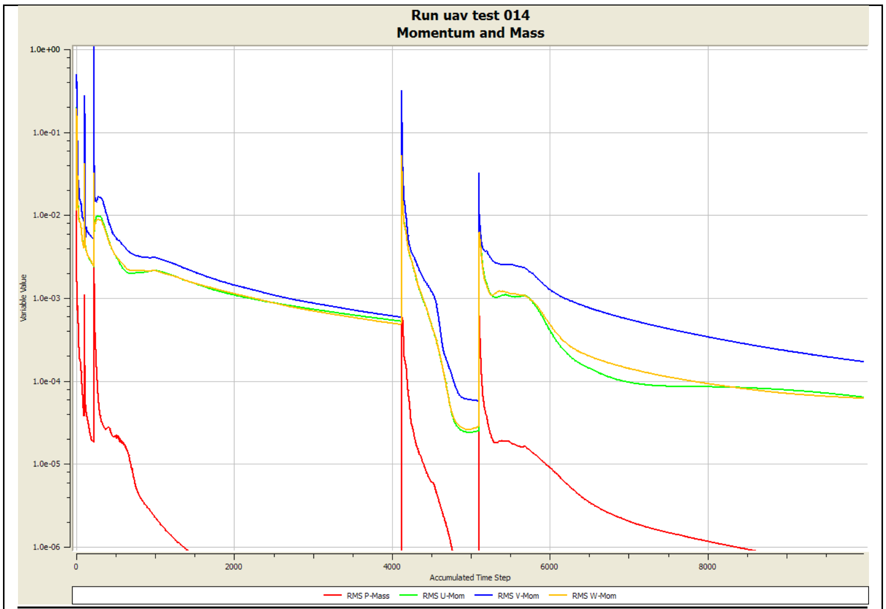

Figure 15: Root Mean Square vs. Time Step, Config. B

Figure 17 contains the streamline pattern originating from the second propeller. The pattern looks promising, because streamlines are not coming out of the duct as straight lines anymore, but contain a bit of *swirl* indicating that the rotating propeller domains are having their intended effect. Because both propellers are rotating in the same direction the flow has a deviation to the left side as seen from the front of the UAV, so this is no surprise.

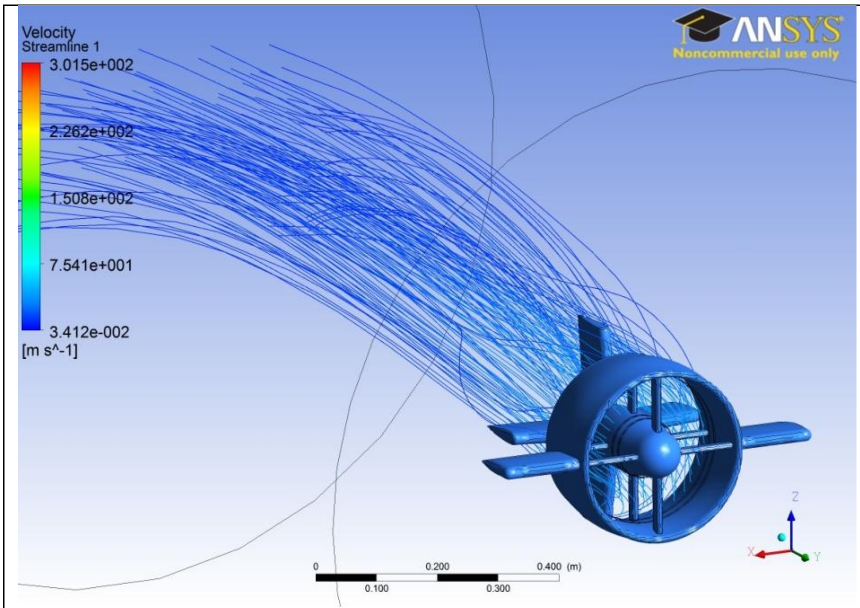

Figure 16: Streamline Pattern, Originating form the Second Propeller, Config. B

The streamline pattern shown in **Figure 18** was obtained by making the inlet surface of the control volume the starting point. In contrast with the previous picture, this one is a bit worrying because some streamlines make a few loops within the control volume domain before leaving through the outlet surface, as if they are kept inside a box. It is hard to tell whether it is worth the effort to find out what causes this behaviour, because there is a good chance that it will be solved by a higher amount of iterations. In the end, this configuration was intended to check whether lowering the pressure jump would introduce swirl to the flow and that assumption has been proved. Knowing this, it is better to fully adapt the model to match the theory, like for example changing the turbulence model, before putting too much effort in this configuration. At the point of turbulence modelling, theory advises to use the SSG Reynolds model, so Configuration C will implement this setting and the strange streamline pattern might just not be observed anymore.

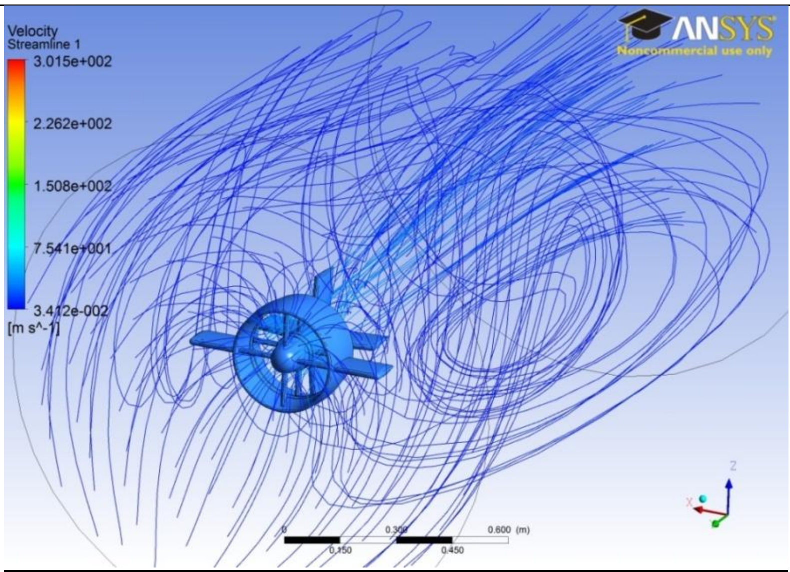

Figure 17: Streamlines, Emerging from control volume Inlet, Config. B

Figure 19 contains the *pressure distribution*. It can be seen right away that the surface pressures are a lot lower than they used to be in the previous case. Compared to Configuration A no unrealistic, lower than 0, pressures are detected anywhere on the surface anymore. The lowest pressures detected are 3.872e003 Pa below the reference value, which is atmospheric pressure. The highest pressures are 1.786e003 Pa above. As far as the qualitative aspect of the pressure distribution concerns, the same can be said as in Configuration A: although the absolute values are different, zones of high pressure are located on the front edges of the control surfaces and on the stator surfaces.

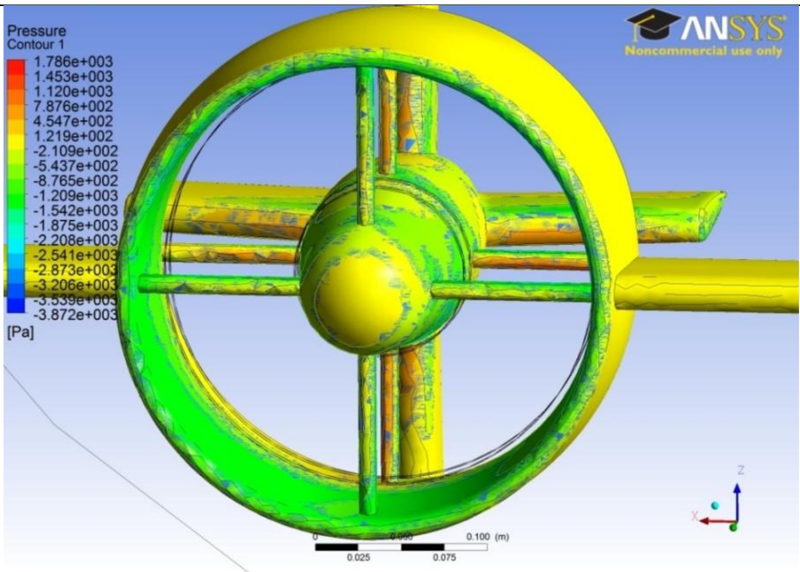

Figure 18: Relative Surface Pressure Distribution, Config. B

Figure 20 shows a velocity vector plot in which emphasis was put on the velocity introduced by the propeller domain. Only the velocity vectors of the domain of Propeller 1 and the control volume are shown. The vectors of Propeller 2 would be pointing in the same direction, but would be longer as an indication of higher velocity. Vectors positioned on the outside of the duct no longer point in opposite direction as was the case with Configuration A.

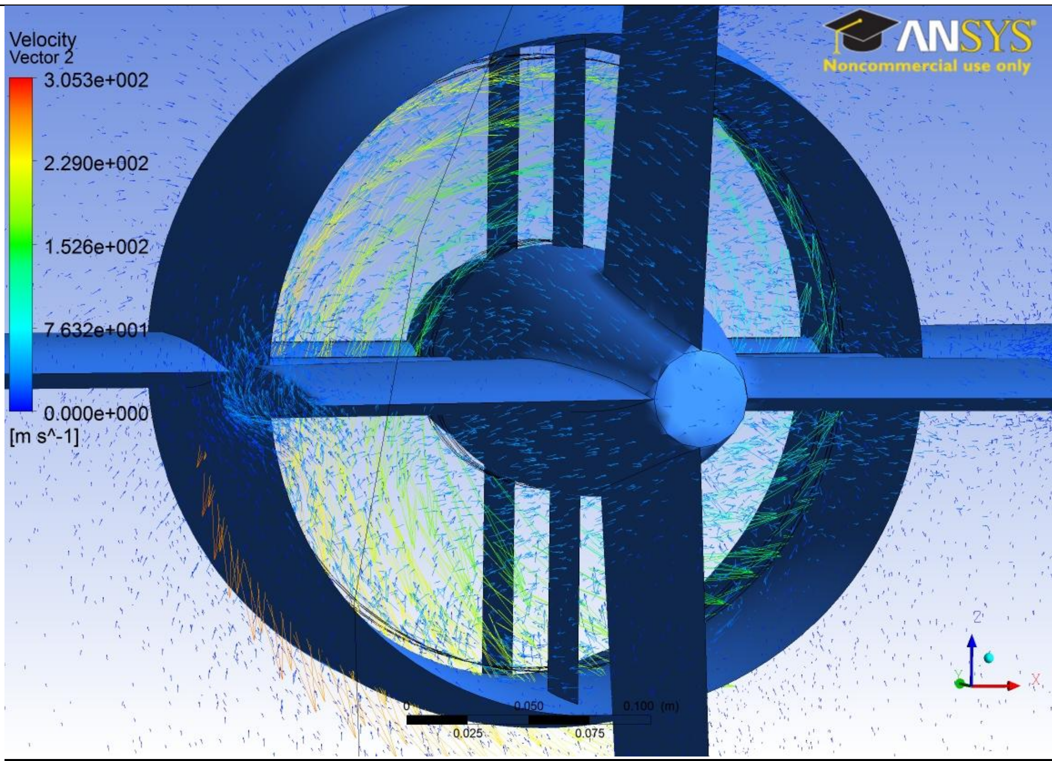

Figure 19: Velocity Vector Plot of the Air Flowing Around the UAV, Config. B

Configuration B is clearly an improvement compared to the results obtained before. Now that the reason for not observing any swirl in the flow behind the UAV has been clarified, the model can be changed to match all settings that were intended initially like for example using the SSG Reynolds turbulence model. Also the inlet boundary condition of the control volume, the pressure jump over the propellers and the rotational direction of the propellers will be changed in up-coming configurations.

2.4.3. Results - Configuration C

Configuration C is different from previous cases in more than one way. First of all, the *turbulence model* has been changed to SSG Reynolds Stress for reasons explained in the pre-processor part about choosing the right turbulence model. Secondly, the concept of applying a pressure jump across the propeller domains was dropped, just to see what would happen. A third parameter was changed compared to Configuration B, which is the rotational direction of the second propeller domain. This domain is now counter rotating compared to the first propeller.

Figure 21 shows that this calculation has been running for quite some time. After 3 earlier runs, a run with Configuration C was performed for as long as 20000 iterations. Much more than any of the configurations before, but still not anywhere near a RMS-value of 1.0e-06!

