TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Department of mechanical and industrial engineering

# DESIGN AND AERODYNAMIC OPTIMIZATION OF A VERTICAL TAKE-OFF AND LANDING FIXED-WING DRONE WITH BRAIN-COMPUTER INTERFACE

AJU-ARVUTI LIIDESEGA FIKSEERITUD TIIBADEGA ÕHUSÕIDUKI PROJEKTEERIMINE JA KERE AERODÜNAAMILINE OPTIMEERIMINE

MASTER THESIS

Student: Martin Simon
Student code: 153654 MATM
Supervisors: Martin Eerme (PhD)
Sada (PhD)

TALLINN, 2018

**AUTHOR'S DECLARATION**

Hereby I declare, that I have written this thesis independently.
No academic degree has been applied for based on this material.
All works, major viewpoints and data of the other authors used in this thesis have been referenced.

TUT Department of mechanical and industrial engineering

**THESIS TASK**

| | |
|---|---|
| **Student:** | Martin Simon, 153654 MATM |
| Study programme: | |
| Main specialty: | |
| Supervisors: | Martin Eerme (PhD) |
| | Sada (PhD) |
| Consultants: | |

**Thesis topic:**

| | |
|---|---|
| (in English) | DESIGN AND AERODYNAMIC OPTIMIZATION OF A VERTICAL TAKE-OFF AND LANDING FIXED-WING DRONE WITH BRAIN-COMPUTER INTERFACE |
| (in Estonian) | AJU-ARVUTI LIIDESEGA FIKSEERITUD TIIBADEGA ÕHUSÕIDUKI PROJEKTEERIMINE JA KERE AERODÜNAAMILINE OPTIMEERIMINE |

**Thesis main objectives:**

1. Design a vertical take-off and landing fixed-wing drone.
2. Optimize the shape of the drone.
3. Build a brain-computer interface on the onboard computer.

**Thesis tasks and time schedule:**

| No | Task description | Deadline |
|---|---|---|
| 1. | Research and decide on general design of the drone. | 02.02.2017 |
| 2. | Design and choose drone propulsion system. | 02.04.2017 |
| 3. | Research STate-of-the-Art optimization methods. | 02.05.2017 |
| 4. | Train models and record data for a motor brain-computer interface. | 02.06.2017 |
| 5. | Aerodynamically optimize the drone. Methods of manufacturing. | 02.12.2017 |

**Additional data and requirements:**

**Language:** English

**Student:** Martin Simon

**Supervisor:** Martin Eerme

# Contents

# List of Figures

# List of Tables

*To my supportive family.*

# 1 Introduction

## 1.1 Motivation

The primary motivation for this thesis comes from a growing need on the market of drones.

The problem: long range, stable and hover-capable drone able to carry a payload.

The solution: tricopter tilt-rotor wing body aircraft.

The secondary of the possibility to make drone pilots more effective.

During the thesis a unique tricopter tiltrotor wing body design was developed, parametrized and aerodynamically optimized. The propulsion system was verified by empirical tests and aerodynamic analysis by benchmarking with known validation cases. The following engineering tasks were solved:

- Propulsion system
- Power system
- Structural and geometric composition
- Tilting Mechanisms
- Aerodynamic optimization

A novel combination was set up utilizing a state of the art acquisition function PESM based Bayesian optimizer (by Harvard Intelligent Probabilistic Systems) alongside with an open source rapid aircraft development and analysis program OpenVSP (by NASA). The results of the optimizer were compared with industry standard MOO optimizers (B-MOO-s and genetic algorithm NSGA2)

## 1.2 Main objectives

| No | Requirement | Type |
|---|---|---|
| 1. | Flight range 10km | Mandatory |
| 2. | Flight/hover time 20 min | Mandatory |
| 3. | Lightweight UAV (<7 kg) | Mandatory |
| 4. | Max wingspan 1.5 m | Mandatory |
| 5. | Payload weight 100g | Mandatory |
| 6. | Expense without on-board computer and camera max 150€ | Mandatory |
| 7. | Autonomous flight capability | Nice-to-have |
| 8. | Brain-Computer interface | Nice-to-have |

**Thesis main objectives:**
1. Design a vertical take-off and landing fixed-wing drone.
2. Optimize the shape of the drone.
3. Build a brain-computer interface on the onboard computer.

## 1.3 Organisation of the Thesis

**Chapter 1 – Introduction**

This chapter explains the motivation behind the thesis tasks and justifies the relevance of the methods used for solving the problem. It also contains the full list of requirements set for the general design of the drone.

**Chapter 2 – Research Overview**

This chapter describes...

**Chapter 3 – Materials and Methods**

This chapter describes...

**Chapter 4 – Design**

This chapter describes...

**Chapter 5 – Validation**

This chapter describes...

**Chapter 6 – Brain-Computer Interface**

This chapter describes...

**Chapter 7 – Conclusions and Future Work**

This chapter describes...

# 2 Theory and Methods

In this chapter the theory behind the main concepts and methods used in the thesis are discussed (except for topics involving the brain-computer interface subtask, which is covered in Chapter 5):

- Theory of aircraft aerodynamics
- Aerodynamic simulation methods
- Propulsion and power system
- Multi-Objective Optimization Methods

## 2.1 Theory of aircraft aerodynamics

In section 3.1 the preliminary design concepts are compared using basic low-speed aerodynamics laws from Joseph Katz and Allen Plotkin textbook Low-Speed Aerodynamics. The formulas and theoretical background of concepts used when designing and optimizing the drone are discussed in this chapter.

The first appeal is to make the aircraft aerodynamically efficient: decrease drag force $D$ to save energy in forward flight and increase lift force $L$ to achieve flight at lower speeds and carry more weight. However, when achieving both of them by choosing a tailless aircraft, stability issues rise. First of which is the aircraft behaviour in a change of pitch - be it due to increase of speed or a gust of wind. If the aircraft is engineered unstable, the nose would pitch up and further increase lift, which can lead to a crash. The second is the tendency to start a wobbling motion called the ”dutch roll” when side wind forces the aircraft to yaw with respect to the airflow. This is an inherent problem to aircraft without vertical stabilizers.

### 2.1.1 Forces of aerodynamic bodies

As illustrated in the free body diagram (figure 2.1) the main forces acting on an aircraft in straight and level flight with no acceleration are in equilibrium. This means $L = W$ and $T = D$, where $L$ is total lift force, $W$ gross weight of the vehicle, $T$ thrust force generated by the propulsion system and $D$ the total aerodynamic drag force acting on the vehicle. This state of steady flight with no moments acting about the airplanes center of gravity ($CG$) is called trim. The main geometry parameters of a flying wing aircraft are shown on figure 2.2: span length $b$, sweep $\Lambda$, root chord $c_0$, tip chord $c_t$ and mean aerodynamic chord $MAC$.

Firstly, based on Newtons third law of motion, a wing can only produce as much lift force as much energy is exerted by changing the direction of the airflow through which the wing flies. Lift force is therefore directly connected how well the body curves the airflow down. The dimensionless total lift force coefficient of a body $C_L$ gives a good estimate of how well the geometry is designed in terms of aerodynamic lift (in a given environmental setup). Relations between the $C_L$ and the lift force necessary for the UAV to maintain flight are given by equation 2.1.

$$L = \frac{\rho S C_L Q_\infty^2}{2} \Leftrightarrow C_L = \frac{2L}{\rho Q_\infty^2 S}, \tag{2.1}$$

where $Q_\infty$ is the flight speed, $\rho$ the air density, $L$ the lifting force ($L = W$), $C_L$ the coefficient of lift and $S$ the projected surface area of the UAV.

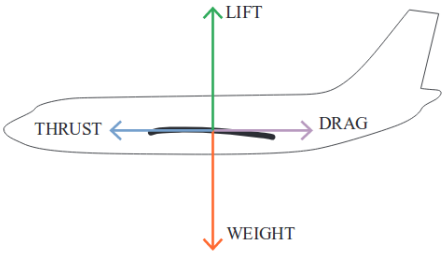



FIGURE 2.1: Free body diagram of forces acting on a trimmed airplane.

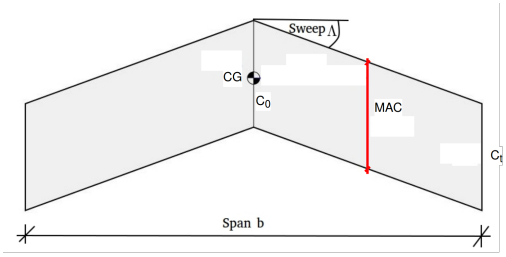



FIGURE 2.2: Basic flying wing geometry .

In essence the aircraft $C_L$ demonstrates lifting capability of the UAV given a certain physics environment (airspeed, fluid properties and the reference area). This value can be both negative (corresponding to negative lift), positive (positive lift) and has a zero value for no lift. In theory a cross-section $C_l$ maximum of 3.06 is possible to obtain for a non-varying airfoil section (cross-section of infinite length)[2]. In practice, for a simple airfoil geometry this is not possible and depending on the exact task maximum values between 1.0 to 2.0 are considered very good.

However, in the context of the final design optimization the overall $C_L$ value is not maximized, but kept at a certain minimum value with respect to the surface area in order to maintain enough lift $L$. This value value is therefore a constraint to decide the angle of attack $\alpha$ used for that particular wing iteration.

For a flying body to maintain stable flight without acceleration the propulsion system needs to overcome the drag force $D$. The total coefficient of drag $C_D$ can be used to approximate the thrust force $T$ necessary to overcome combined drag forces and maintain at a certain airspeed.

The two main components of drag produced by a wing are parasitic and lift-induced drag (equation 2.2). The other drag components: wave and ram drag are trivial for the low-airspeed UAV case and are therefore neglected. Relations between the $C_D$ and the drag force overcome to maintain a given flight speed are given by equation 2.3.

$$D = D_i + D_o; C_D = C_{D,i} + C_{D,o} \tag{2.2}$$

Parasitic drag is a combination of form, skin friction and interference drag, which are connected to the geometric size of the wing. Broadly, the increase in geometric values (chord length, projection area, surface area) result in the increase of parasitic drag. An analytical approximation exists for the parasitic drag $D_o$ at zero lift angle of attack:

$$D_o = \frac{\rho S C_{D_o} Q_\infty^2}{2} \Leftrightarrow C_{D_o} = \frac{2D_o}{\rho Q_\infty^2 S}, \tag{2.3}$$

where $Q_\infty$ is the flight speed, $\rho$ the air density, $D$ the drag force overcome to maintain speed ($T = D$), $C_D$ the coefficient of drag and $S$ the projected surface area of the UAV.

However, the lift-induced drag depends on pressure differences due to induced lift effects. As the airflow around the wing gets modified, demanding a higher angle of attack to generate more lift, drag is induced. This causes the airflow to produce wingtip vortices and thereby induce downwash behind the

wing. The induced drag coefficient $C_{D,i}$ can be found as follows.

$$C_{D,i} = \frac{C_L^2}{\pi AR}, \tag{2.4}$$

The parasitic and lift-induced drag are in an opposite nonlinear correlation with each other with respect to the flight velocity and therefore there is an airspeed where total drag is minimum (figure 2.3). However, at this speed the flight is unstable as fluctuations can cause loss of speed, which can have a snowball effect to the decrease of lift. Therefore, in practice a flight speed that is slightly above the minimum drag speed is chosen for optimality. The topics are further touched upon in the aerodynamic simulation methods section as calculating exact values for these parameters has much to do with the method used for solving them.