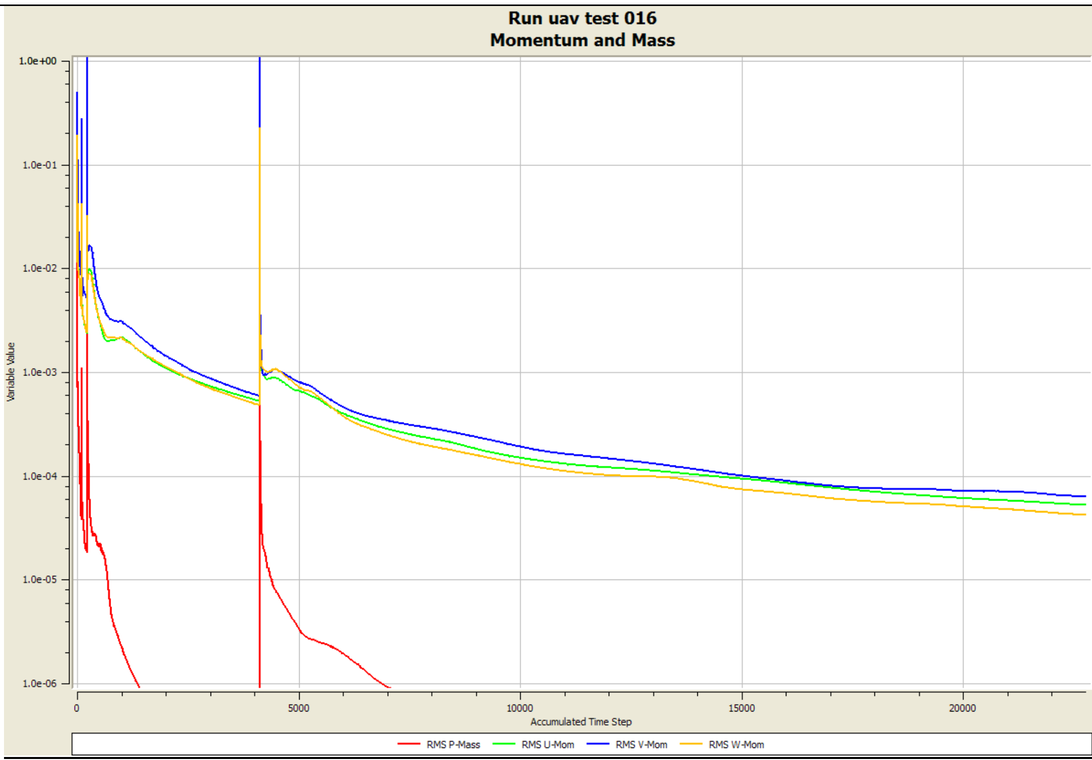

Figure 20: Root Mean Square vs. Time Step, Config. C

The streamline pattern, originating from the second propeller is far from straight when it leaves the duct (**Figure 22**). This result is a little disappointing because the whole project is about finding out whether using counter rotating propellers will cause the flow that leaves the duct to flow past the control surfaces in a straight way. *It looks like the direction of the flow is very much dependent on the direction of rotation of the second propeller and that the fact that the first propeller is rotating in the opposite direction, doesn't make up for that*.

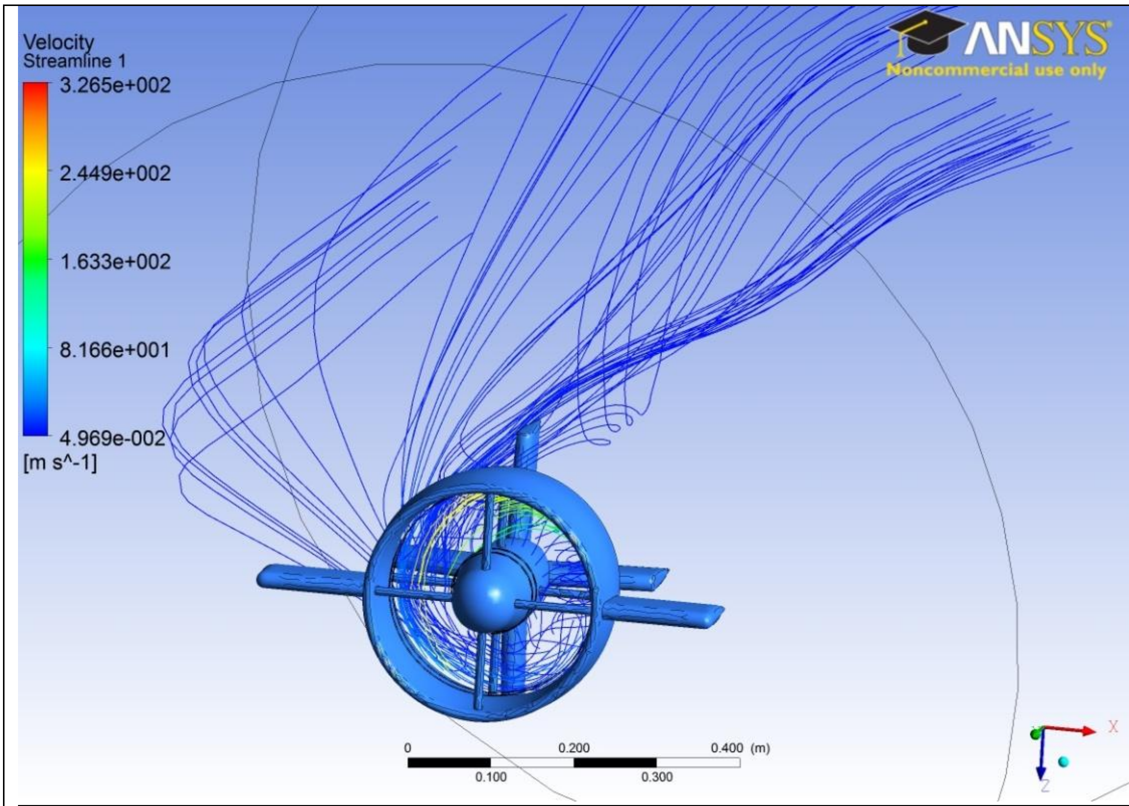

Figure 21: Streamline Pattern, Originating form the Second Propeller, Config. C

Even more interesting is **Figure 23**, displaying the streamlines originating from the first propeller. Some streamlines make it all the way through the second propeller before leaving the duct, but just as many streamlines enter and leave the duct at the front side. This phenomenon is probably caused by the fact that the second propeller causes pressure build-up on its inlet forcing the air particles coming out of the first propeller to move in reversed direction. So, while the second, faster spinning, propeller was intended to make sure the air can flow all the way through the duct, the exact opposite seems to be happening.

Before jumping to conclusions too soon, one has to take into account that the pressure jump over the propeller domain was left out in this configuration, so re-introducing this model parameter might solve the problem of particles moving in the wrong direction.

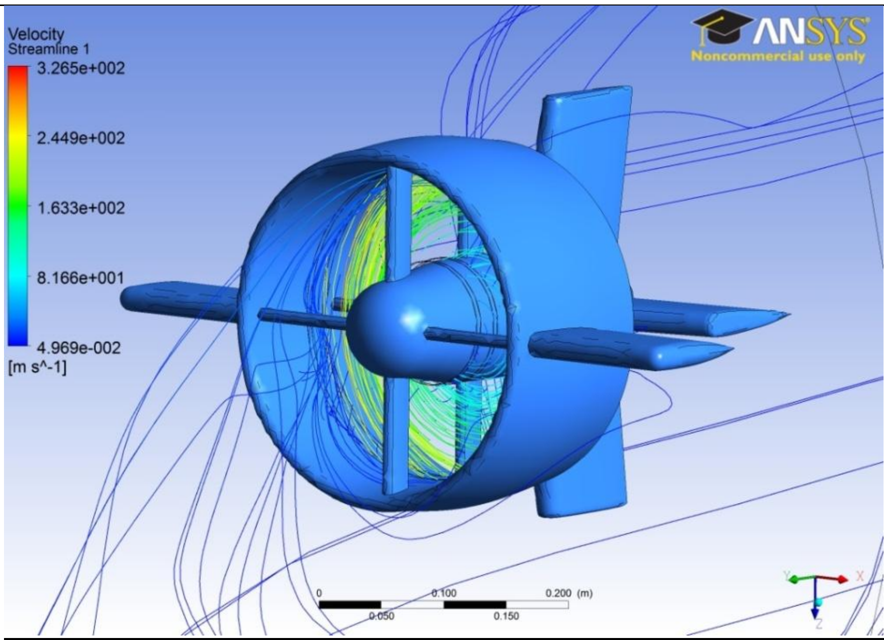

Figure 22: Streamline Pattern, Originating form the First Propeller

The pressure distribution (**Figure 24**) displays an overall decrease in relative pressure on the surface of the UAV as a result of leaving out the pressure jump. The Inlet boundary condition of the control volume is the only factor causing axial fluid velocity, so the pressure applied by the flow on the stagnation surfaces is lot less than in previous configurations.

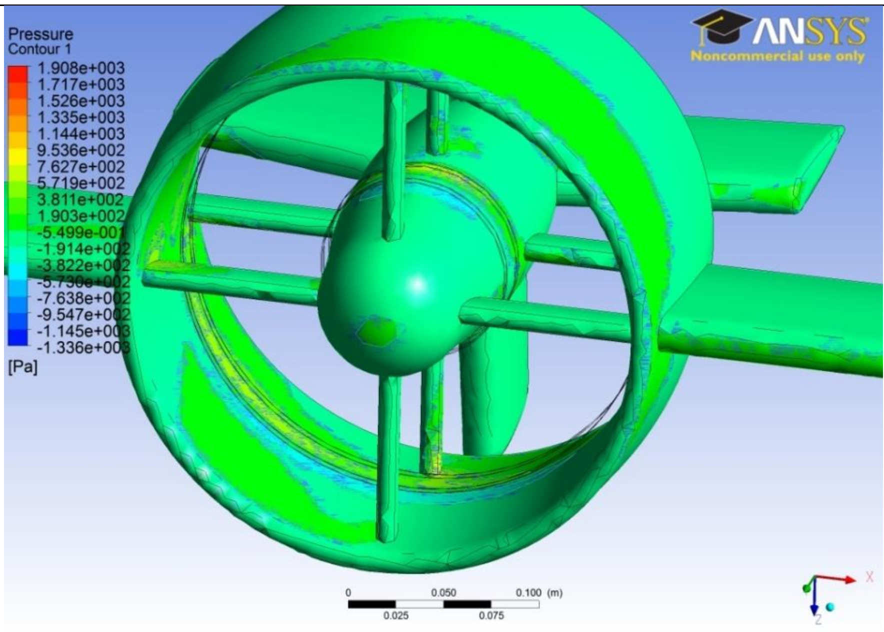

Figure 23: Relative Pressure Distribution over the UAV Surface, Config. C

The last picture (**Figure 25**) again a plot of velocity vectors in the control volume domain. For clarity, 3D arrows were used to emphasize the flow of particles from the inside of the duct towards the front end of the duct. This is something that was seen earlier in **Figure 23**.

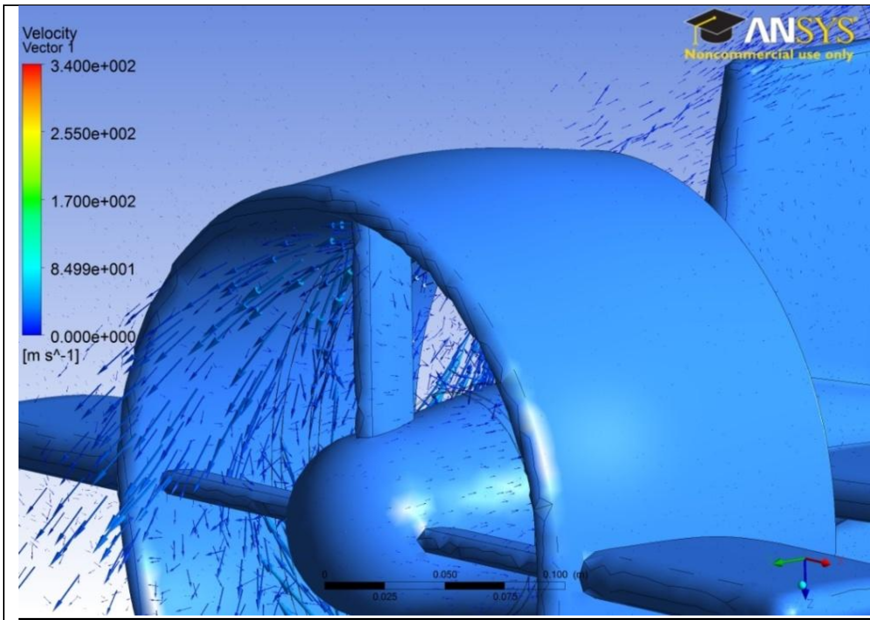

Figure 24: Velocity Vector Plot of the Air Flowing Around the UAV, Config. C

2.4.4. Results - Configuration D

This configuration is intended to get rid of the streamlines that were getting out at the front of the duct in Configuration C and to get the streamlines at the back of the UAV to exit straight. To solve the first issue, the pressure jump was re-introduced in order to catch all the air particles and force them through the duct. This time, the pressure jump equals -3.0e-3 bar for each propeller instead of -2.5e-3 bar in Configuration B, because this value is roughly the amount of pressure needed across the surface of the propellers *in order to lift the weight of the UAV during take-off*. Because at take-off, no lift is generated by the wings or the duct, this is the greatest force that ever needs to be generated by the propellers. In order to reduce the effect of Propeller 2 on the direction of the streamlines, the speed of Propeller 1 was increased to a value of 17000 rpm, just 1000 rpm under the speed of Propeller 2.

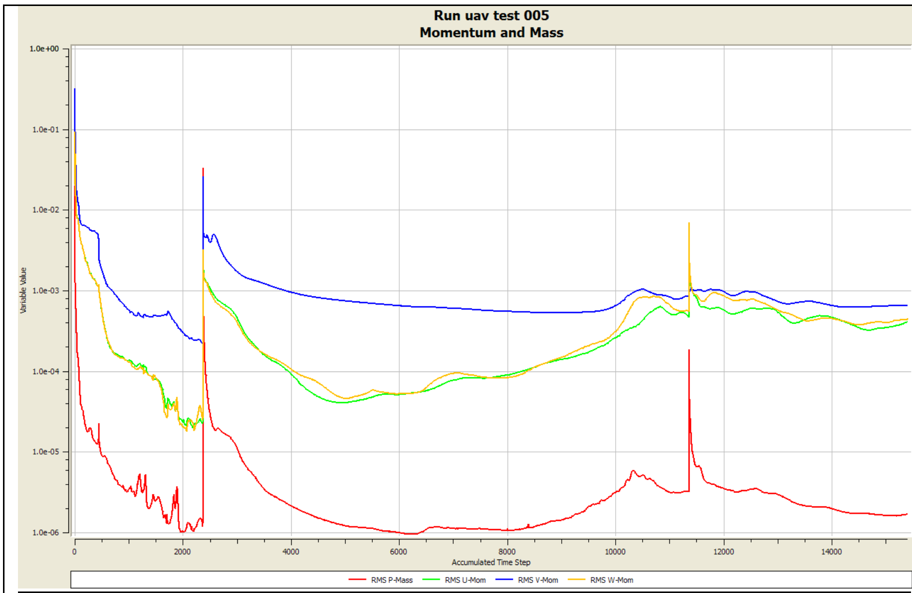

Figure 25: Root Mean Square vs. Time Step, Config. D

Figure 26 shows the RMS versus Time Step plot, in which it can be seen that the second calculation has been running for quite some time: for about 9000 iterations. A remarkable detail is the fact that the RMS values *are going up again* after some time. It looks like at Time Step = 6200 the best results are being obtained(!). *This shows that waiting for more iterations, not always leads to better results*! At a time step of about 11500, the solution was terminated and restarted again in order to obtain results with a quality comparable to those that would have been obtained at a time step of 6200. In the end the calculation was paused and the results obtained at Time Step = 1550.

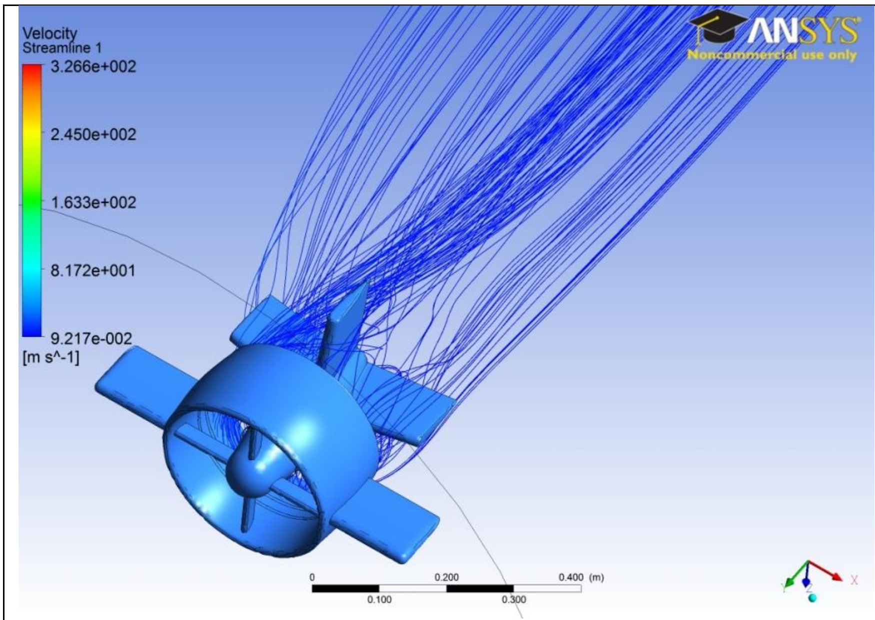

Figure 26: Streamline Pattern, Originating form the Second Propeller, Config. D

As can be seen from the streamline pattern emerging from the second propeller, the deviation in direction of the streamlines is a lot less than was the case with Configuration C. Although there is a slight deviation to the left, this is not as bad as it used to be. *From this it can be concluded that keeping the amount of revs of the first and second propeller close to each other, leads to a more concentrated bundle of streamlines in the slipstream of the plane*.

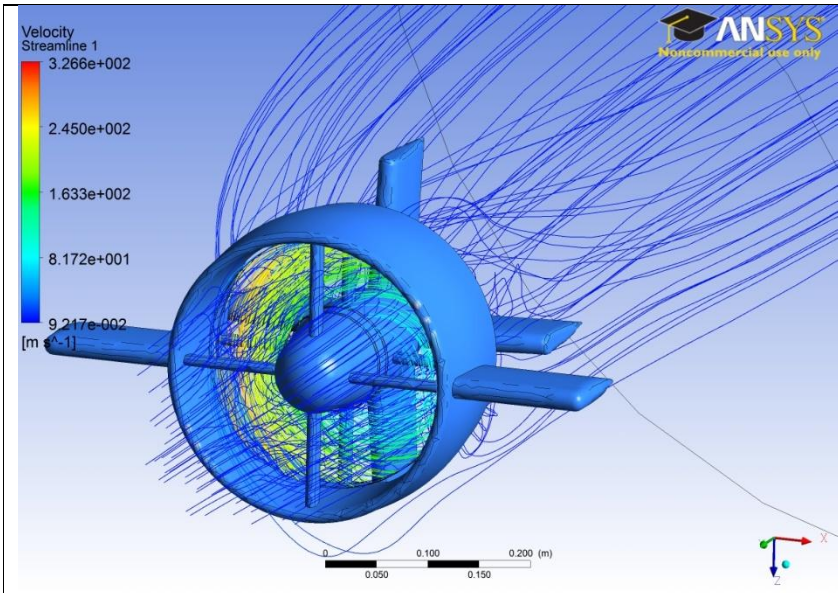

Figure 27: Streamline Pattern, Going through the First Propeller

Apart from the change in rotational speed of propeller 1, a pressure jump across the propellers was again applied. As can be seen in **Figure 28**, no streamlines are coming out at the front-side of the duct anymore, which was a problem in Configuration C. This, points out the relevance of not only modelling the rotation of the propellers, but also the pressure difference that would normally be caused by them. *Future configurations will always contain both the rotating domain property and the pressure jump*.

When the relative pressure distribution over the surface of the UAV (**Figure 29**) is being observed, it can be seen that the distribution of the pressure is almost uniform. Except from the surface located in between the two propellers on the inside of the duct and the surface of the fuselage at the location of the second propeller, all surfaces have approximately the same green colour. This means the relative pressure at these locations is approximately equal to atmospheric pressure as can be seen on the scale. The areas where the pressure is higher than atmospheric pressure are located in the proximity of the propellers. This is due of the high velocities that occur at these places.

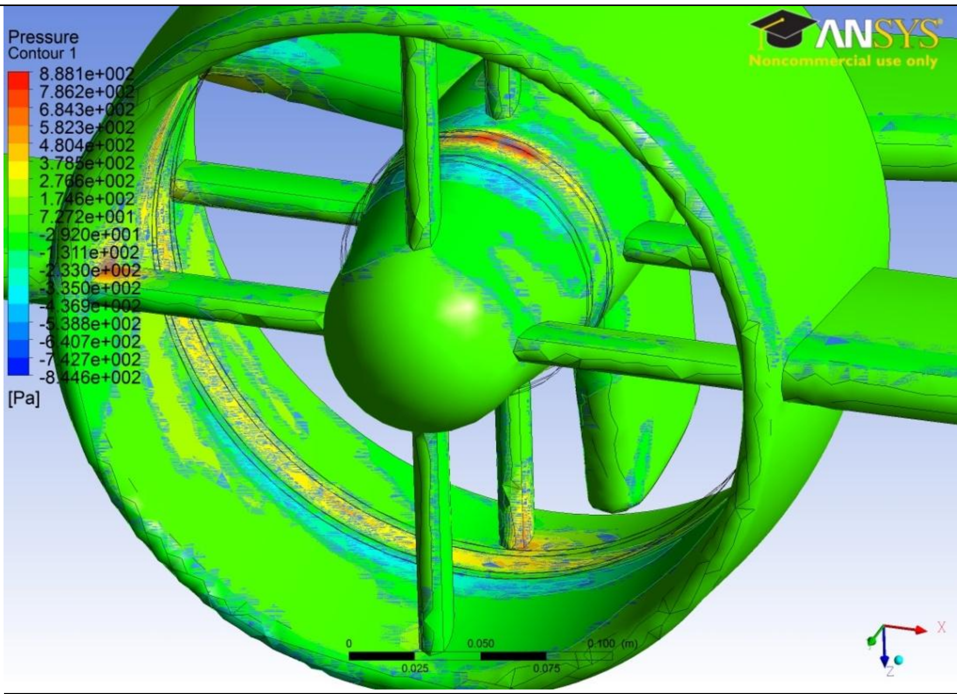

Figure 28: Relative Pressure Distribution over the UAV Surface, Config. D

Figure 30 shows the velocity vectors in and around the duct. High velocities are observed on the inside of the duct and are higher close to the inside wall. This is because the flow is accelerated towards the direction of increasing radius.

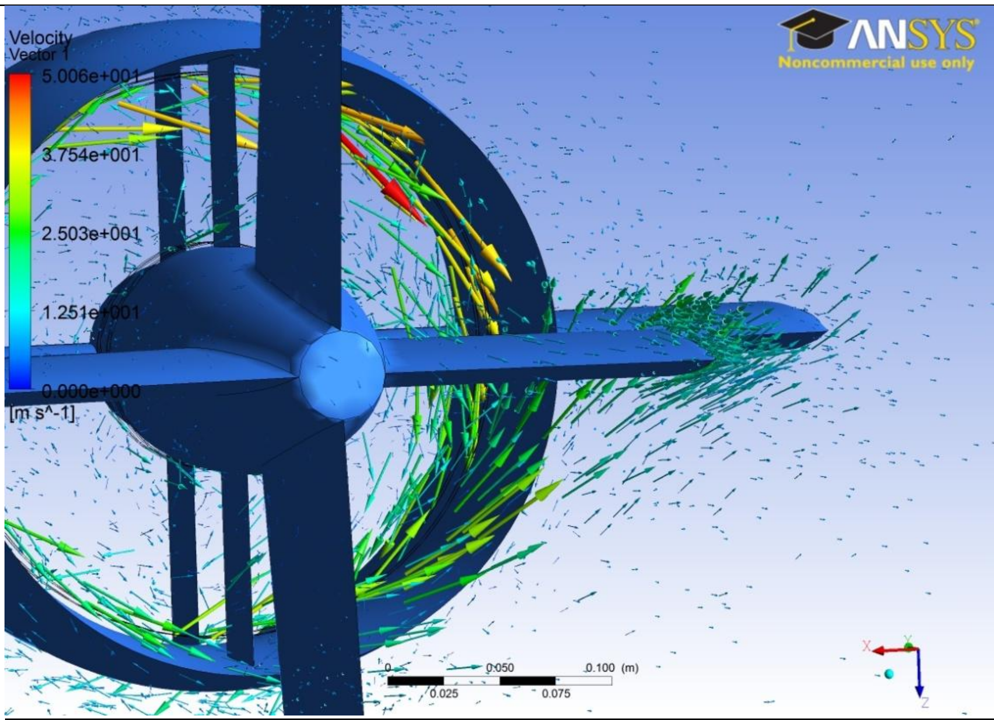

Figure 29: Velocity Vectors in the proximity of the Duct; Control Surface Tips

2.4.5. Results - Configuration E

Configuration E was used for the final simulation using the rotating flow domain concept, and is considered to be generating the *best results*. In **Table 3** it was stated that this configuration uses the SSG Reynolds turbulence model, a 0 relative pressure inlet boundary condition for the inlet surface of the control volume, counter rotation of the propellers with speeds very closely approximating the real-world values and realistic pressure jumps across the propeller domains. In addition to this, the calculation was run for a total amount of around 8000 iterations, sufficient for the values of the RMS to level out as can be seen in **Figure 31**.