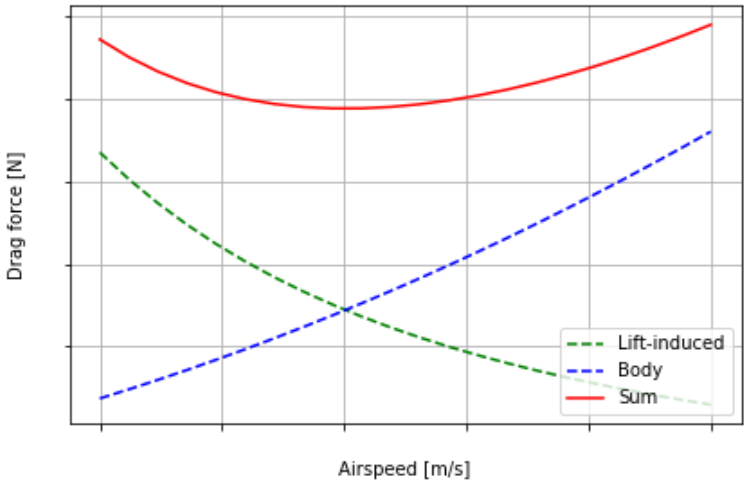



FIGURE 2.3: Illustration of drag component composition over time.

#### 2.1.1.1 Induced drag and wingtip vortices

Based on two papers by Ludwig Prandtl (1922 and 1933) on the ”lifting line” theory two lift distributions compete for minimizing induced drag. As stated in subsection 2.1.1 introduction, disturbances in the induced downwash cause induced drag. Prandtl proposed the elliptic lift distribution in his 1922 paper (figure 2.4a), which has a constant downwash angle. Although it has great performance characteristics in ideal conditions, it can stall abruptly and cause loss of control. This method is very difficult to achieve on swept wings and control surfaces are much more complex to design. It also has structural disadvantages as the wings would need to be very rigid up to the wing tips and in the trailing edge.

Prandtl pointed out the structural disadvantages of the elliptical lift distribution and proposed the bell-shaped lift distribution in his 1933 paper (figure 2.4b). As can be seen from the illustration, instead of having a constant downwash it has a smoothly varying downwash that crosses to an upwash near the tip of the wing. This causes vortices over the wing instead of having wingtip vortices and by optimizing a non-linear leading edge twist, the effect of the induced drag can be reduced enough to have the more efficient structure to outweigh an elliptical lift distribution wing. In 2016 a patent was applied for a PRANDTL-D wing by a NASA research team, who proved the efficiency and stability benefits for flying wing aircrafts. This method has proven to increase the maximum $L/D$ ratio by ca 10% using the same weight for structure as the elliptic wing. The method also reduced transversed yaw effect, which can lead to a Dutch roll (discussed further in subsection 2.1.2.2).

In practice it is very difficult to achieve ideal lift distributions and they are optimized very specifically to a certain cruising speed and climbing condition. Another method to have weaker wing-tip vortices is to have very long wings or winglets, which also act as an extension of the wing. As swept wings have wingtips far behind the $CG$, the winglets also act as vertical stabilizers and having airflow aligned to

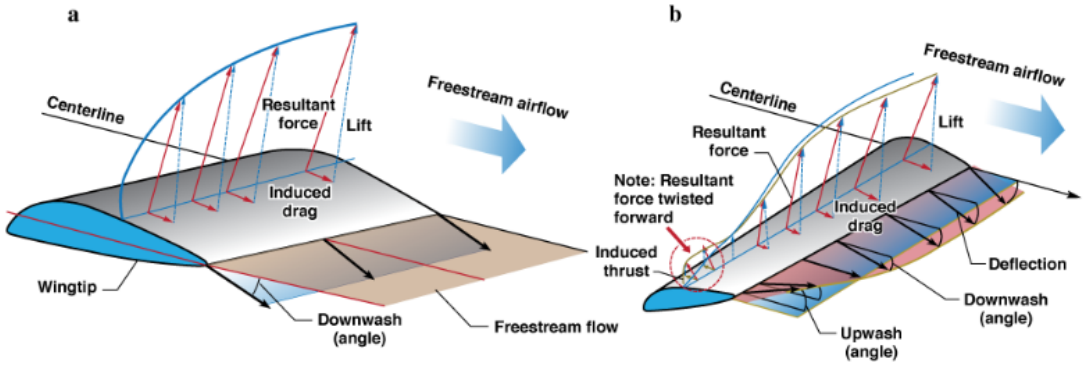



FIGURE 2.4: Comparison of the elliptic and bell shaped lift distribution effects.

the sweep angle, they also act as a sail increasing forward thrust. Using this same thrust the winglets can attribute to either negative or positive pitching moment. As the $CG$ of flying wings is designed to have a negative pitching moment, this effect can be designed to help balance it in case of deviation of the angle of attack.

### 2.1.2 Stability of aerodynamic bodies

Fixed wing aircraft have three stability axis through its center of gravity $CG$ (figure 2.5): lateral (or roll over $x$-axis), longitudinal (or pitch over $y$-axis) and normal (or yaw over $z$-axis). In flight the aircraft rotates around these axis, having a moment to describe each rotation: $M_x$, $M_y$ and $M_z$. These moments are evaluated by moment coefficients $C_{Mx}$, $C_{My}$ and $C_{Mz}$ respectively, which give insight on how the aircraft would behave in disturbances, such as gusts of wind or turbulence. Calculating the derivative values for these coefficients with respect to a disturbance, such as change in pitch or yaw angle gives good insight to how the aircraft would behave in the real life system. There are two sides to finding the optimal value: the more stable the airplane, the more difficult it is to maneuver.

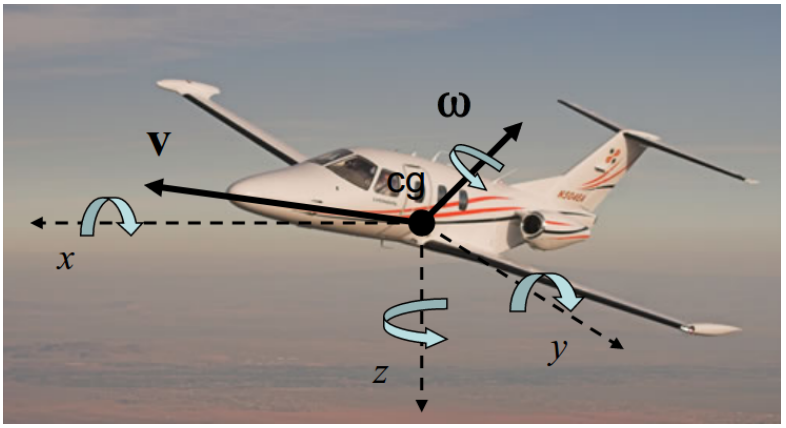
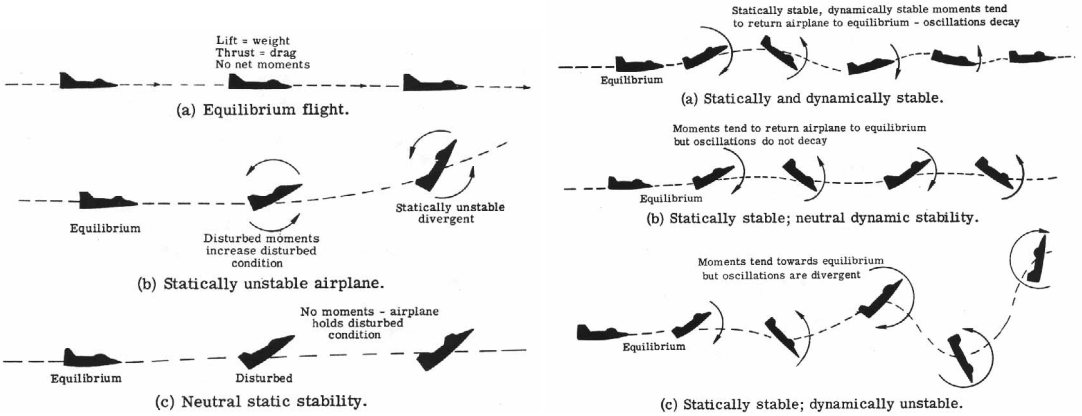



FIGURE 2.5: Illustration of aircraft stability axis orientation.

The stability can be further separated into static and dynamic stability. An aircraft that has static positive stability will return to its previous state of equilibrium after disturbances from turbulance or wind. In addition, if this aircraft has dynamic positive stability it will oscillate with decreasing deflection to the previous state, whereas negative dynamic stability would cause it to increase the deflection oscillation until stall. These two types for longitudinal stability are illustrated on figure 2.6. The relevant types of stability in respect to the aerodynamics optimization task and conceptual design reviews are discussed in the following subtopics: 2.1.2.1 Longitudinal static stability and 2.1.2.2 Dutch roll.




(A) Aircraft static stability. (B) Aircraft dynamic stability.

FIGURE 2.6: Wing.

#### 2.1.2.1 Longitudinal static stability

In order to achieve positive static stability in the longitudinal direction, the airplane needs to have an opposite change in pitching moment ($M_y$) to the change of the angle of attack $\alpha$:

$$\frac{dM_y}{d\alpha} < 0 \tag{2.5}$$

This would cause the aircraft to return to its original attitude as shown in figure 2.7. The pitching moment is defined as the sum of aerodynamic forces of pressure fields and drag force and is considered positive when it causes the nose to pitch up. The relation between $M_y$ and the pitching moment coefficient $C_{My}$ is described in equation 2.6. If the lift at the leading edge of the geometry causes a higher moment around the $CG$ than other force moments combined the aircraft will pitch its nose up (positive pitching moment direction).

$$M_y = \frac{\rho S c C_{My} Q_\infty^2}{2} \Leftrightarrow C_{My} = \frac{2M_y}{\rho Q_\infty^2 Sc}, \tag{2.6}$$

where $Q_\infty$ is the flight speed, $\rho$ is the air density, $M$ the pitching moment, $C_M$ the pitching moment coefficient, $c$ the length of the chord of the airfoil and $S$ the projected surface area of the UAV.

The stability condition would transform to:

$$\frac{dM_y}{d\alpha} = \frac{\rho S c Q_\infty^2}{2} \frac{dC_{My}}{d\alpha} \Rightarrow \frac{dC_{My}}{d\alpha} < 0 \tag{2.7}$$

The resultant aerodynamic force acts through a point called the center of pressure ($CP$), which is analogous to the weight distributions center of gravity ($CG$) (equation **??**). As there is no pitching moment ($C_{My} = 0$) in $CP$ this would intuitively be an ideal reference for designing the $CG$ location. This would be possible for a symmetrical airfoil, but most efficient airfoils used are asymmetric and the $CP$ location varies in respect to the change of the angle of attack:

$$CP = f(\alpha) \tag{2.8}$$

However, in engineering it is useful to use another point for reference: the aerodynamic center as in this point the pitching moment remains constant when the $\alpha$ changes:

$$\frac{dC_{My}}{d\alpha} = 0 \tag{2.9}$$

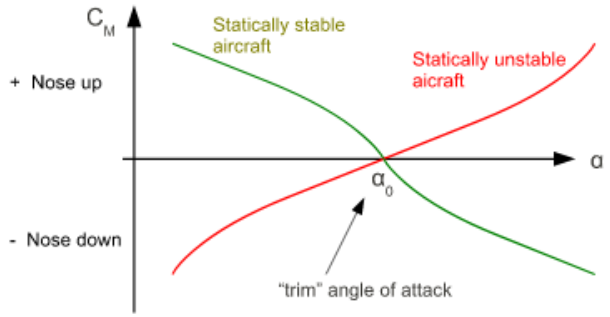



FIGURE 2.7: Comparison of statically stable and unstable aircraft's behaviour with respect to change of $\alpha$.