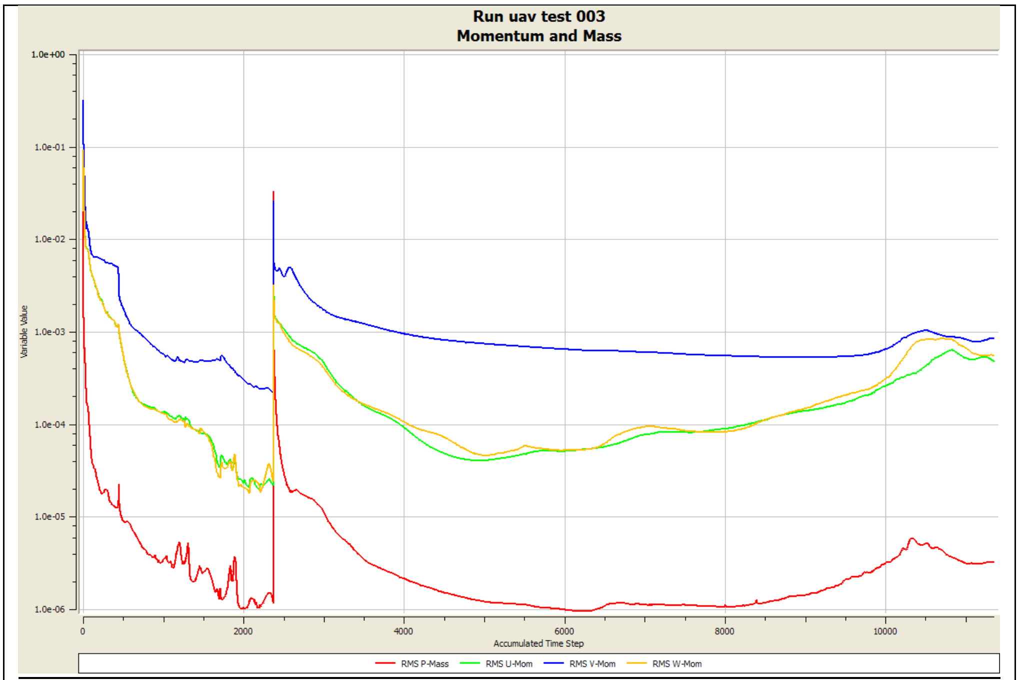

Figure 30: Root Mean Square vs. Time Step, Config. E

Figures 32 and **33** show the streamline pattern entering and leaving the duct as seen from different angles. By changing the speed of the first propeller to exactly the same value as the speed of propeller 2 (in absolute sense), the streamline pattern at the back of the UAV is now *more compact than in any of the configurations before*. This can best be observed from **Figure 33**, in which the streamline pattern is seen from above. Still the direction of the second propeller is the dominant factor in determining the direction of the outflow, but the influence has been made very small compared to Configuration C in which the deviation to the left was much larger.

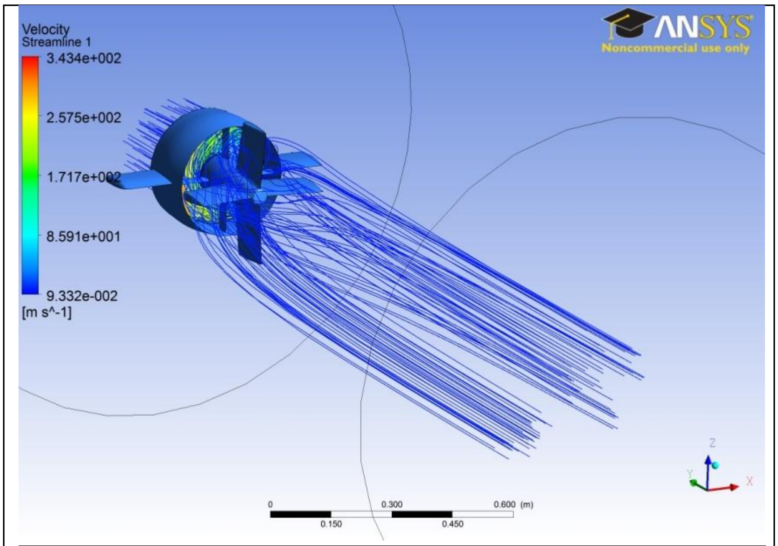

Figure 31: Streamlines entering and leaving the Duct, Config. E

This raises the question why the speed difference has been investigated anyway. Initially the speed difference was modelled in order to make sure the flow leaving the first propeller would be able to go through the second propeller without experiencing a ‘bottle-neck’, but modelling a pressure jump over the propellers at the same time will make sure that the flow goes through with no problem whatsoever. So making sure the first propeller spins just as fast as the second propeller only seems to cause an advantage, being a smoother streamline pattern over the control surfaces!

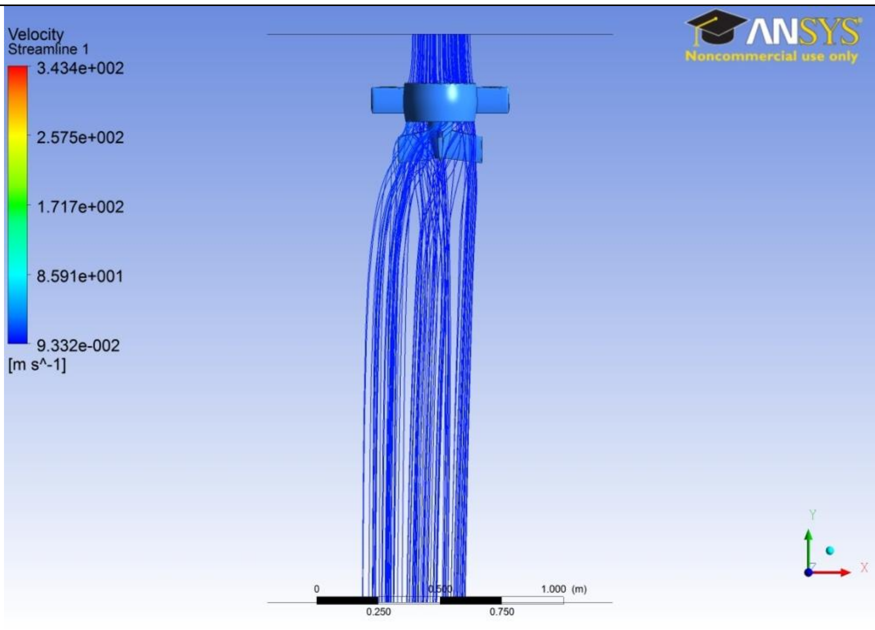

Figure 32: Streamlines entering and leaving the duct as seen from above, Config. E

A plot of the pressure distribution (**Figure 34**) shows a relative pressure on the surface of the UAV that is 0 almost everywhere, except from the location of the propellers and certain parts of the stators with which the duct is connected to the fuselage. So the pressure jump alone doesn't cause the flow to accelerate fast enough to cause any noticeable pressure differences on the surfaces of the UAV. In contrast to this, the rotating propeller domains certainly have an influence on the pressure. In real life, the work done on the airstream would cause the UAV to accelerate causing stagnation pressure on the surface. Something that is not observed using this model.

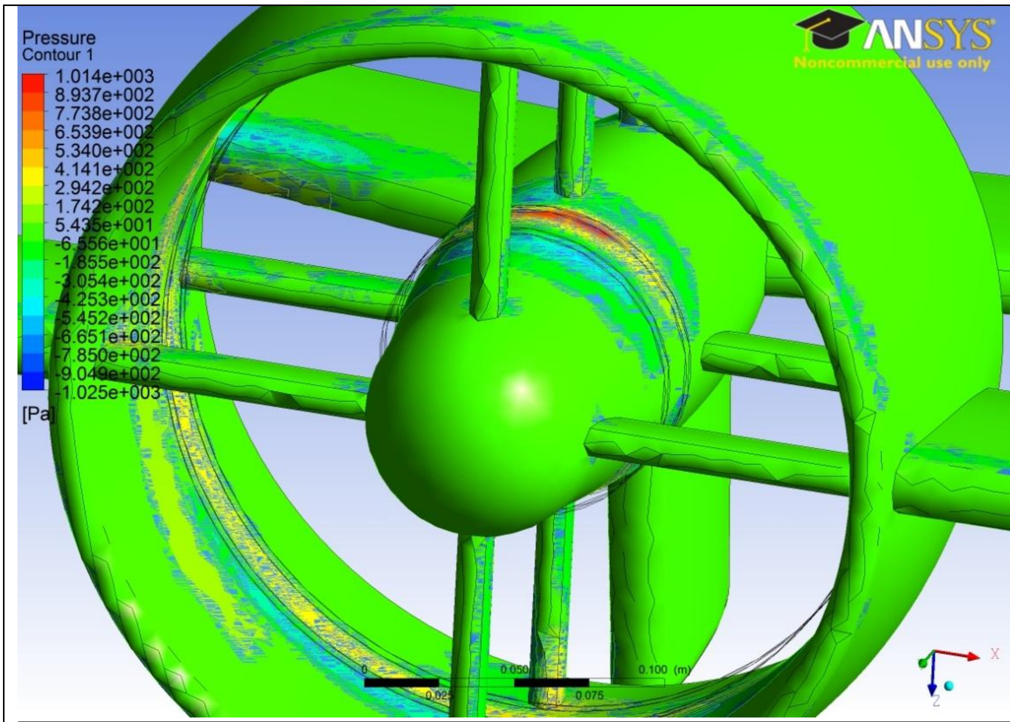

Figure 33: Relative Pressure Distribution over the UAV Surface, Config. E

The last two pictures (**Figures 34 and 35**) show the velocity vectors close to the vehicle. Around the edge of the right control surface area in **Figure 34** large velocity vectors are seen. This seems to be caused by airflow, coming from the bottom side of this surface. At the other control surface wings this is not observed, which makes the flow appear to be asymmetrical. The same has to be concluded from **Figure 35**, which clearly shows that on the left side of the duct velocities are higher. There is no obvious cause for this asymmetrical behaviour, because all the boundary conditions are symmetrical. Future research should confirm of reject this behaviour, based on more simulations and a very close review of the geometry.

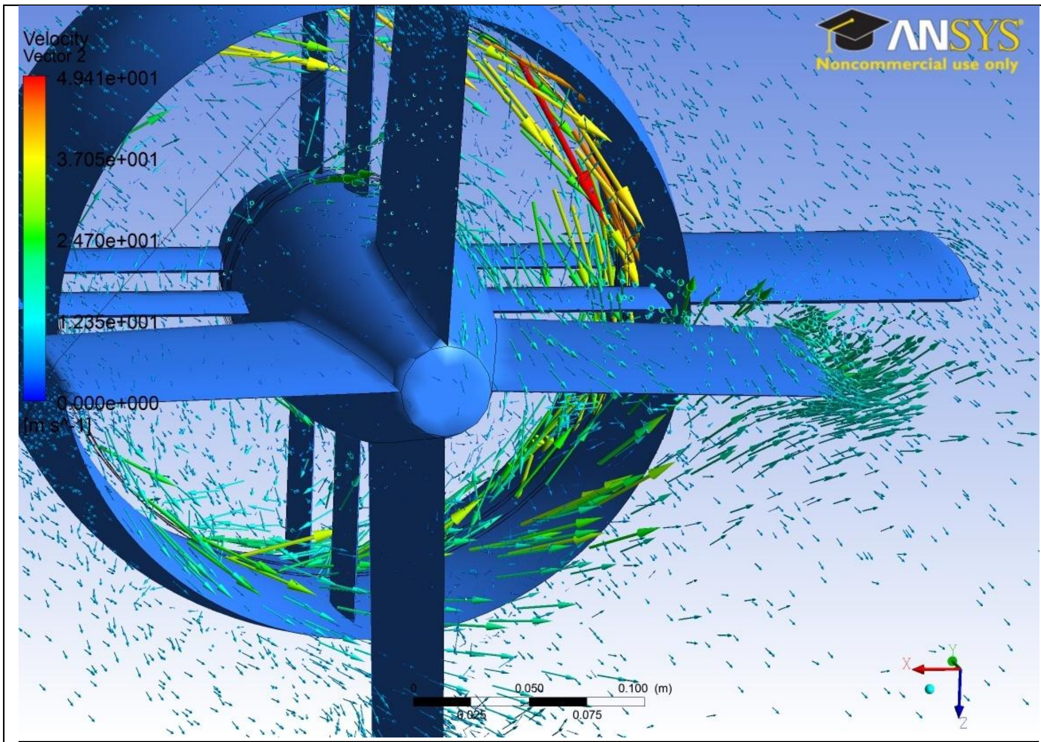

Figure 34: Velocity vectors in the proximity of the duct

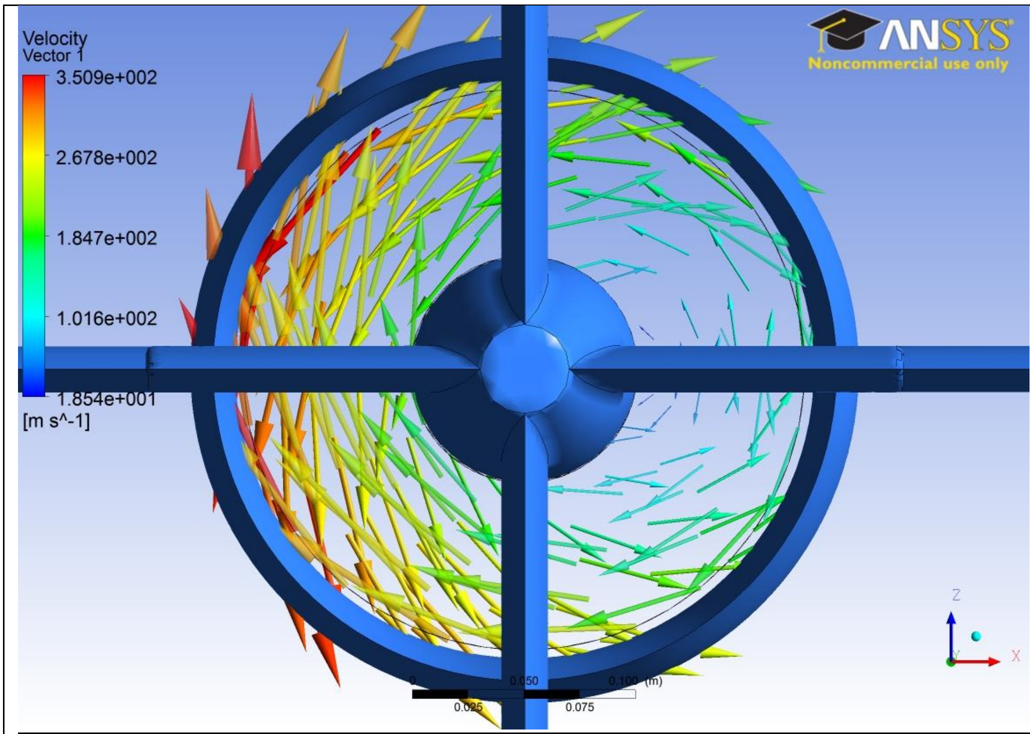

Figure 35: Velocity vectors at the location of the propellers

Chapter 3: Analysis of Case II, Replacement of Rotor by Empty Space with Velocity B.C.'s at Domain Surfaces

This chapter describes the implementation of the approach outlined in Chapter 1, for the case of modelling the two propellers by means of a velocity boundary condition on the surfaces that are created by subtracting the propeller disks from the control volume. **Figure 36** gives an overview of the configuration.