Therefore, in order to to have a negative pitching moment, the $CG$ should be placed fore of the $AC$. In case the aircraft consists of many aerodynamic bodies the resultant point of all their separate aerodynamic forces is found, the neutral point $NP$. If the $CG$ is placed in this position, the aircraft will have neutral longitudinal static stability as seen in figure 2.6a.

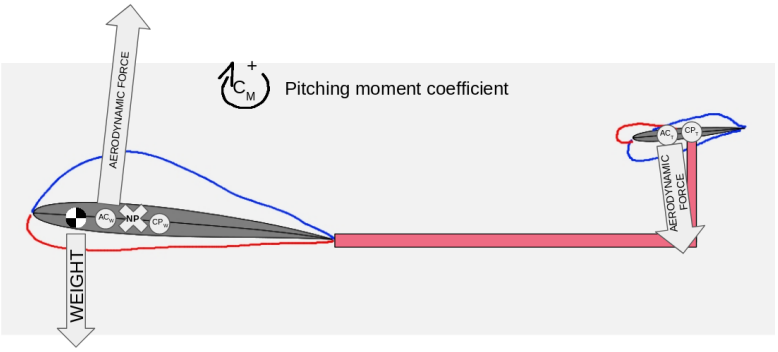



FIGURE 2.8: Pitching moment components of a conventional aircraft.

A conventional tailed aircraft has the horizontal stabilizer to counteract the moment caused by the $CG$ (figure 2.8). It is a self-regulating system, with which an increase of $\alpha$ would cause less downwards aerodynamic force from the tail and thus the moment stabilizes. Flying wing type aircraft have sometimes a modified airfoil, which acts in a similar manner. The airfoils trailing edge twists up (reflex), creating a downwards lift-force (figure 2.9). This effect decreases the overall lift of the aircraft and increases drag. Having swept wings helps reduce the negative effects by increasing the moment arm and bringing the $AC$ more aft.

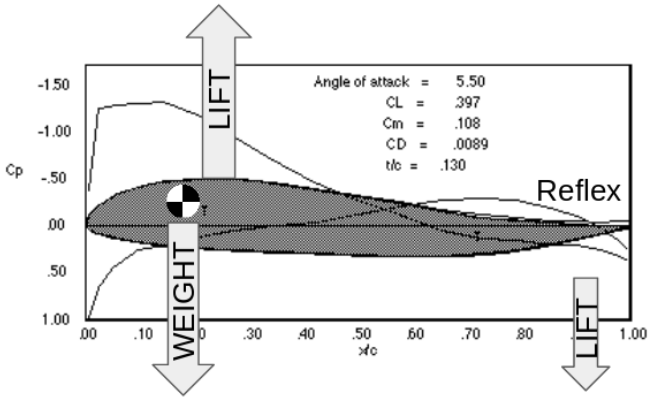



FIGURE 2.9: Coefficient of pressure distribution of a reflexed airfoil.

Finding a broad location of the $AC$ can be done using the following formula:

$$MAC = \frac{2c_0(1+\lambda+\lambda^2)}{3(1+\lambda)} = 0, \tag{2.10}$$

where $c_0$ is the root chord and $\lambda = c_t/c_0$ is taper ratio.

A more exact location of $AC$ can be found using iterative fluid flow methods, such as Vortex Lattice method and Computational Fluid Dynamics (subsection 2.2.2).

In the optimization section of the design chapter one of the optimized parameters is the derivative of $C_{My}$ with respect to change in pitching angle $\alpha$ this gives a good estimate of the static stability of the aircraft pitching moment. As we are dealing with a wing body UAV, the counteracting surfaces have very little effect to balance the pitching moment. Therefore, in order to achieve a stable flying wing, the derivative of $C_{My}$ could be minimized by a counteracting $CG$, but in the case of a tricopter flying wing, its position is constrained by the propellers setup. To restore some of the stability the wing sweep can be increased, which increases drag and changes the effect of lift. Higher Aspect Ratio ($AR$, equation 2.11) UAV-s on the other hand have a higher $C_L$ and lower $C_D$ (long range aircraft have $AR \approx 8$), but decrease the surface area of the wing and therefore decrease the lift (and drag) as well.

$$AR = \frac{b^2}{S}), \tag{2.11}$$

where $b$ is wing span and $S$ wing planform area.

Therefore, there are a vast amount of parameters that could be tuned as well as output parameters that could supplement the assessment of the aircraft. After careful consideration and lengthy trials the parameters given in this subsection were chosen. Some were chosen on the basis of best practice, analytical evaluation and robustness, others due to solver and geometry software issues (described in more detail in the Methodology chapter).

#### 2.1.2.2 Dutch roll

The second stability issue affecting tailless aircraft is a dynamic instability effect called dutch roll, which can occur due to yaw or roll deviations and take long to dampen. The dutch roll is a sequential alternation between a yawing and rolling motion of tumbling. If the swept flying wing yaws, the wing that rotates forward will have an increase in both drag and lift. The first of which will cause the motion to dampen and rotate back towards the original state. The second will cause that side of the aircraft to roll up a bit, thus increasing the drag and wobbling into a yawing motion. This dynamic motion is illustrated on figure 2.10.

On conventional aircraft this oscillatory instability is tackled with a response from the rudder. Flying wings can reduce the effect by preventing the lift force from increasing/decreasing by adding wing fences and winglets that prevent the air from span-wise sweeping over the wing. Having winglets underneath the drone has three positive side effects: decreased drag, increase of roll and the pitching moment stability.

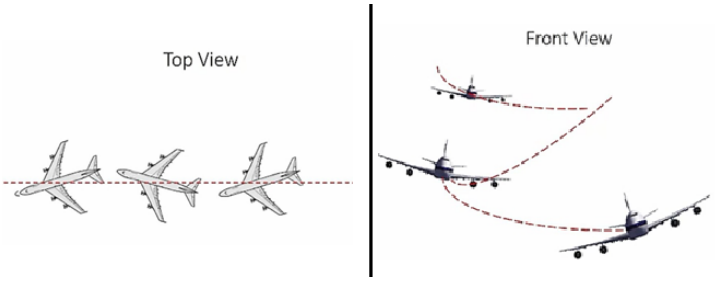



FIGURE 2.10: Dutch roll illustrated with a Boeing 747.

### 2.1.3 Drone propulsion

The propulsion system encompasses the sequence starting with choosing a suitable propeller and propeller speed, which is the basis to making an estimation to choose the power parameters of the motor, electric speed controllers (ESC-s) and batteries. As the chosen design has a tilt-rotor tricopter setup, the propulsion system should be optimized to be sufficient for hovering and forward flight. I the design chapter estimates are made to assemble a suitable propulsion system and in the results chapter these estimates are validated.

#### 2.1.3.1 Hovering

The momentum theory is an analytic approach to conserve global balance of mass, momentum, and energy to resolve forces of various flight modes. In this subsection relevant equations and solving sequence for the hover condition are presented. The main intuition is similar to lift aerodynamics: to conserve hover the massflow of air through the rotor plane must be in equilibrium with the rotor-crafts weight (figure 2.11). The rotor disc a has radius $R$, area $A$ and is spinning at an angular velocity of $\Omega$ (rad/s). There are $N_b$ number of blades that have a chord length $c$ and zero-lift drag $C_{d0}$. The solidity of the rotor is $\sigma$.

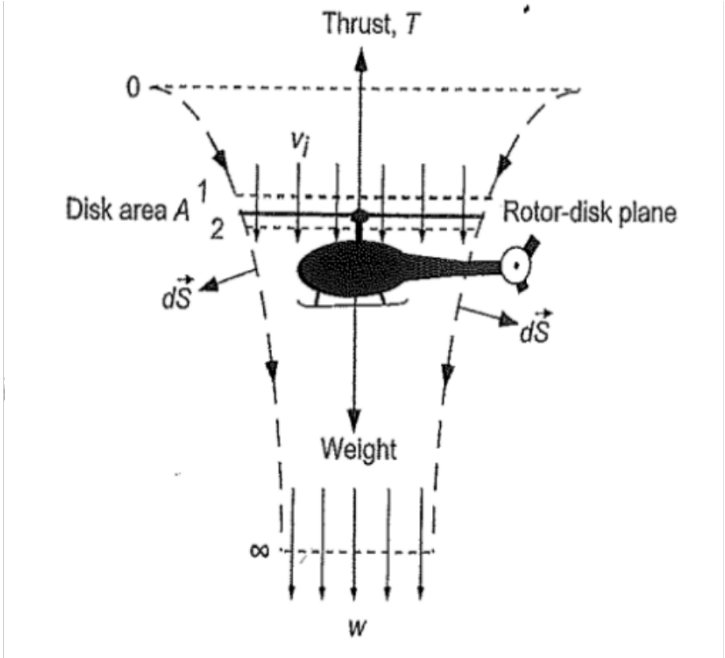



FIGURE 2.11: Basics of an ideal momentum theory based hovering system.

The ideal momentum theory assumes:

- Flow is considered 1D and uniform through the rotor disk.
- Flow is incompressible, inviscid and irrotational.
- The rotor is viewed consisting of infinite number of blades adding momentum to the flow.
- The pressure difference on either side is discontinuous, while the velocity is continuous.
- The wake is a uniform jet (meaning no swirl).

The thrust coefficient can be expressed as

$$C_T = \frac{T}{\rho_{air} A \left(\Omega R\right)^2} \tag{2.12}$$

The non-ideal power coefficient is given as

$$C_p = \kappa \frac{C_T^{3/2}}{\sqrt{2}} + \frac{\sigma C_{d0}}{8} \tag{2.13}$$

Non-ideal losses are expected to add around 15% so $\kappa = 1.15$.

One of the design parameters is the power loading which is the thrust generated relative to power used

$$\text{PL} = \frac{T}{P_1} = \frac{C_T}{(\Omega R)\, C_p} \tag{2.14}$$

The hover time can then be found from energy capacity:

$$t = \frac{E_{available}}{P_{use}} \tag{2.15}$$

#### 2.1.3.2 Transition to forward flight

The transition effects from hover to forward flight and characteristics of forward flight are found analytically using the momentum analysis for forward flight. In addition to the forces of lift from wing, thrust from rotor, weight of the aircraft and drag forces of the aircraft the characteristics of rotor discs are considered as well. The rotor momentum theory for forward flight gives an estimate of the power consumption of the propellers and stable states with an airspeed $Q_\infty$ and rotor angle $\beta_r$. $_r$ is measured from hovering position and therefore a value of 90° indicates forward flight mode.

To find energy losses of the rotors, the induced velocity at the rotor disc needs to be calculated:

$$v_i = \frac{v_h^2}{\sqrt{(Q_\infty \cos\beta_r)^2 + (Q_\infty \sin\beta_r + v_i)^2}}, \tag{2.16}$$

which can be solved iteratively and where the hover velocity is

$$v_h^2 = T/2\rho A \tag{2.17}$$

The ideal rotor power is then be estimated

$$P_i = T(v_i + Q_\infty \sin\beta) \tag{2.18}$$

15 % loss for non-ideal effects and drag estimations are added to $P_i$ yielding a total estimated power demand

$$P = 1.15P_i + P_0, \tag{2.19}$$

where

$$P_0 = \frac{1}{8}\rho N_b \Omega^3 c C_{d0} R^4 \tag{2.20}$$

### 2.1.4 Propulsion power system

As can be deduced from equations 2.14 and 2.20 a key factor of hover and forward flight time is angular speed $\Omega$. The power loading $PL$ has an inverse relation to $\Omega$ and the power losses a cubic relationship to $\Omega$. Therefore, propulsion systems using lower voltages and motors with lower voltage to angular speed rating demand less power and are more efficient. Lower voltage systems have a lower maximum thrust level and batteries have lower discharge rates. As we are dealing with a tricopter, which thrust per rotor is semi-high an optimal middle ground should be found.

The discharge rate also sets some limits when choosing battery types. Although some Li-ion type batteries have better gravimetric ($Wh/kg$) as well as volumetric ($Wh/l$) energy densities (figure 2.12), Li-Po batteries should be preferred as they have more than five times higher discharge ratings. The discharge rate is a rating by manufacturers to find the maximum current that can be constantly drawn. It can be found by using the following formula:

$$C_r = \frac{P_{use}}{UE_r}, \tag{2.21}$$

where $C_r$ is C-rate needed, $U$ is the power system voltage and $E_r$ is rated energy stored in Ah (rated capacity of the battery given by the manufacturer).

The hover time can then be found as

$$t = \frac{E_{available}}{P_{use}}, \tag{2.22}$$

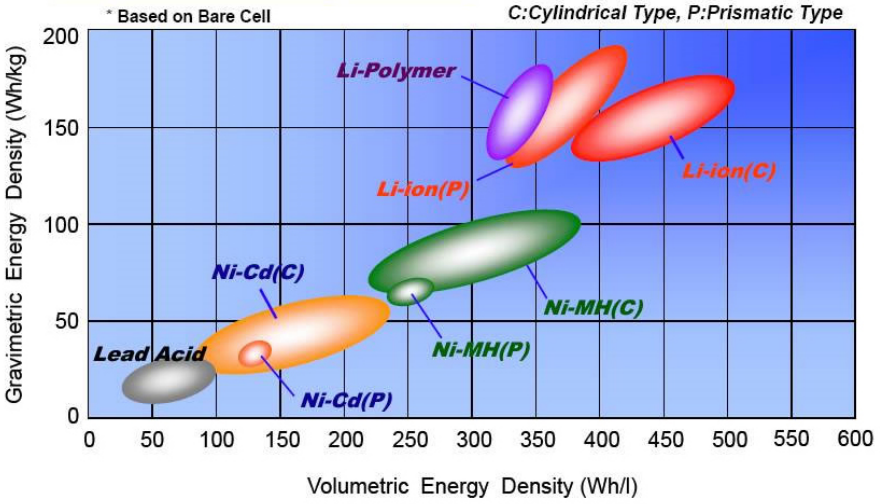



FIGURE 2.12: Overview of volumetric and gravimetric energy densities of batteries by compound.

Based on decisions of power supply and propeller dimensions, a suitable motor should be chosen. Brushless motors type is favoured among drone builders as they are more durable, efficient and most importantly weigh less compared to brushed motors. The other aspect to consider is torque moment arm: inrunners have a shorter arm than outrunners. Therefore, outrunners have higher torque and a more suitable angular speed for multirotors.

## 2.2 Aerodynamic simulation methods

In this section the physics solvers used for aerodynamic design are discussed. As correlation between geometric parameters and the aerodynamic properties of the vessel are not analytically solvable a variety methods can be used to find the optimum. Selecting a suitable physics solver for the task depends on a balance between insight to the geometry, accuracy and the results and computational cost. When using cheap physics solvers in a small feature space Grid Search can be an option to guarantee the global optimal solution. However, for each added parameter the total cost of the search gets exponentially bigger.

Two physics solver methods were used: 3D Vortex Lattice Method solver (VLM, program: OpenVSP) for global optimization and design review and a 3D Computational Fluid Dynamics Finite Volume Method solver (CFD CVM, program: Ansys Fluent). For meshing SnappyHexMesh from OpenFOAM was used, but for simulations it is much less stable and problems are more difficult to solve than Ansys Fluent. Considered solver methods are given in table 2.1. As Tornado lacked many of the features to build am automated optimization system, OpenVSP was used instead.

| Solver | Case | Method (interaction language) | CPUh | accuracy | usage |
|---|---|---|---|---|---|
| **XFoil** | 2D airfoil | Panel Method (command line) | low | low | easy |
| **OpenVSP** | 3D UAV | Vortex Lattice Method (Python) | medium | low | medium |
| **Tornado** | 3D UAV | Vortex Lattice Method (Matlab) | medium | low | medium |
| **OpenFOAM** | 3D UAV | CVM NS CFD (C++) | very high | medium | difficult |
| **Ansys Fluent** | 3D UAV | CVM NS CFD (C++) | very high | medium | medium |

TABLE 2.1: List of considered physics solvers.

### 2.2.1 Computational Fluid Dynamics

Computational Fluid Dynamics Control Volume Method takes a volumetric approach to iteratively solve the governing Navier-Stokes equations (NS). This is a thorough and CPUh costly method. The simulation space is divided into small control volumes (figure 2.13) and initial values for crucial elements of fluid flow are generated to the sides of those volumes. At each iteration the partial differential equations of the NS are solved over the volumes. When overall mass, energy and momentum balance and residuals converge, the iterations stop.

The method is very sensitive to the control volume mesh quality. As quality automatic CVM mesh creation is difficult to achieve and even airfoil analysis can take a few days to converge, it is not suitable for global optimization, but is used to verify the results.[3]

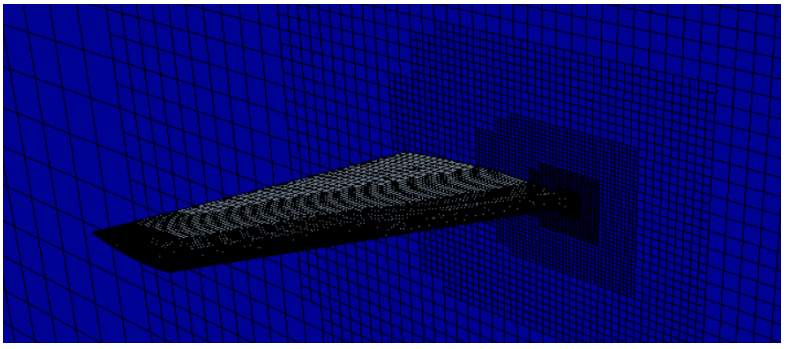


FIGURE 2.13: Example of an OpenFOAM mesh of Onera M6 wing.

### 2.2.2 Vortex Lattice Method

Similarly to CVM CFD iterating the values of a physics control volume environment to a close to equilibrium state vortex lattice method divides the body of the aircraft into sections. Each section is then solved separately using the panel method by assuming in that position a wing of infinite span and non-varying cross-section. In this method NS equations are not solved. Instead it uses an inviscid linear-vorticity panel method with a Karman-Tsien compressiblity correction to solve the airfoil as a 2D case by dividing it into a few hundred small panels. XFoil, which is used within this project for preliminary comparison of optimization algorithms, uses the same technique for 2D airfoil analysis (figure 2.14).

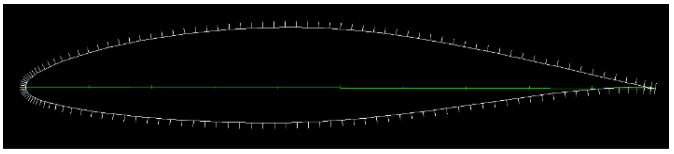


FIGURE 2.14: Example of an XFoil airfoil mesh.

The VLM builds on the panel method solutions by cutting the aerodynamic body into sections and interpolating the results over the span (figure 2.15). On the course of this the model is considered as an infinitely thin sheet of vortices. Thus, the method neglects influence from thickness and viscosity. This simplified method results with very low CPUh requirements and has lower accuracy outside specific conditions.

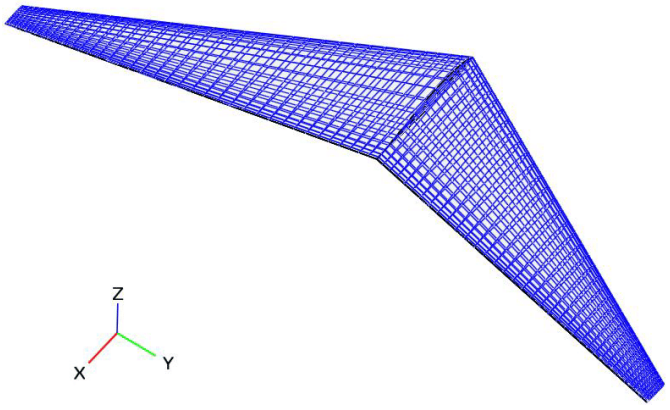



FIGURE 2.15: Example of an OpenVSP flying wing section mesh.

Inaccuracies occur when compressibility and thermal energy needs to be balanced and also near stall and turbulent conditions[4]. VLM is also limited to wings with $AR > 3$ as the rotational flow effects that reach the mid chord are not accounted for. Furthermore, as is also shown in the validation subsection lifting line based VLM method does not work well with thin plates outside $\alpha = \pm 5°$ as the frontal area growth, which increases considerably is not accounted for. The CFD solver solved the flat plate problem at lower and higher angles of attack well.

Despite the drawbacks, within these limitations complex wing geometries can be analyzed using the method and since OpenVSP has an option to blend the wing sections, it is ideal for BWB UAV prototyping. Also, the computational cost is quite cheap: approximately 15 min per model.[5] Tornado,

As OpenVSP outputs values of $CM_y$ it can be used to find the $AC$ position of the aircraft after solving the simulation at two different angles of attack. By comparing the coefficients of moment and lift of the aircraft at two different angles of attack $\alpha_1$ and $\alpha_2$.

$$AC = \frac{C_{My1} - C_{My2}}{C_{L1} - C_{L2}} \tag{2.23}$$

## 2.3 Multi-Objective Optimization Methods

In the optimization task of the project a Bayesian multi-objective optimization method (B-MOO) is used to search for a global optimum in the the less intuitive portion of the decision space. When selecting the optimization method, a state-of-the-art B-MOO acquisition function Predictive Entropy Search (PES, by Harvard Intelligent Probabilistic Systems []) is compared to conventional B-MOO types (table 2.2) and to a non-dominated sorting genetic algorithm NSGA2.

| **Method** | **Full name** |
|---|---|
| **PES** | Predictive Entropy Search |
| **EHI** | Expected Hypervolume Improvement |
| **ParEGO** | Pareto Efficient Global Optimization |
| **SmsEGO** | S-Metric Selection-based Efficient Global Optimization |
| **SUR** | Sequential Uncertainty Reduction |

TABLE 2.2: List of implemented Multi-Objective Optimization methods.