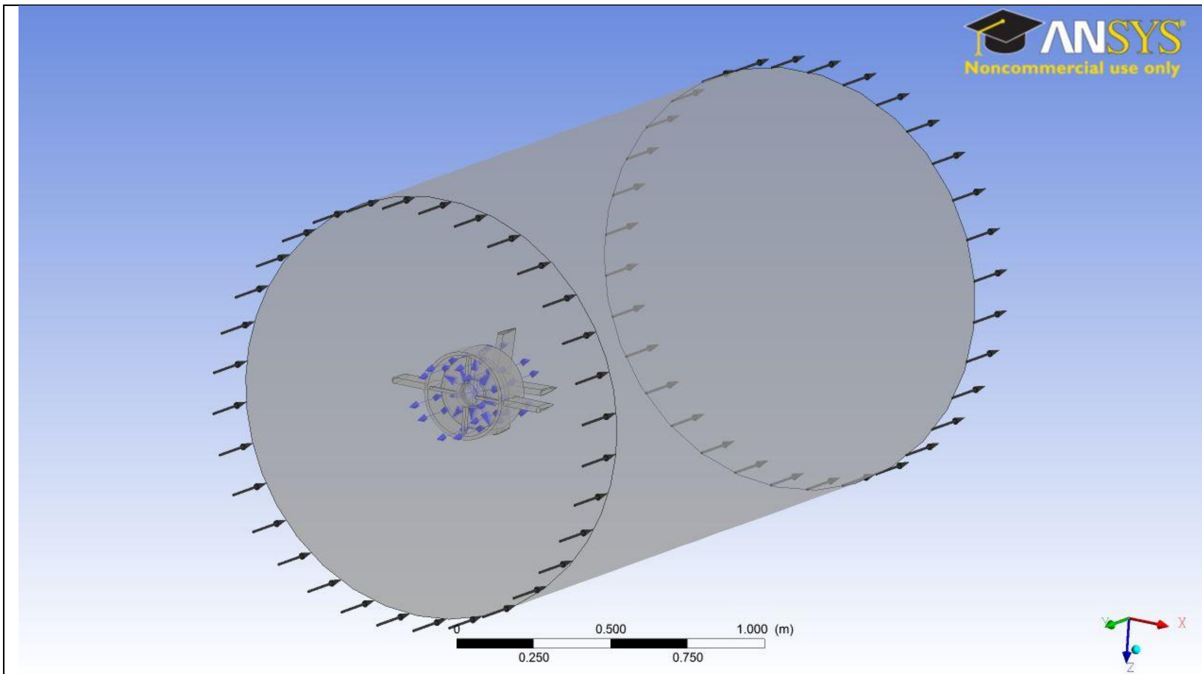

Figure 36: Domain Overview, ANSYS CFX-Pre

3.1 Creating the Geometry - Adjustments compared to Case I

Just as in the first case, the control volume consists of a cylinder from which the geometry of the UAV and two disks have been subtracted. This situation is different from Case I with respect to the way in which *boundary conditions* are applied.

In this report two approaches to this problem have been used. One of them, *Case I*, chooses to model the disks as separate fluid domains, having a rotational velocity compared to the control volume. In that case the fluid contained within the disks has the same properties as the air in the control volume, but it is rotating with the same speed as the propellers would do in the real design. Air is forced to flow from the inlet of the propeller to the outlet by a negative pressure gradient.

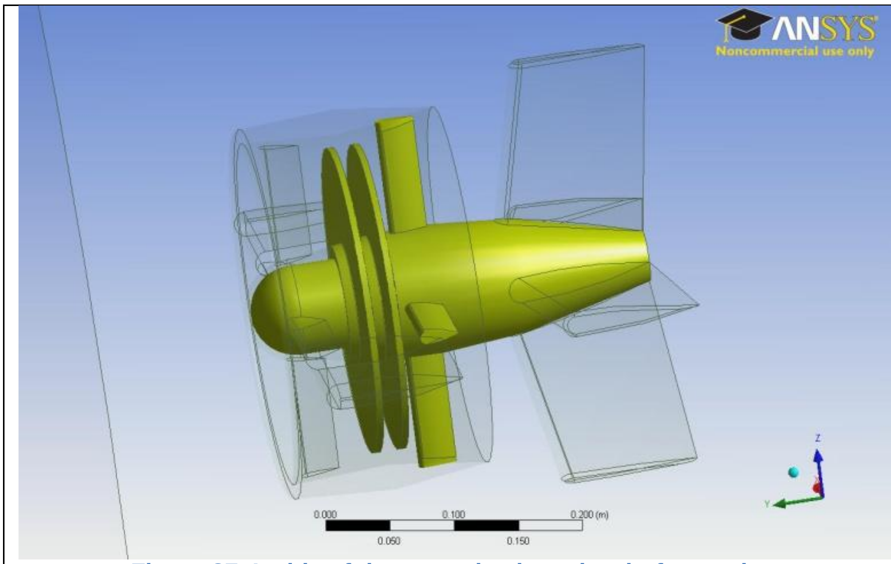

Figure 37: Inside of the control volume in wireframe view

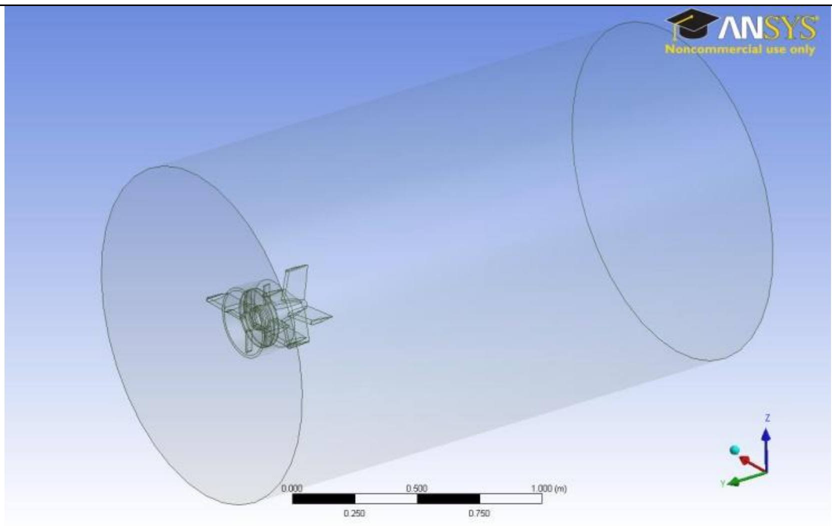

Figure 38: Control volume, overview

Then another approach was chosen in which the cut-out disks would not be replaced by a fluid domain, but by *empty space* (*Case II*). Without changing the boundary conditions the empty spaces would become an *obstacle* for the airflow, just like the UAV surface, so the B.C.'s had to be of the *surface velocity and – direction* kind. More specifically: cylindrical velocity components. By applying a velocity B.C. on the propeller inlet- and outlet sides, the flow will behave as if it would pass through the propeller domain gaining momentum from the propeller blades. When this approach is used, it is up to the user to *predict the effect of the spinning rotor blades* on the flow velocity and direction, which requires the designer of the model to have a good knowledge of propeller systems. Because I haven't got this knowledge, the model is just as good as my ability to guess what the influence of the propellers will be on the flow velocity. Also, this model doesn't include pressure differences generated by the propellers, because there are no domain interfaces. Pressure jumps can only be applied when the air moves from one domain to the next.

3.2.1. Meshing - Method

Even when automatic mesh generation is chosen there are still some options to choose from. For example the type of elements to be used:

- Automatic (let the software decide)
- Tetrahedrons
- Hex Dominant
- Sweep
- Multizone

'Tetrahedrons' has been chosen, because it is the most used element type for applications involving automatic mesh generation for complex 3D geometry.

3.2.2. Meshing - Sizing

Sizing of the elements is important in order to assign regions with larger or smaller element sizing. A small element size results in a large number of elements, whether large elements result in fewer elements. Generally regions with *high pressure or velocity gradients* need a *fine mesh* using small elements, while for regions without, a coarse mesh will do. The reason for this is to keep the overall number of elements within a reasonable amount, so the calculation time won't be too long.

In order to control the amount of elements, the surface of interest can be selected together with the dimension of a single element and a growth factor. *The UAV surface has been selected as the surface of main interest and an element size of 0.008 meters has been chosen in combination with a growth factor of 1.2*. The result of the chosen method in combination with the sizing options is shown in the **Figures 39** to **41**.

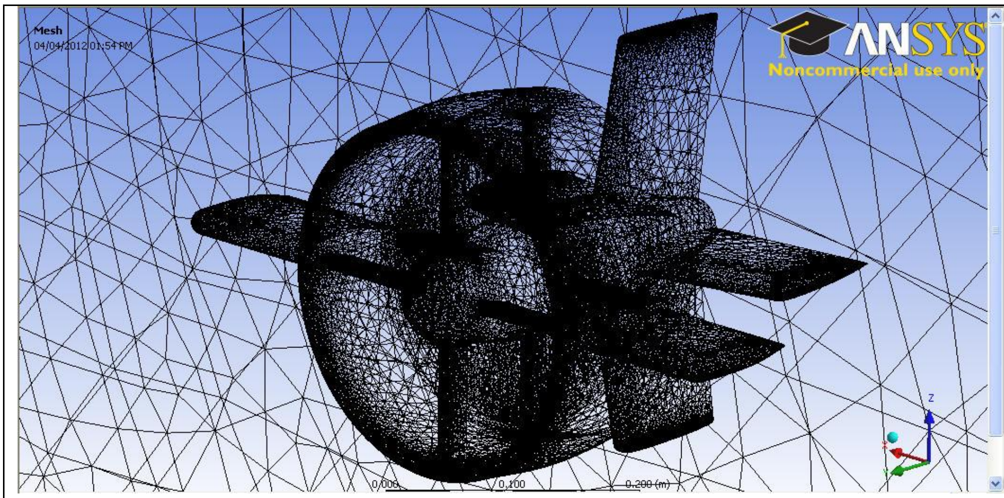

Figure 39: Volume mesh of control volume. For clarity only surface elements are visible.