The optimization task is set up as a black-box, sending the parameter vector to the physics solver (adding points to the decision space) and retrieving the results vector to the objective space (figure 2.16).

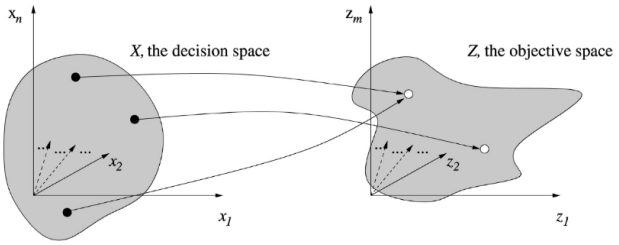



FIGURE 2.16: The multi-objective decision and objective space paradigm.

As in aerodynamics analysis complex correlation between certain geometry parameters and the results are not analytically solvable a variety methods can be used to find the optimum. When using computationally cheap physics solvers in a small feature space Grid Search can be an option to guarantee the global optimal solution. However, for each added parameter the total cost of the search gets exponentially bigger. Considering an evaluation time of 1 s per parameter evaluation would mean 10 s for a 10 sample point grid with just one tunable parameter. When increasing the number of parameters to 10, the total time consumption increases to $10^{10}$ seconds, which is more than 300 years. The ability to have a complex and fully controllable geometry illustrates the curse of dimensionality.

Tackling the problem, scientists and engineers have both developed functions to create a smaller and more efficient decision space. As an the Kulfan CST curve can produce a greater number of aerodynamically efficient airfoil shapes with the same number of control points than its predecessors. However, with the growing functionality of a curved line the intuition and convexity of the optimum solution tends to become more intricate. With these tasks gradient based numerical solvers are not very efficient either - thus, global optimizers and surrogate modeling are used. The latter seem very promising to give engineers a real-time feedback to the designs physics condition (e.g new modules of Ansys [**?** ]), whereas the former are more suitable for solving undocumented scenarios. Bayesian Optimization is emerging as an efficient way to tackle the curse of dimensionality related problems, when finding a global optimum.

### 2.3.1 Bayesian Optimization

Bayesian Optimization is a method that utilizes previous knowledge to fine-tune the input space parameters so that the next test case explores the feature space where it has the highest potential of increasing the reward. In essence it utilizes the Base Theorem on Gaussian Processes ($\mathcal{GP}$) as shown in equations 2.24 and 2.25.

$$P(f|\mathcal{D}) = \frac{P(f)P(\mathcal{D}|f)}{P(\mathcal{D})} \tag{2.24}$$

$$P(f|\mathcal{D}) = \mathcal{GP}(f; \mu_{\mathbf{y}|\mathcal{D}}, K_{\mathbf{y}|\mathcal{D}}) \tag{2.25}$$

Gaussian Processes ($\mathcal{GP}$), in this context, can be thought of as multivariate Gaussian distributions over an infinite dimensional space based on previous evaluations ($\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$). The resulting $\mathcal{GP}$ (equation 2.26) is a set of distributions over functions described by a mean function $\mu$ and covariance function $K$. The resulting probability map can be used to predict the loss value for the next evaluation.

$$f(\mathbf{X}) \sim \mathcal{GP}(\mu(\mathbf{X}), K(\mathbf{X}, \mathbf{X}')) \tag{2.26}$$

An acquisition function is used to determine which evaluation setup would be the most valuable to observe (equation 2.27) and the real and predicted loss are compared after the new experiment.

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n) \tag{2.27}$$

New Gaussian Processes are formed based on all prior experiments as the next iteration begins. The basic procedure of Bayesian Optimization for N iterations is shown in Algorithm 1. The main hyperparameters to tune is the acquisition function and the kernel.

**Algorithm 1** The Basic Bayesian Optimization Algorithm

Surrogate function f to propose a vector of Pareto optimal points $\mathbf{X}_P$

1: **for** $n = 0, 1, ...N$ **do**
2: $f(\mathbf{X}) \sim \mathcal{GP}(\mu(\mathbf{X}), K(\mathbf{X}, \mathbf{X}'))$
3: $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}|\mathcal{D}_n)$
4: $\mathbf{y}_{n+1} = f(\mathbf{x}_{n+1})$
5: $\mathcal{D}_{n+1} = \{\mathcal{D}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$
6: **end for**

### 2.3.2 Bayesian MOO tuning

Acquisition functions determine the balance between exploration (evaluating points around low mean) and exploitation (evaluating points near high uncertainty) of the space. The following functions are considered and compared: Expected Hypervolume Improvement (EHI), Sequential Uncertainty Reduction (SUR), Pareto Efficient Global Optimization (ParEGO), S-Metric Selection-based Efficient Global Optimization (SMSego), Predictive Entropy Search (PES). A single objective task and four acquisition functions were programmed in Python to visualize and compare the process (figure 2.17).

Expected Improvement (equation 2.29) is widely used because of its property to not get stuck in local minimas as its predecessor Probability of Improvement (PI) (equation 2.28) does. In the equations$\Phi$ denotes the cumulative density function of standard normal distribution and $\phi$ denotes the probability density function of standard normal distribution. The PI only selects the next point with the highest

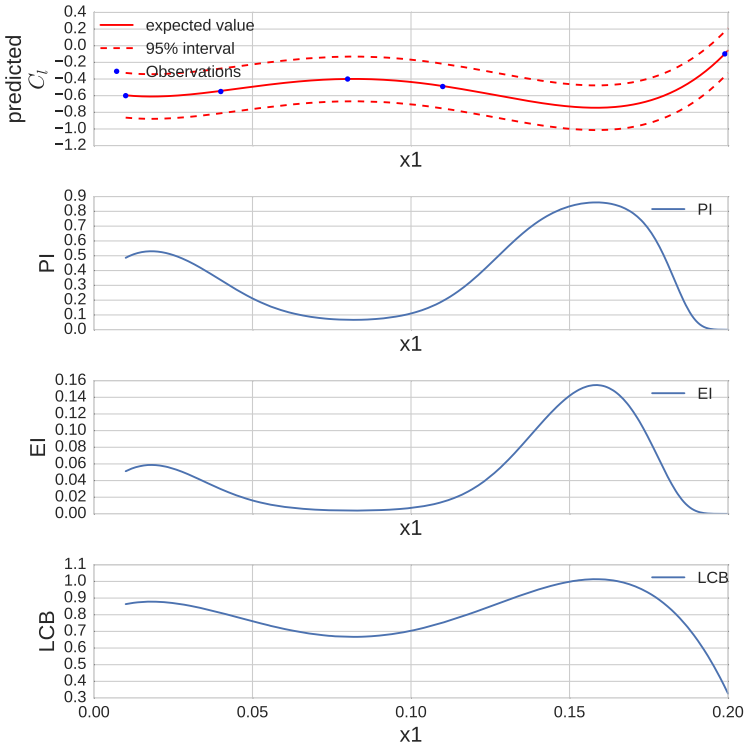



FIGURE 2.17: Comparison of acquisition functions proposing the next test point.

variance below the optimal evaluation in the tested data set $\mathcal{D}$. The EI on the other is a tradeoff between exploitation (a low mean value) and exploration (high variance).[6]

$$\alpha_{\mathsf{PI}}(\mathbf{x}; \mathcal{D}_n) = \Phi\left(\frac{\hat{y} - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})}\right) \tag{2.28}$$

$$\alpha_{\mathsf{EI}}(\mathbf{x}; \mathcal{D}_n) = (\hat{y} - \mu_n(\mathbf{x}))\Phi\left(\frac{\hat{y} - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})}\right) + \sigma_n(\mathbf{x})\phi\left(\frac{\hat{y} - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})}\right) \tag{2.29}$$

EHI is an EI algorithm utilizing hypervolumes for multi-objective optimization, finding Pareto optimal solutions. ParEGO, SMSego and SUR have their roots in EI to some degree. ParEGO uses weights to convert a multi-objective problem into a single objective problem, thus becoming a fairly inexpensive analysis $O(N)^3$, where N is number of observations made.[7]

SMSego uses the Pareto front in a multi-objective framework: utilizing the S-metric or hypervolume of the Pareto front to decide which solution is evaluated next. Maximizing the hypervolume within a real multi-objective optimization shows superior results compared to ParEGO, but also has a higher computational cost $O(KN^3)$, where K is the number of objectives.[8]

SUR also utilizes the Pareto front hypervolumes for real multi-objective optimization, but has a more complex strategy of sequential reducing by sampling entropy (an uncertainty measure)[9]. In essence SUR looks for the next point, which reduces the uncertainty of the minimizer. SUR is considered computationally very expensive as it relies on voluminous GP simulations. EHI, SMSego and SUR are limited to 2-3 objective functions.[10]

A different method, PES or PESM for a multi-objectvie case is a recent method to tackle more than 3 objectives without converting them into a single objective problem. PES chooses evaluations to reduce the entropy of the posterior distribution over the Pareto set. Unlike the other methods discussed, Entropy Search based methods do not have closed-form expressions of the distribution and evaluating the expected utility is much more complex. The computational cost of PES is in a linear connection to

the size of objectives K and it allows decoupling of evaluations. Therefore, PESM is the most prominent acquisition function reviewed during this project. PESM is much more complex regarding its algorithm, but is described as equation 2.30. [7]

$$\alpha_{\text{PES}}(\mathbf{x}) = H[\mathbf{x}_*|\mathcal{D}] - \mathbb{E}_{\mathbf{y}}\{\mathbf{H}[\mathbf{x}_*|\mathcal{D} \cup (\mathbf{x}, \mathbf{y})]\}, \tag{2.30}$$

where the next decision $\mathbf{x}_*$ should be made based on the entropy given data $H[\mathbf{x}_*|\mathcal{D}$ and the expectation $\mathbb{E}_{\mathbf{y}}$ with respect to the entropy and given data. This form of the acquisition function is not feasible, therefore machine learning engineers work out methods to utilize the essence of the model.[7]

The other major parameter of B-MOO, kernel, determines how the distributions are formed over the mean values (e.g how smooth, ”wiggly” or periodic they are). The default Spearmint GP kernel is used and is based on a Matérn 5/2 covariance function, which according to gptools documentation is one of the fastest of the Matérn kernels.[11] The Matérn kernels (equation 2.31) are controlled by the length-scale (l) and a smoothness paramete $\nu$. The parameter $\nu$ gives extra flexibility to better adapt to the true underlying function. It is suggested in literature that 3/2 and 5/2 might be most suitable for this projects problem as 1/2 is too rough and 7/2 would not work well in noisy data.[12]

$$k_M(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}(\frac{\sqrt{2\nu}r}{l})^{\nu}K_{\nu}(\frac{\sqrt{2\nu}r}{l}) \tag{2.31}$$

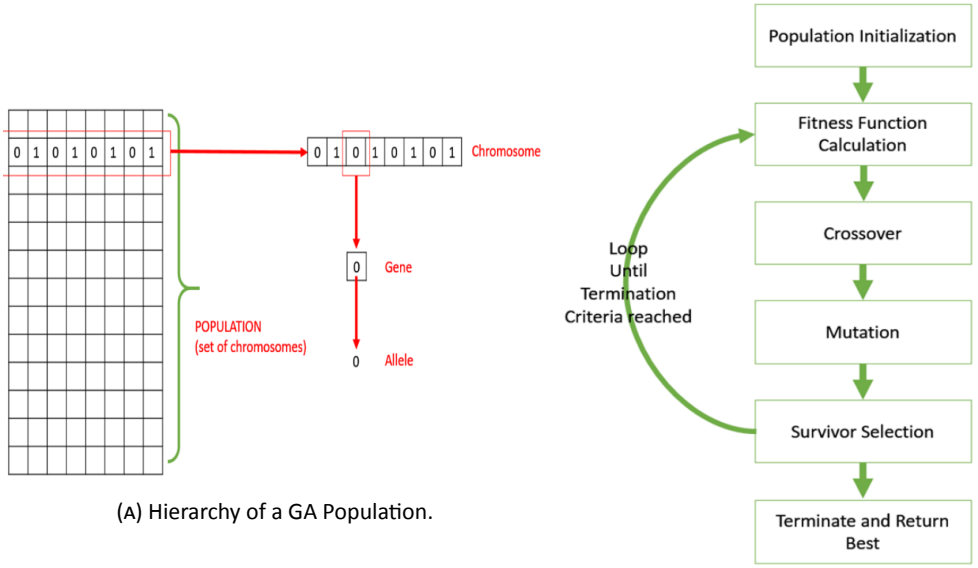
A stable and well-capturing kernel was very difficult to compose and default settings yielded the best along with fine-tuning $\alpha$ (a noise correction metric) to the data. The results shown on Figure 2.17 were retrieved with $\alpha = 1e - 10$. Although it is document in numerous documentation that PI would have issues with finding a global minima, it seems to rank pretty close to the EI. However, EI, seems to stake more on exploration as its peak is steeper. LCB seemed to yield flatter values and therefore might be slower when dealing with a non-convex problem.

### 2.3.3 Comparison to NSGA2

Genetic Algorithms (GA) is a large set of search based optimization techniques inspired by the natures process of natural selection (figure 2.18a). Unlike Bayesian Optimization methods discussed in the previous subsection, GA-s evaluate new points by generation, which consist of many chromosomes (selected evaluation points). Each chromosome consists of genes, i.e the decision space parameters, which have values (alleles) for that specific evaluation point.[13] The algorithm (figure 2.18) is initialized by creating a random or engineer selected generation. The generation is then evaluated by a fitness function. Methods for crossover, mutation, elitism and survivor selection can be used to create the next generation of chromosomes. The next chromosomes can be created by choosing the fittest chromosomes and combining their gene alleles. Mutation is used to further enhance finding a globally optimal solution as this gives random values to some chromosome genes. If the termination criteria has been reached (either a given number of iterations or a certain fitness value), the algorithm loop stops.

In the preliminary steps of choosing a MOO, the EI B-MOO was compared to a non sorting genetic algorithm NSGA2. NSGA2 is a simple Pareto efficiency based genetic algorithm, which is able to solve non-convex and non-smooth problems enforcing constraints using a tournament selection-based strategy. Mutation is also possible in this evolutionary algorithm.[14]

A 2D decision and 2D objective space was optimized: the impact of the airfoil leading edge upper and lower curve variables $x_1$ and $x_2$ were were compared to $C_l$ and $C_d$. Using Kulfan CST and Bezier curve Python scripts to form an airfoil the XFoil 2D physics solver evaluated proposed airfoil shapes. While the EI was simple enough to be coded up from scratch, NSGA2 was used from a Pyhton optimization package DEAP[15]. Although the Kulfan-CST airfoil curve generator has more flexibility to explore the decision space geometry, the Bezier curve was used in the following analysis as XFoil became unstable to many seemingly smooth shapes. The NSGA2 parameters mutation, generation size and number of generations were considered when tuning the model. Given that the maximum number of evaluations

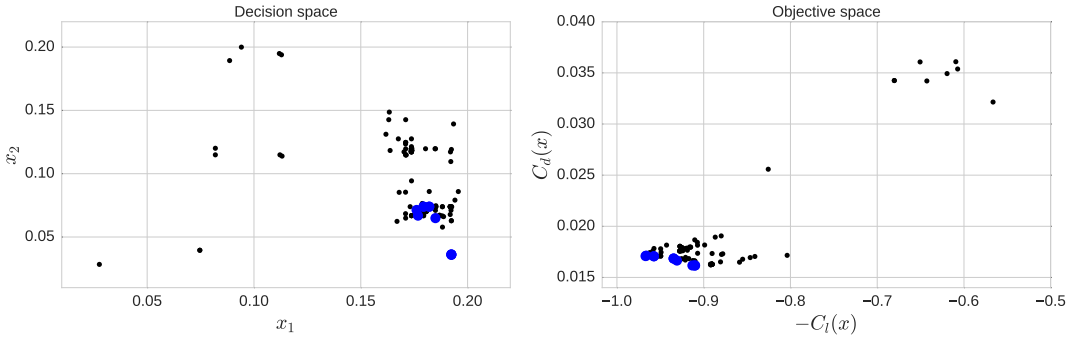



(A) Hierarchy of a GA Population.

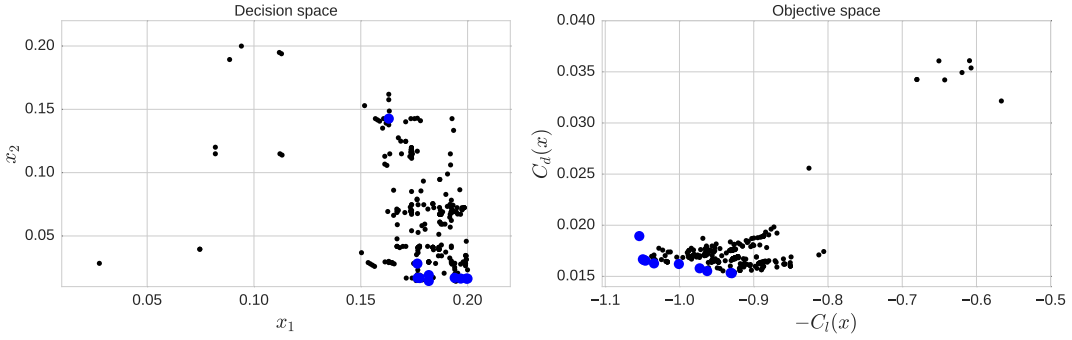
(B) A typical genetic algorithm sequence.

FIGURE 2.18: Basics of Genetic Algorithms.

would be 300, 10 genoms by 30 generations with a mutation factor of 0.2 yielded the best results (Figure 2.19a). After 100 data points the best $C_l$ value is 0.967 and after 300 trials 1.05 (Figure 2.19b). It can be observed that the algorithm decreases the $x_2$ value in the decision space to achieve improvement from iteration 100 to iteration 300.



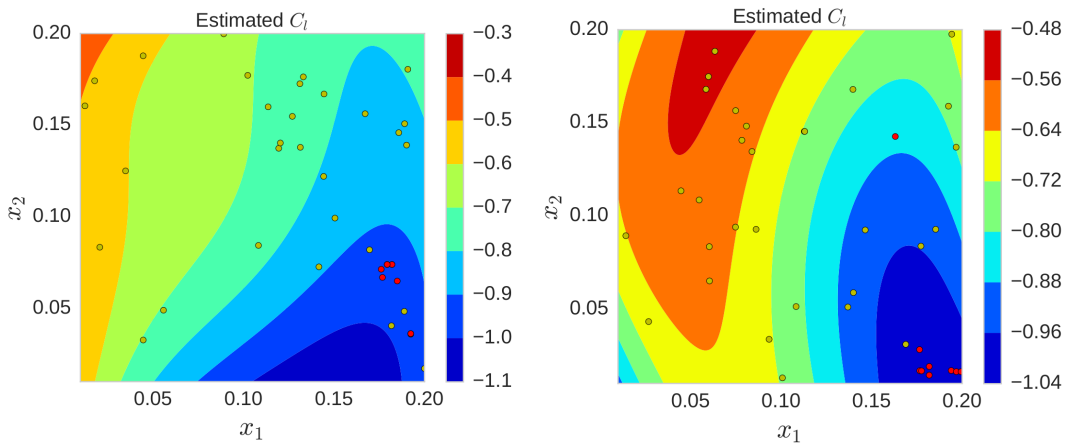

(A) 100 NSGA2 examined data points, Pareto-optimal solutions given by blue dots.




(B) 300 NSGA2 examined data points, Pareto-optimal solutions given by blue dots.

FIGURE 2.19: NSGA2 Decision and Objective space paradigm.

Using EI B-MOO, XFoil was first run 30 random data points to generate (within the same constraints as NSGA2) and the prediction results were compared to NSGA2 solutions (figure 2.20). The EI-s best suggestion data point was run and the $C_l$ value was 1.0495 (GPR predicted 1.084). This illustrates why Bayesian methods clearly have an advantage over GA-s finding a global minimum value of a multi-objective and multi-variable problem. As can be seen from figure 2.20a, although after 30 random tests EI has a pretty solid estimation of the optimal area, NSGA2 at 100 data points is farther from that area.




(A) NSGA2 at 100 evaluations compared to a Bayesian contour map based off 30 random points.

(B) NSGA2 at 300 evaluations compared to a Bayesian contour map based off 30 random points.

FIGURE 2.20: NSGA2 optimization results (red dots are optimal) compared to a Bayesian paradigm.

# 3 Design and Optimization

In this chapter the drone is designed and aerodynamic optimization is initialized. After working through a general design morphological table, preliminary assumptions are made to the shape, structure, weight and propulsion system of the aircraft. Once the main frame and mechanisms are chosen the aerodynamic optimization task is set up and solved. Results of the optimization task and validation are discussed in the results and validation section (Chapter 4).

## 3.1 General Design Concept

Over a dozen general concepts were researched and/or experimented with (based on the morphological matrix in table 3.1) before choosing the final solution (shown in bold). Following is an analysis of top 6 solutions considered.

| No | Vertical Propulsion | Horizontal Propulsion | Transition Method | Horizontal Stability | Vertical Stability | Roll Stability |
|---|---|---|---|---|---|---|
| 1. | One propeller | One propeller | Separate propulsion systems | Tail horizontal stabilizer | Tail vertical stabilizer | Tail vertical stabilizer |
| 2. | Two propellers | Two propellers | **Tilt-rotor** | Vectored thrust | Vectored thrust | Vectored thrust |
| 3. | **Three propellers** | **Three propellers** | Tilt-wing | **Winglets** | **Reflex of airfoil** | Counter rotating propellers |
| 4. | Four propellers | Four propellers | **Control surfaces** | **Control surfaces** | **Control surfaces** | Tail rotor |
| 5. | Jet propulsion | Jet propulsion | | | | **Bell-shaped lift distribution** |
| 6. | Duct Fan | Duct Fan | | | | **Control surfaces** |

TABLE 3.1: Morphological table of general design concept of the drone.