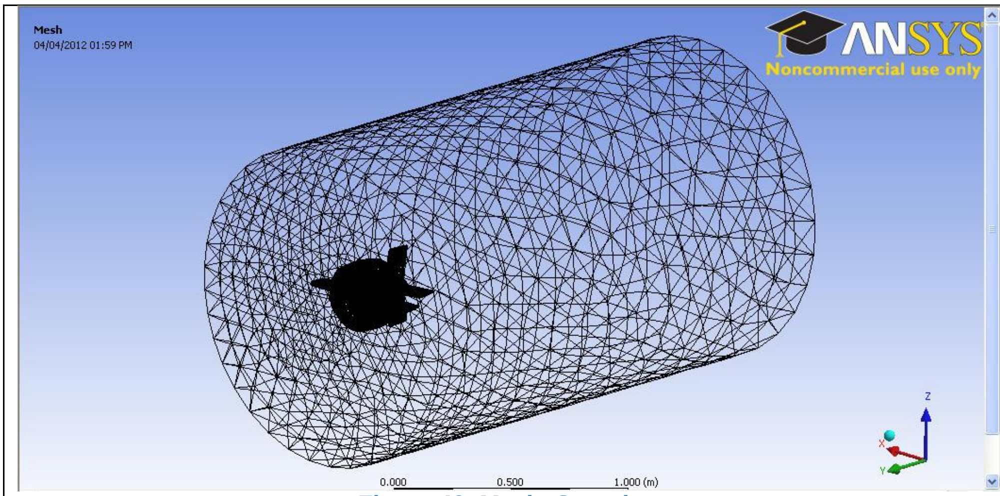

Figure 40: Mesh, Overview

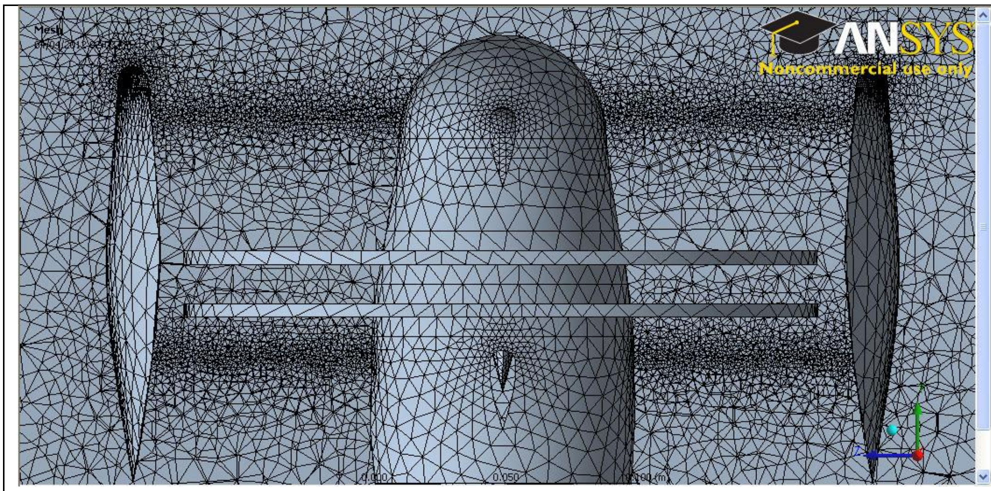

Figure 41: Element size increases from the surface of the UAV towards the boundaries of the domain

3.2.3. Meshing - Statistics

To end this section, a **Table 4** gives some *mesh statistics*. Numbers of elements way over the number reported in the table would result in memory errors, arising from the fact that meshing is performed with an educational version limited to a certain amount of elements.

The fact that ANSYS mesher returns the amount of elements contained in the model, makes it easy for the user to check whether or not the volume of choice has been meshed, in this case the control volume. When switched to wireframe mode, it looks like only a surface mesh was created (**Figure 40**), but obviously this cannot be the case when looking at the amount of elements.

Parameter	Lowercase
Minimal element size	1.5127e-003 m
Maximal element size	0.3025e 0 m
Nr. of nodes	264945
Nr. of elements	1470811

Table 4: Mesh Statistics

3.3.1. Pre-processing - Introduction

The pre-processor is used to create the problem setup. This paragraph will explain which settings were used for the calculations of Case II. It soon will become clear that most of the settings are identical to those of Case I. Whenever this is the case, references will be made to that particular paragraph in Chapter 2.

3.3.2. Pre-processing - Steady State or Transient

The same settings have been used as in Case I for reasons mentioned in paragraph 2.3.1. .

3.3.3. Pre-processing - Thermal Model

The same settings have been used as in Case I for reasons mentioned in paragraph 2.3.2. .

3.3.4. Pre-processing - Domain Specification

Compared to Case I, only one instead of three domains are present, which from now on will be called the control volume. The properties that apply to this domain are given in Attachment A under "**Case II, Replacement of Rotor by Empty Space with Velocity B.C.'s at Domain Surfaces**".

3.3.5. Pre-processing - Boundary Conditions

After specifying the domain properties, the boundary conditions have to be set. Compared to Case I, there are no domain interfaces, for there is only 1 domain: the control volume. The boundary conditions can also be found in **Attachment A**.

3.3.6. Pre-processing - Solver Control

Now that the boundary conditions have been set, the solver control settings have to be adjusted in a way they are suitable for the kind of flow model that needs to be resolved. Setting the solver control parameters has been an iterative process of running calculations and checking if the results are satisfactory or that more iteration steps are required. In case the solution had terminated even before the maximum amount of iterations had been reached, while the streamline pattern or the pressure distribution were still not satisfactory, the convergence criterion was lowered. Case II used the same solver control settings as Case I does.

3.4.1. Results - Introduction

Just like in Case I, different configurations have been used in order to get an idea of the usefulness of the model. Because this model is much simpler than the model in Case I, there aren't as many parameters to change as in the previous model. Because the SSG Reynolds model produced good results in Case I, it will be used again in this case. Because there are no domain interfaces, a pressure jump cannot be applied on the inlet and outlet surfaces of the propeller surfaces. Because of this restriction, the flow needs to be guided towards the duct entrance in some other way. That is the reason why there is a velocity inlet boundary condition at the control volume inlet. The boundary condition at the outlet of the control volume is just like in Case I a relative pressure of 0 Pa, equal to a pressure of 1 atm. This only leaves the cylindrical velocity components on the inlet and outlet of the propeller surfaces to be chosen. The axial velocity component at the in- and outlet surfaces is based on the cruise velocity of the plane, which is 20 m/s. Based on the average rotational speed of the propellers and their dimensions, the average circumferential velocity has been calculated, which appears to be 107 ms^{-1}. The value for the radial velocity component is a guess: the same as the axial velocity component. This is the configuration that was used for obtaining the results.

3.4.1. Results – Configuration A

Figure 42 shows the streamline pattern coming out of the first propeller. At the location of the first propeller, one clearly sees the effect of the velocity boundary condition. Just after the first propeller the flow is forced in the opposite circumferential direction by the equal but opposite velocity boundary condition applied at Propeller 2. **Figures 44** to **46** show the same situation from different angles. It looks like the streamlines have a preference to emerge from two of the four quarters of the duct rather than to go through the other two as can be seen from **Figure 43**. In this picture the streamline pattern looks denser on the right than it does on the left side. The same seems to be going on at the lower side, on the left (as seen from above). This causes the two densely packed bundles of streamlines to cross in the center of the outflow.

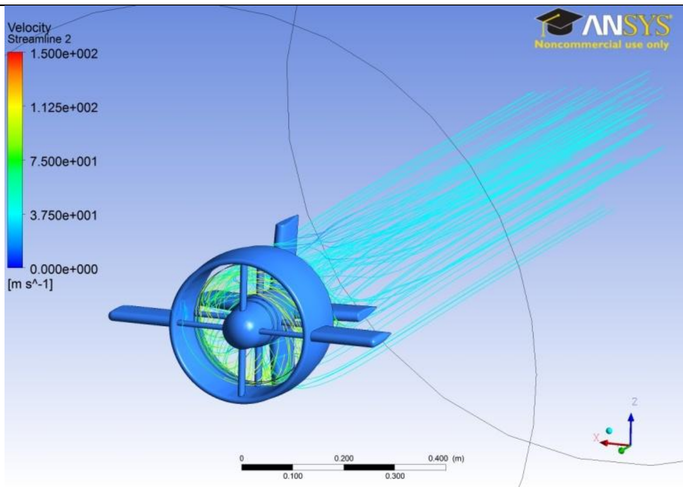

Figure 42: Isometric View of streamlines emerging from Rotor 1

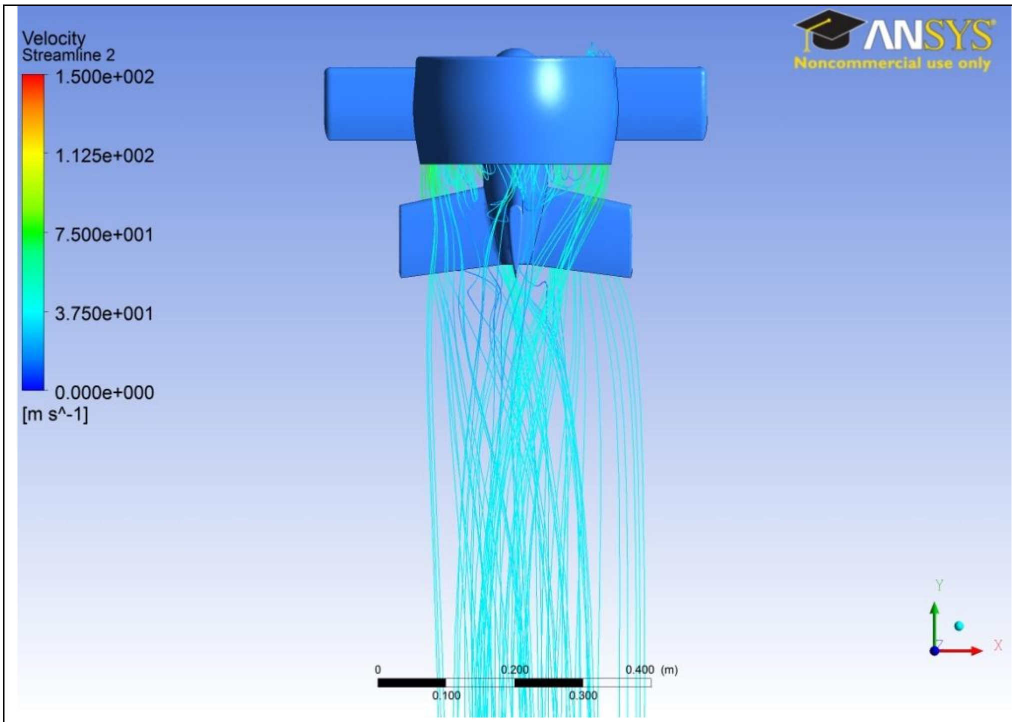

Figure 43: Top-View of the same streamline pattern as was shown in Figure 40

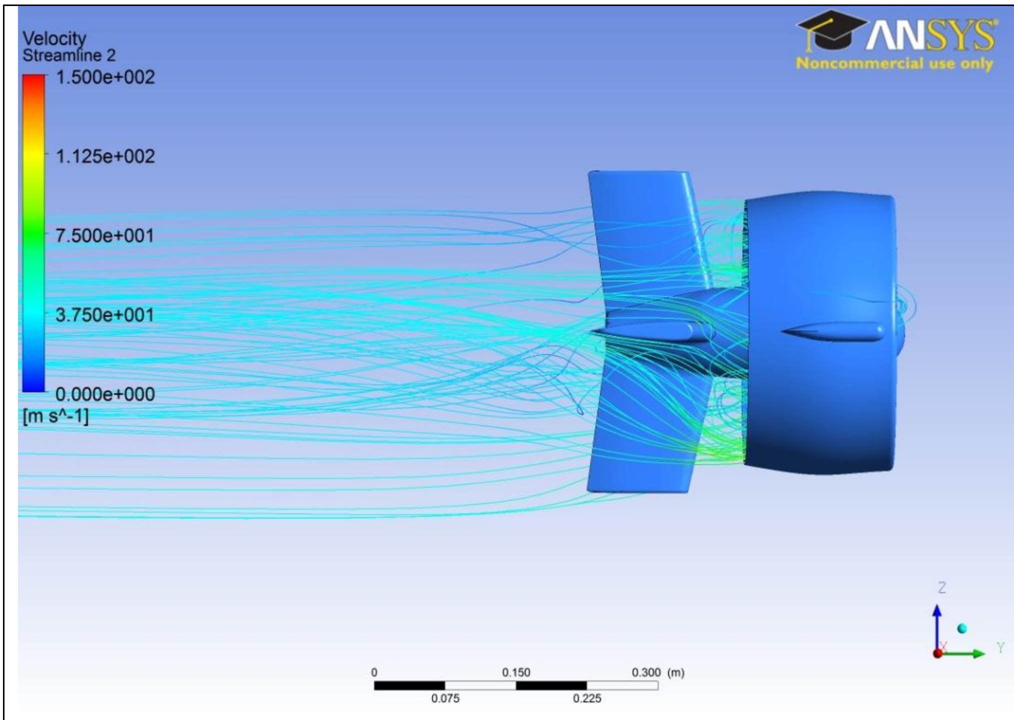

Figure 44: Streamline pattern from Figure 40 and 41 as shown from the side

Figure 45 shows that although the outflow is pretty straight, just like in Case I the direction of the second propeller is of great influence to the overall direction of the flow. It seems to be hard to compensate the effect of the first Propeller by making use of a second one.