### 3.1.1 Design Concept 1 - dual tilt-propeller tail-sitter

The dual tilt propeller tail-sitter (figure 3.1) takes off and lands on its tail, or in this particular sketch on its winglets. The transition from vertical hover to forward flight is made by using two tilting rotors and control surfaces as levers. The simplicity of the design makes it light-weight and aerodynamic. It is, however, difficult to keep stable and hovering demands much power. Increasing control surfaces and using an airfoil with reflex could help overcome the first impediment. This will not help in case there is wind. As the propellers would be used for forward flight as well, an optimal propulsion system that is effective in both flight modes might be impossible.

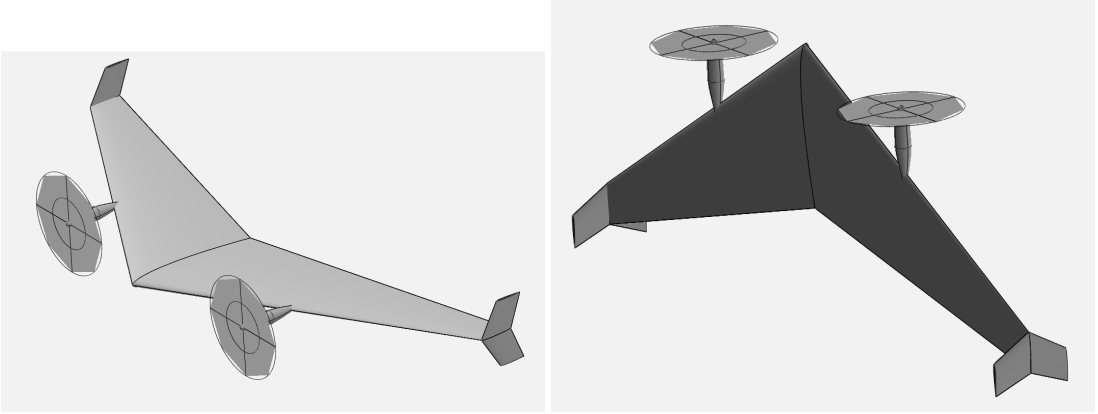


FIGURE 3.1: Sketches of a dual tilt-propeller tail-sitter.

### 3.1.2 Design Concept 2 - dual propeller tilt-wing

The dual propeller tilt-wing drone(figure 3.2) with a tail is much more stable than the tail-sitter in forward flight. Thanks to the tail and tilting wings this design would allow to speed down the forward flight to do some tasks, which would otherwise need hovering: e.g feature recognition for a payload delivery. However, it loses the simplicity of the geometry and is aerodynamically less optimal. It still has the same problem to tackle of finding an optimal between two flight modes.

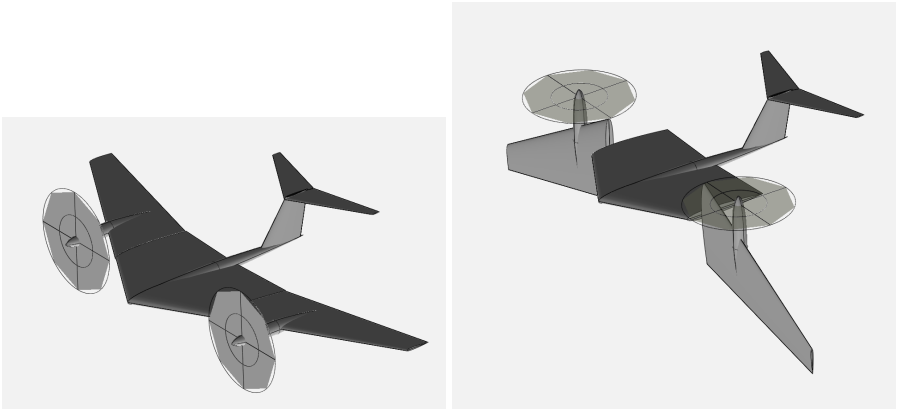


FIGURE 3.2: Sketches of a dual propeller tilt-wing.

### 3.1.3 Design Concept 3 - tilt-propeller flying wing tricopter

Tricopters are a unique group of multirotors as their tail-rotor sits on a servomotor, which tilts it in order to pan. This iteration uses a typical tricopter setup with two modifications (figure 3.3): firstly, a wing covers the otherwise bare frame of the tricopter, and secondly tilt mechanisms are added to both front and the tail-rotor to transition between forward flight and hover. The solution is much more stable in hover than dual propeller solutions, while it also decreases the difference between needs of the different flight mode propeller and motor dimensions.

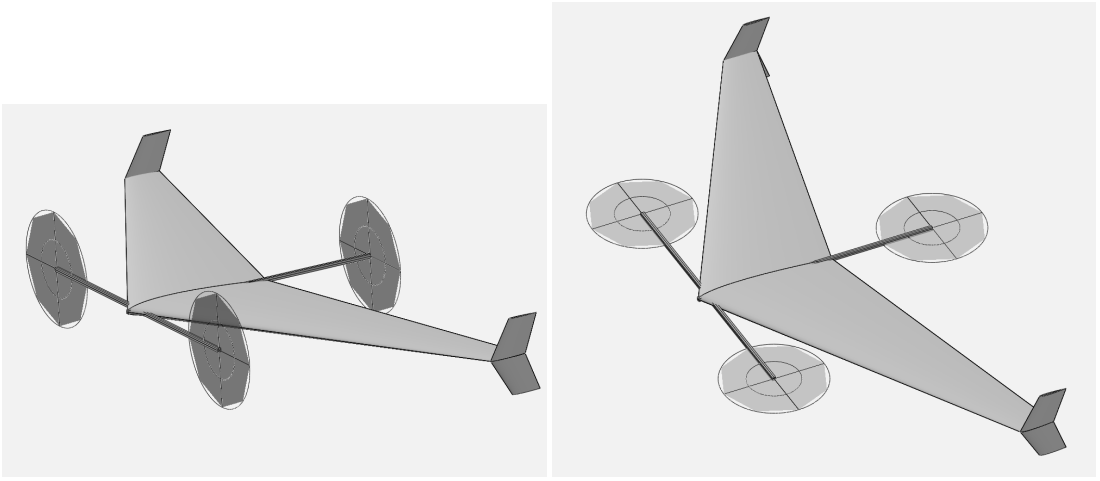


FIGURE 3.3: Sketches of a tilt-propeller flying wing tricopter.

### 3.1.4 Design Concept 4 - tilt-propeller flying wing quadcopter

Quadcopters are even more stable than tricopters in hover mode, however, the increase in electrical equipment onboard the drone does not pay off in terms of budget, nor in increase of payload. In fact, the structural build of the drone (figure 3.4) becomes bulkier and thus the wingspan might become an obstacle when finding a low cruise speed. The increase of hover time from three to four propellers is most likely trivial as preliminary estimations show an equivalent increase of gross weight.

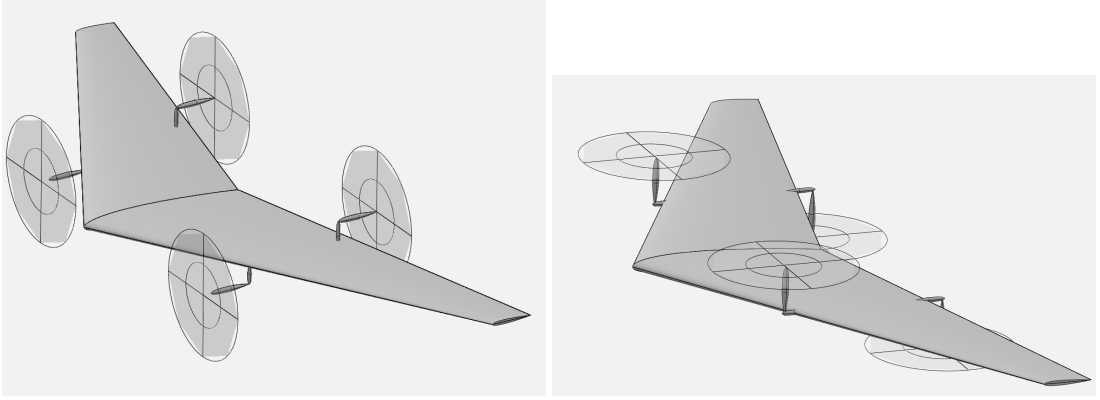


FIGURE 3.4: Sketches of a tilt-propeller flying wing quadcopter.

### 3.1.5 Final General Design Concept

After careful consideration the tilt propeller flying wing tricopter was chosen as the general design of the drone. As can be seen from table 3.2, the chosen solution is the only one estimated to meet all

conditions. The dual propeller systems have inferior hover time, which could be increased to an extent by adding batteries, but would also make hover less inherently even less stable. Also, DC2 build will become more expensive and difficult due to forces on the tilting mechanism. On the other hand adding a fourth rotor (DC4) would increase the size and weight of the drone due to structural and mechanisms needs. This version could potentially deliver longer hover time and longer range, but would also significantly increase the cost of the drone.

| | **DC1: Dual tilt-propeller tailsitter** | **DC2: Dual propeller tilt-wing** | **DC3: Tilt-propeller tricopter wing** | **DC4: Tilt-propeller quadcopter wing** |
|---|---|---|---|---|
| **Weight** | 1kg | 1.5kg | 1.5kg | 3kg |
| **Cost** | 80 € | 150 € | 100 € | 450 € |
| **Hover time** | 5-10 minutes | 5-10 minutes | 20 minutes | 30 minutes |
| **Hover stability** | Poor | Poor | Good | Best |
| **Range** | 20 km | 15 km | 30 km | 70 km |
| **Flight speed** | up to 150 km/h | up to 80 km/h | up to 150 km/h | up to 150 km/h |
| **Pitch stability** | Poor | Best | Good | Good |
| **"Dutch roll"** | Poor | Best | Good | Good |
| **Build simplicity** | Medium | Medium | Difficult | Difficult |
| **PID complexity** | Difficult | Difficult | Medium | Medium |
| **Ease of use** | Difficult | Difficult | Medium | Best |

TABLE 3.2: Comparison of General Design Concepts.