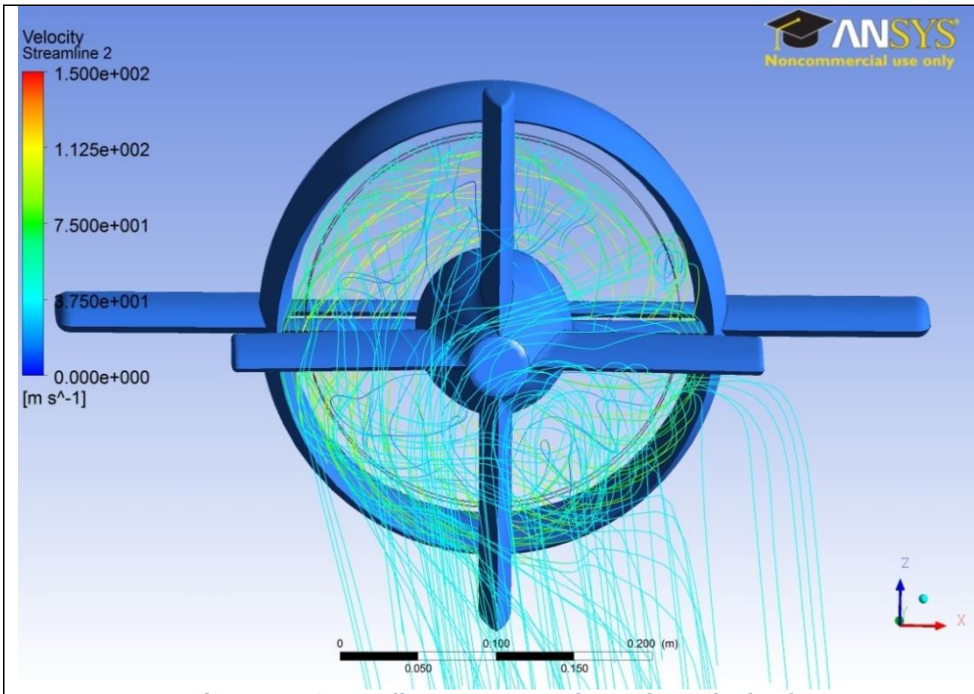

Figure 45: Streamline pattern as shown from the back

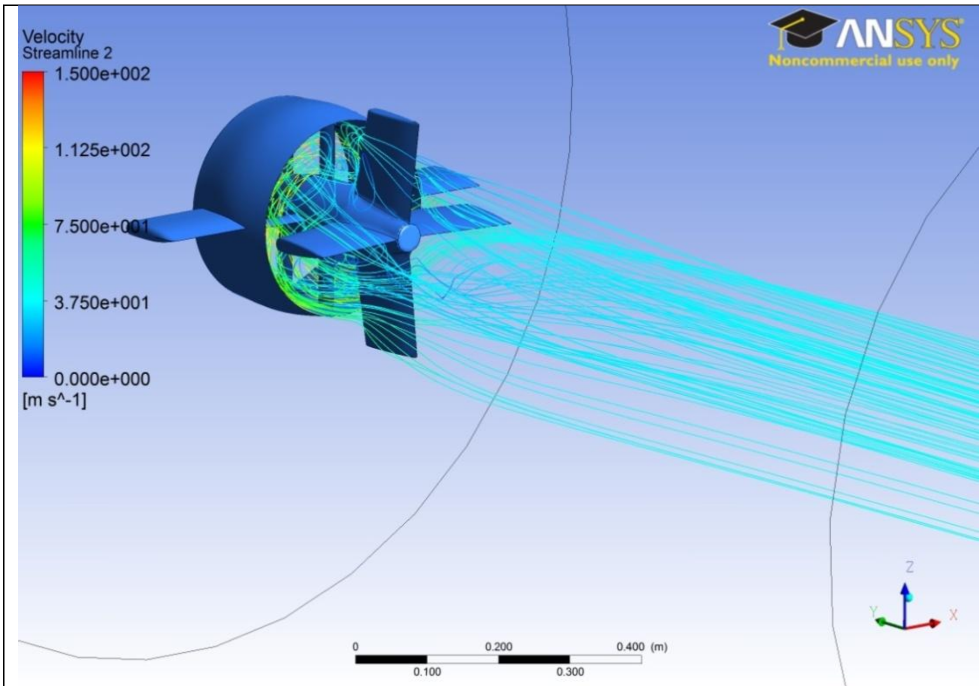

Figure 46: Streamline pattern as shown from the back, Isometric

The last picture (**Figure 47**) shows the distribution of relative pressure across the surface of the UAV. Although there is a lot of yellow colour in the picture, equal to a pressure of -1.550e002 Pa, the front facing surfaces of the wings, certain areas on the duct and parts of the control surfaces show an orange colour indicating a (higher than atmospheric) pressure of 1.375e003 Pa. This pressure raise is caused by the flow hitting those surfaces at speed. That is also the reason for the red colour on the inside of the duct and on the stators in the outflow of the propellers. On the fuselage there is also a zone with a blue colour, indicating a very low pressure. This is caused by a *weakness in the model*. When the propeller disks are subtracted from the control volume, together with the geometry of the UAV, part of the fuselage disappears when the inner diameter of the disk is the same as the diameter of the fuselage at that point. So the pressure that occurs in this area should be ignored.

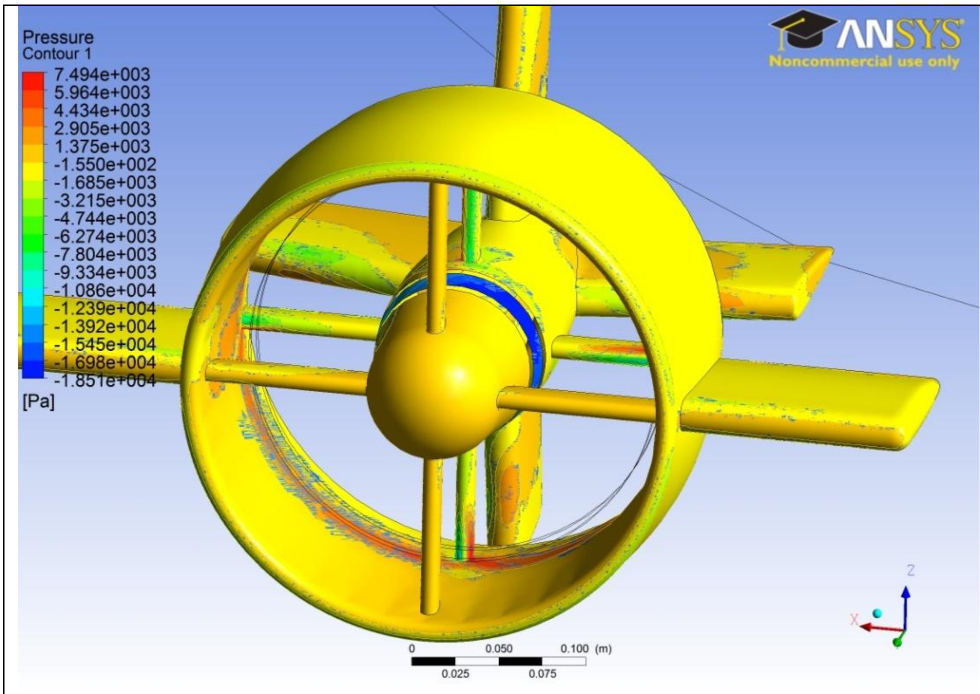

Figure 47: Relative pressure distribution at UAV surface

Chapter 5: Conclusion and Suggestions for further research

The aim of the project was to investigate what the effect would be of using counter rotating propellers for the propulsion of the UAV, developed at RMIT University. It was assumed that by using two propellers that are rotating in opposite directions, the outflow from the duct would be straighter than would be the case when a single propeller is used. This is important because the UAV is controlled by adjusting the position of the ailerons on the control surfaces, along which the outflow passes. When the flow already has an angle with respect to the control surfaces, the UAV won't be flying straight when the ailerons are in their neutral position. This is very undesirable, because airplane design should always be aiming for stable flight conditions.

Two approaches were used in modelling the UAV of which the approach described in Chapter 2 is the most realistic one, because both the pressure jump across the propellers and the counter rotation is integrated. The approach in Chapter 3 is only able to model the velocity at the location of the propellers, but cannot model the pressure jump. For this reason, the results of the first approach are supposed to be the most realistic ones.

Configuration E shows that the streamline pattern isn't symmetric, although otherwise was expected. Also the velocity distribution at the surface of the propellers isn't uniform, as can be seen from **Figure 35**.

This leaves the question whether the concept of counter rotation really doesn't work that well, or the model still contains some physical or numerical errors.

For the purpose of future research into the propulsion of the UAV this report is certainly of value, because every step that has been taken in order to build the model has been described, so the mistakes that were made while building the model won't me made twice. Another thing is that the model has been made using software that is available at the School of Aerospace and Mechanical and Manufacturing Engineering these days, in contrast with the models that had been made before and could not be edited anymore. As a result, work can be started from the models that have been made during this project.

A suggestion for improvement is related to symmetry. The numerical model created for the purpose of this internship project can be simplified by making use of symmetry. The UAV and the control volume surrounding it can be split into two identical parts, saving computer memory and for this reason computation time. When this is done, one should keep in mind that the rotating parts are not symmetrical, so a solution has to be found for this problem first. When the model can be simplified, this leads to quicker analysis of the results generated, after the model parameters have been changed. A lot of time was spent waiting for solutions to converge, while after the model had converged it seemed that the streamline pattern didn't make any sense at all. This makes waiting for those solutions a waste of time, except for the fact that it gave some information about what did not work.

Another advantage of using symmetry is that the streamline pattern is more likely to be symmetrical. The reason for this is that the asymmetry of the streamline pattern might be caused by the model geometry, although this is hard to see with the naked eye. When one half of the UAV is an exact copy of the other half of the UAV, this cause for the asymmetric flow pattern can be ruled out.

References

- LEAP Australia, Melbourne-based Company, specialised in ANSYS. LEAP Australia delivers its services to companies working with ANSYS and to students for educational purposes.
- www.cfd-online.com, forum for technical professionals and students working with CFD-software
- Ducted-fan UAV.pdf, author unknown
- Lampe A., Aerodynamic design of a VTOL UAV, *Internship on behalf of Masters at RMIT*, *Melbourne*, January 2008
- Zhao H., Eng B., Development of a Dynamic Model of Ducted Fan VTOL UAV, 2008
- Chatrenet N., Design of a Mini Tail-Sitter Ducted Fan VTOL UAV, December 2005
- ANSYS software, product and program documentation

Attachment A: Domains, Boundaries and Domain Interfaces

Case I: Replacement Propeller by Rotating Fluid Domain

Control volume, Domain Properties:

Domain Type:	Fluid Domain
Material:	Air at 25 ^oC (298 K)
Morphology[1]:	Continuous Fluid
Reference Pressure:	1 atm
Buoyancy[2]	Non Buoyant
Domain Motion:	Static
Mesh Deformation	None
Heat Transfer:	Isothermal
Turbulence Model:	SSG Reynolds Stress

1.) Morphology is the property that describes the connectivity of the medium. In case water droplets in an air stream are to be described, 'dispersed fluid' would be a good choice. In this case we are dealing with a continuous medium, which is air at 20 ^oC.
2.) Buoyancy plays a role when *differences in density* occur in the medium as a result *temperature variation*. This is called natural convection. Also in case of *multicomponent flow*, buoyancy can be included because differences in material density may cause convection. In this model *velocities won't exceed Mach 0.3* because velocities that high are unlikely to occur for propeller-driven aerial vehicles. For this reason the air can be considered *incompressible*. Also the influence of temperature differences can be neglected because *isothermal* properties are being assumed.
3.) There are no significant sources of heat within the control volume. In real life, the engine driving the propellers would be causing some inflow of heat in the domain, but even then the high amount of air flowing past the surfaces of the UAV will not allow for heat build-up. As a result the air temperature will be constant.

4.) Turbulence

Turbulence plays an important role in generating a CFD-model for a propeller-driven aerial vehicle, because of the large *disturbances* of the flow created by the propeller. Turbulence consists of *fluctuations in the flow field in time and space* and is a complex 3 dimensional process that is both *unsteady* and consists of many scales. Turbulence occurs when the inertia forces in the fluid become more significant compared to viscous forces, and is characterized by a *high Reynolds number*.

In theory turbulence can be described by the *full Navier-Stokes equations*, but doing that would require a mesh so fine that no computer would be able to solve them. Not now, and not in the nearby future. That's the reason why most methods in CFD make use of *turbulence models*. A right choice for the turbulence model is very important in pre-processing the case at hand.

Most of the turbulence models are *statistical models*. When looking at time scales much larger than the time scales of turbulent fluctuations, turbulent flow could be said to exhibit *average characteristics*, with an additional *time-varying*, fluctuating component. In general, turbulence models seek to modify the original Navier-Stokes equations by the introduction of averaged and fluctuating quantities to produce the *Reynolds Averaged Navier-Stokes (RANS) equations*. These equations represent the mean flow quantities only, while modelling turbulence effects without a need for the resolution of the turbulent fluctuations. Turbulence models based on the RANS equations are known as Statistical Turbulence Models due to the statistical averaging procedure employed to obtain the equations. Simulation of the RANS equations greatly reduces the computational effort compared to Direct Numerical Simulation and is generally adopted for practical engineering calculations.

However, the averaging procedure introduces additional unknown terms containing products of the fluctuating quantities, which act like additional stresses in the fluid. These terms, called *'turbulent'* or *'Reynolds' stresses*, are difficult to determine directly and so become further unknowns. The Reynolds (turbulent) stresses need to be modelled by additional equations of known quantities in order to achieve "closure". The equations used to close the system define the *type of turbulence model*. Turbulence models close the Reynolds averaged equations by providing models for the computation of the Reynolds stresses and Reynolds fluxes.

CFX models can be broadly divided into two classes: eddy viscosity models and *Reynolds stress models*, of which the last type of model will be used in case of *rotational flow conditions*.

In flows where the turbulent transport or non-equilibrium effects are important, the eddy-viscosity assumption is no longer valid and results of eddy-viscosity models might be inaccurate. Reynolds Stress models naturally include the effects of streamline curvature, sudden changes in the strain rate, secondary flows or buoyancy compared to turbulence models using the eddy-viscosity approximation. Among the types of flow that can be modelled by the Reynolds Stress model are 'Free shear flows with strong anisotropy, like a strong swirl component, like flows in rotating fluids.' and 'Flows with strong streamline curvature.', which is obviously the case.

Reynolds Stress models have shown superior predictive performance compared to eddy-viscosity models in these cases. This is the major justification for the Reynolds Stress models, which are based in transport equations for the individual components of the Reynolds stress tensor and the dissipation rate. These models are characterized by a higher degree of universality. The penalty for this flexibility is a *high degree of complexity in the resulting mathematical system*. The increased number of transport equations leads to *reduced numerical robustness*, requires *increased computational effort* and often prevents their usage in complex flows.

Three varieties of the Reynolds Stress Model are available which use different model constants, but in practice one of those proves to be more accurate, which is the *SSG Reynolds Stress Model*. Especially for *swirling flows* this model produces superior results. Compared to the k-epsilon model, the Reynolds Stresses model has six additional transport equations that are solved for each time step or outer coefficient loop in the flow solver. The source terms in the Reynolds Stress equations are also more complex than those of the k-epsilon model. As a result of these factors, outer loop convergence may be slower for the Reynolds Stress model than for the k-epsilon model.

In principle the same time step can be used for all turbulence model variants, but pragmatically the *time step should be reduced* for the Reynolds Stress model due to the increased complexity of its equations and due to numerical approximations made at general grid interfaces and rotational periodic boundary conditions.

So in brief: the Reynolds Stress Models has been chosen as the preferred turbulence model, or more specifically the SSG RSM. Doing so makes a demand for decreased time steps compared to the more commonly used k-epsilon model.

[*]) A great part of the explanation about different turbulence models has been subtracted from the document CFX Theory and CFX Modeling Guide, provided together with the ANSYS software, to make the user familiar with the theory on which the CFX software is based.

<u>Control volume, Boundary Conditions:</u>

Propeller1_Inlet *Side 1 2* Type: Interface Location: Propeller1_Inlet_CV	***Propeller2_Inlet*** *Side 1 1* Type: Interface Location: Propeller2_Inlet_CV

Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux	Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller1_Shaft *Side 1* Type: Interface Location: Propeller1_Shaft_CV Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux	***Propeller2_Shaft*** *Side 1* Type: Interface Location: Propeller2_Shaft_CV Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller1_Outlet *Side 1 1* Type: Interface Location: Propeller1_Outlet_CV Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux	***Propeller2_Outlet*** *Side 1 1* Type: Interface Location: Propeller2_Outlet_CV Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller1_Wall *Side 1* Type: Interface Location: Propeller1_Wall_CV Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux	***Propeller2_Wall*** *Side 1* Type: Interface Location: Propeller2_Wall_CV Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Inlet Boundary Type: Opening Location: Inlet Flow Regime: Subsonic Mass and Momentum: Opening Pressure; Relative Pressure = 0 Pa Turbulence: Zero Gradient	***Outlet*** Boundary Type: Opening Location: Outlet Flow Regime: Subsonic Mass and Momentum: Opening Pressure; Relative Pressure = 0 Pa; Turbulence: Medium (Intensity = 5%)
UAV_Surface Boundary Type: Wall Location: UAV_Surface Mass and Momentum: No Slip Wall Wall Roughness: Smooth Wall Heat Transfer: Adiabatic	***Wall*** Boundary Type: Wall Location: Wall Boundary Details: Free Slip Wall Heat Transfer: Adiabatic

Propeller1, Domain Properties:

Domain Type:	Fluid Domain
Material:	Air at 25 ^{o}C
Morphology:	Continuous Fluid
Reference Pressure:	1 atm
Domain Motion:	Rotating, **17000 rpm**, axis of rotation: global Y
Heat Transfer:	Isothermal
Turbulence Model:	SSG Reynolds Stress

Propeller1, Boundary Conditions:

Propeller1_Inlet *Side 1 1* Boundary Type: Interface Location: Propeller1_Inlet Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller1_Shaft *Side 2* Boundary Type: Interface Location: Propeller1_Shaft Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller1_Outlet *Side 1* Boundary Type: Interface Location: Propeller1_Outlet Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller1_Wall *Side 2* Boundary Type: Interface Location: Propeller1_Wall Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux

Propeller2, Domain Properties:

Domain Type:	Fluid Domain
Material:	Air at 25 ^{o}C
Morphology:	Continuous Fluid
Reference Pressure:	1 atm
Domain Motion:	Rotating, **-17000 rpm**, axis of rotation: global Y
Heat Transfer:	Isothermal
Turbulence Model:	SSG Reynolds Stress

Propeller2, Boundary Conditions:

Propeller2_Inlet *Side 1* Boundary Type: Interface Location: Propeller2_Inlet Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller2_Shaft *Side 2* Boundary Type: Interface Location: Propeller2_Shaft Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux

Heat Transfer: Conservative Interface Flux
Propeller2_Outlet *Side 1* Boundary Type: Interface Location: Propeller2_Outlet Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux
Propeller2_Wall *Side 2* Boundary Type: Interface Location: Propeller2_Wall Mass and Momentum: Conservative Interface Flux Turbulence: Conservative Interface Flux Heat Transfer: Conservative Interface Flux

Domain Interfaces:

Propeller1_Inlet Interface Type: Fluid-Fluid Domain: Control Volume-Propeller1 Boundaries: Propeller1_Inlet_CV; Propeller1_Inlet Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Pressure Change; Pressure Change: -150 Pa	***Propeller2_Inlet*** Interface Type: Fluid-Fluid Domain: Control Volume-Propeller2 Boundaries: Propeller2_Inlet_CV; Propeller2_Inlet Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Pressure Change; Pressure Change: -150 Pa
Propeller1_Shaft Interface Type: Fluid-Fluid Domain: Control Volume-Propeller1 Boundaries: Propeller1_Shaft_CV; Propeller1_Shaft Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Mass Flow Rate = 0 kgs^{-1}	***Propeller2_Shaft*** Interface Type: Fluid-Fluid Domain: Control Volume-Propeller2 Boundaries: Propeller2_Shaft_CV; Propeller2_Shaft Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Mass Flow Rate = 0 kgs^{-1}
Propeller1_Outlet Interface Type: Fluid-Fluid Domain: Control Volume-Propeller1 Boundaries: Propeller1_Outlet_CV; Propeller1_Outlet Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Pressure Change; Pressure	***Propeller2_Outlet*** Interface Type: Fluid-Fluid Domain: Control Volume-Propeller2 Boundaries: Propeller2_Outlet_CV; Propeller2_Outlet Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Pressure Change; Pressure

Change: -150 Pa	Change: -150 Pa
Propeller1_Wall Interface Type: Fluid-Fluid Domain: Control Volume-Propeller1 Boundaries: Propeller1_Wall_CV; Propeller1_Wall Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Mass Flow Rate = 0 kgs^{-1}	***Propeller2_Wall*** Interface Type: Fluid-Fluid Domain: Control Volume-Propeller2 Boundaries: Propeller2_Wall_CV; Propeller2_Wall Interface Model: General Connection Frame Change/Mixing Model: Frozen Rotor Pitch Change: None Mesh Connection Method: GGI Mass and Momentum: Conservative Interface Flux Interface Model: Mass Flow Rate = 0 kgs^{-1}

Case II, Replacement of Rotor by Empty Space with Velocity B.C.'s at Domain Surfaces

Control volume, Domain Properties:

Property	**Value/Description**
Domain Type	Fluid Domain
Material	Air at 25 ^oC
Morphology[1]	Continuous Fluid
Reference pressure	1 atm
Buoyancy[2]	Non Buoyant
Domain	Stationary
Mesh Deformation	None
Fluid Model[3]	Isothermal
Turbulence Model[4]	SSG RMS

Control volume, Boundary Conditions:

Inlet Type: Inlet Location: Inlet Flow Regime: Subsonic Mass and Momentum: Relative Pressure = 0 Pa Turbulence: Zero Gradient	***Outlet*** Type: Opening Location: Outlet Flow Regime: Subsonic Mass and Momentum: Opening Pressure, Relative Pressure: 0 Pa Turbulence: Medium (Intensity = 5%)
UAV_Surface Type: Wall Location: UAV_surface Mass and Momentum: No Slip Wall Wall Roughness: Smooth Wall	***Wall*** Type: Wall Location: Wall Mass and Momentum: Free Slip Wall
Propeller1_Inlet Type: Opening Location: Propeller1_Inlet Flow Regime: Subsonic Mass and Momentum: Cylindrical Velocity	***Propeller2_Inlet*** Type: Opening Location: Propeller2_Inlet Flow Regime: Subsonic Mass and Momentum: Cylindrical Velocity

Components Axial: -20 ms^{-1} Radial: 20 ms^{-1} Theta: 107 ms^{-1} Axis Definition: Rotation Axis: Global Y Turbulence: Medium (Intensity = 5%)	Components Axial: -20 ms^{-1} Radial: 20 ms^{-1} Theta: -107 ms^{-1} Axis Definition: Rotation Axis: Global Y Turbulence: Medium (Intensity = 5%)
Propeller1_Outlet Type: Opening Location: Propeller1_Outlet Flow Regime: Subsonic Mass and Momentum: Cylindrical Velocity Components Axial: -20 ms^{-1} Radial: 20 ms^{-1} Theta: 107 ms^{-1} Axis Definition: Rotation Axis: Global Y Turbulence: Medium (Intensity = 5%)	***Propeller2_Outlet*** Type: Opening Location: Propeller2_Outlet Flow Regime: Subsonic Mass and Momentum: Cylindrical Velocity Components Axial: -20 ms^{-1} Radial: 20 ms^{-1} Theta: -107 ms^{-1} Axis Definition: Rotation Axis: Global Y Turbulence: Medium (Intensity = 5%)
Propeller1_Wall Type: Opening Location: Propeller1_Wall Flow Regime: Subsonic Mass and Momentum: Cylindrical Velocity Components Axial: -20 ms^{-1} Radial: 20 ms^{-1} Theta: 107 ms^{-1} Axis Definition: Rotation Axis: Global Y Turbulence: Medium (Intensity = 5%)	***Propeller2_Wall*** Type: Opening Location: Propeller2_Wall Flow Regime: Subsonic Mass and Momentum: Cylindrical Velocity Components Axial: -20 ms^{-1} Radial: 20 ms^{-1} Theta: -107 ms^{-1} Axis Definition: Rotation Axis: Global Y Turbulence: Medium (Intensity = 5%)

Attachment B: Project Planning

Week	Date	Activity
01	06 Feb – 12 Feb	Meet with supervisors, arrange a workplace, get access to facilities. Reading Literature, Problem Specification
02	13 Feb – 19 Feb	Adapting SolidWorks-model and importing it to ANSYS. During this stage of the project a lot of support was obtained from online forums, like www.cfd-online.com, and tutorials, widely available on the web.
03	20 Feb – 26 Feb	Working out ICEM-tutorials (GAMBIT not available), applying knowledge about meshing to the UAV-case
04	27 Feb – 04 Mrch	Switch from using ICEM as meshing program to ANSYS Mesher; Making adjustments to the geometry in order to get the mesh right
05	05 Mrch – 11 Mrch	Reading CFX-Pre manual in order to get familiar with the different options and to get to know which options should be used for the UAV-case
06	02 Mrch – 18 Mrch	First runs with ANSYS CFX. Propellers are not included yet, on behalf of simplicity. Stepwise adjustments towards a more advanced model will point out modeling errors.
07	19 Mrch – 25 Mrch	After having achieved some results with the simple model, propellers are added in the form of rotating flow domains. Small changes in model geometry can cause the program to produce errors during meshing. Detailed examination of the problematic areas is required to get a clue of what is causing the error.
08	26 Mrch – 01 Aprl	Results with the rotating propeller domain have been achieved. Close attention is being paid to all the different settings in the pre-processor, like model settings and boundary conditions. Runs are being performed with more iterations than was the case before.
09	02 Aprl – 08 Aprl	On advice of a student of the SAMME, a different approach was used in which the rotating domain was changed for a velocity boundary condition on the propeller surfaces. Results were obtained and compared with earlier results.
10	09 Aprl – 15 Aprl	More runs were performed using the model with rotating domains, some of which overnight due to long waiting times. Values of parameters are changed in between runs in order to better understand the way the model works. Results are being analyzed and documented. A start has been made with the report, in which the modeling method has been described extensively.
11	16 Aprl – 22 Aprl	Running calculations, analyzing results, report writing
12	23 Aprl – 29 Aprl	Making final adjustments to the model, drawing conclusions, making final adjustments to the report, final meeting with supervisors.