## 3.2 Propulsion

As three identical rotors are used, each rotor is required to generate a minimal thrust equal to a third of the weight of the drone:

$$T \geq \frac{1}{3}W \approx 5\,\text{N}$$

Roughly estimating a power load $PL = 0.1$ and considering that ideally the rotors will be equally loaded, the total power consumption is estimated to be

$$P \approx \sigma \frac{15\,\text{N}}{0.1} = 300\,\text{W},$$

where $\sigma$ is a safety margin to prevent heating of electronics devices such as motors, ESC-s and batteries. Based on preliminary estimations a 11.1V power system is chosen as when comparing batteries from retailers they have an optimal energy density and discharge rate suitable for this particular setup. When considering lower voltage batteries, the discharge rate might be lower than required by the system.

$$C_r = \frac{300\,\text{W}}{11.1\,\text{V}2.2\,\text{A}} = 12.3\,\frac{1}{\text{h}}$$

A rough estimation is also set for the motor constant velocity based on formula 2.14: between 1200 kV to 1500 kV. Two types of brushless outrunner motors were considered: disk and canister. The disk type motors have a flatter winding geometry and therefore should have higher torque. Based on comparisons of weight, price, maximum power load recommendation by the manufacturer and user

feedback at first the Turnigy L2210 1400kV disk type outrunner was chosen. As experimental results showed, the decision was changed due to technical difficulties with grub screws (discussed in subsection 4.1).

| Name | Type | kV | Battery | Mass | Power | Price | Thrust test* | User approval |
|---|---|---|---|---|---|---|---|---|
| Turnigy L2210 | Disk | 1400 | 7.4, 11.1V | 50g | 210W | 10.79$ | 8.8N - 8x4.5 - 150W | High |
| Turnigy Park450 | Canister | 1200 | 7.4, 11.1V | 66g | 175W | 20.14$ | 8N - 8x4.3 Prop - 150W | High, reliable, efficient |
| Turnigy D2822 | Canister | 1450 | 7.4, 11.1V | 38g | 160W | 9.98$ | 8.5N - 8x6 - 170W | High, efficiency issues |
| NTM 2826A | Canister | 1200 | 11.1, 14.8V | 58g | 215W | 15.59$ | 7N - 8x4.5 - 145W | Good, bearing issues |
| Turnigy D2826 | Canister | 1400 | 7.4, 11.1V | 50g | 205W | 11.33$ | 9N - 8x6 - 273.8W | Good, well balanced |

* Thrust test results from user feedback (Thrust - propeller dimensions - power).

TABLE 3.3: Comparison of brushless outrunner motors.

Air density at 15 °C sea level was used: $\rho_5 = 1.225\,\mathrm{kg/m^3}$ for hover calculations. The zero-lift drag for the propeller is assumed approximately $C_{d0} \approx 0.01$, but this can vary depending on the specific geometry. Figure 3.5 shows the power load versus thrust coefficient of a varying rotor diameter from 10 cm to 30 cm (calculated using formulas from subsection 2.1.3.1). The optimum is found to correspond to a diameter of 226 mm corresponding to a rotor solidity of $\sigma = 0.04$.

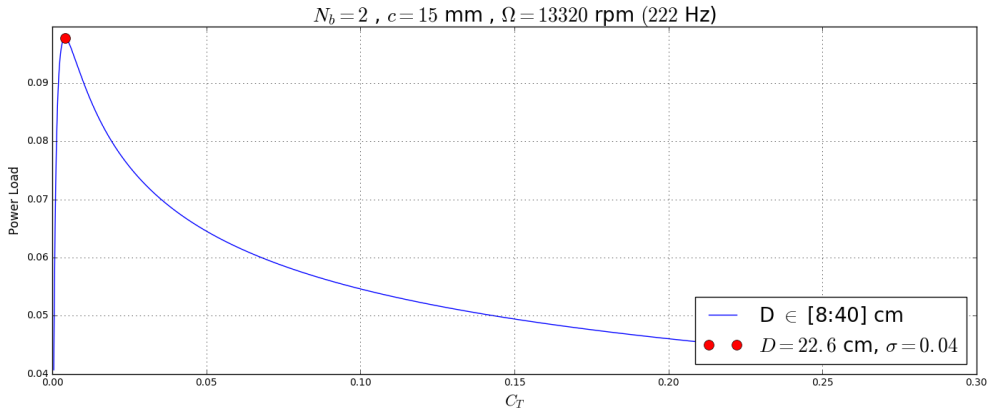



FIGURE 3.5: Power loading vs. $C_T$

On figure 3.6 the power demand of 3 rotors and available power is shown (camera, onboard computer and telemetry supply has already been subtracted). The power supply is sufficient for climbing and maneuvering. The weight of the drone increases by ∼1.8 N as a battery added. As the thrust is changed, it also alters $C_T$, $C_p$ and $P$ as noted in the equations. The relation between hover time and added batteries is shown in 3.7. It is clear that at least 3 batteries are needed to provide hover longer than the 20 minutes required.

The drag coefficient for the UAV central body, including exterior devices, is approximated as $C_{D,P} \approx 0.1$ with an estimated frontal area of $A = 0.2\mathrm{m}^2$. The wing drag is estimated with the lift-induced drag using formula 2.3 and the induced drag with the formula 2.4. Drag components are shown in 3.8. It is evident that in forward flight the least amount of energy is lost near 11 m/. A suitable cruise speed would be slightly above it to maintain stability of the aircraft in turbulence or other deviations. These results do not take into account the more complex setting of the rotors, which affects both the performance the wing and the propellers.

A 9'' (228.6 mm) diameter carbon fiber propeller with a 5'' (127 mm) pitch is chosen as it fits closest to the optimal propeller diameter. The medium pitch length would work well in faster forward flight speeds as well as hover climb rate.

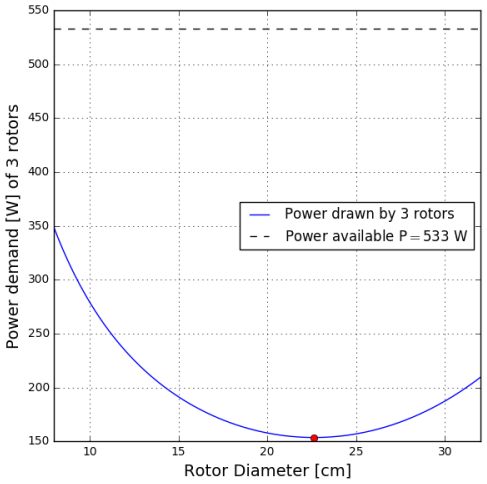



FIGURE 3.6: Power demand vs available power for hover at a $N_b = 2$, $c = 15$ mm

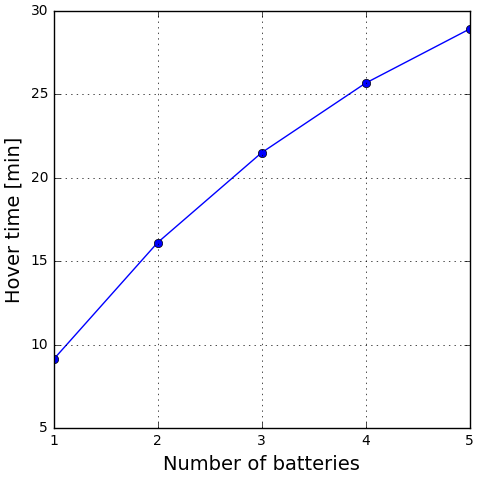



FIGURE 3.7: Expected hover time with 1 or more batteries.

As the UAV is a tilt-rotor tricopter, it can take advantage of hybrid flight modes, which are impossible for other kinds of drones: while cruising at speeds where the wing itself is alone not sufficient enough to maintain flight it can continue cruising getting partial lift force from the rotors. This is highly beneficial as at the same time the positive angle of the wing adds lift force from the forward flight mode as well. This possibility to extend flight time at low speeds is beneficial to many observational tasks, such as identifying objects with machine learning algorithms.

Using methods from section 2.1.3.2 an estimation is found how the range and flight time would alter if the wing would travel through the air at a fixed angle of attack and forward flight would only depend on the rotor changing its tilting position at a fixed angular speed.

Figure 3.9 illustrates how flight time and range estimates change over a vector of $\beta_r = 0°$ (vertical hover) to $\beta_r = 90°$ (horizontal forward flight mode) over airspeed velocity from $0\,\mathrm{m/s}$ to the slightly above optimal point of $12\,\mathrm{m/s}$.

The proposed preliminary design of the aircraft at the proposed cruise speed of $12\,\mathrm{m/s}$ gives good promise of flight time over 50 min and range over 30 km. Also hover time over 25 min is found to be likely with this setting. Therefore, based on these preliminary calculations, the chosen propulsion system alongside the general geometric concept of the drone fulfill requirements.

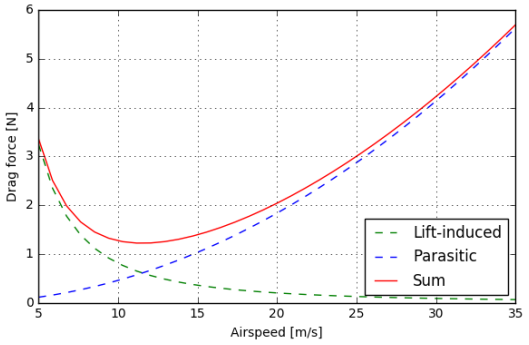



FIGURE 3.8: Drag force components over airspeed.

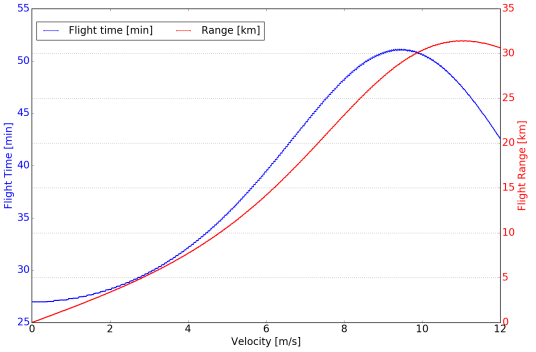



FIGURE 3.9: Flight time and range estimates .

## 3.3 Structure and Mechanisms

### 3.3.1 Structure

Research shows that very few flying wing tricopters are on the market and none have a tilting rear motor. Some hobby project blogs have revealed that the main issue with the rear tilting rotor is that it breaks often as it is far away from the main frame and has to face downwards in order to be usable for forward flight. Another issue commonly complained about commercial wings is the vibration carrying over the booms to the navigation system.

As discussed in subsection 2.1.2, a fixed wing forward flight aircraft that does not have a tail requires a carefully positioned $CG$ up front from the $AC$. The recommended static margin $SM$ for flying wings is approximately 10-15%. Tricopters, on the other hand, prefer the $CG$ to be positioned at equal distance from each of the rotors centers (as shown on figure 3.10). Therefore, a compromise in the design must be made in order to secure stability.

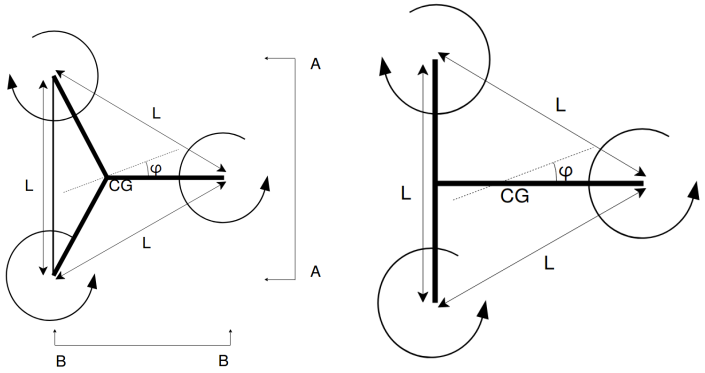



FIGURE 3.10: Typical structural configurations of tricopters.

All mentioned structural aspects have been taken into account developing the rough estimate of a reliable structure (figure 3.11). The main hull material is selected extruded polystyrene (XPS) as it is very light ($\rho =\sim 20\,\mathrm{kg/m^3}$), easy to process, cheap and has been proven by the industry to perform well. If later experiments reveal the material is not strong enough, it can be covered with fiber glass. Detailed drawings are provided in the appendix for manufacturing the frame and mechanisms part of the project.

The traditional Y-frame tricopter has three rods symmetrically joint in the $CG$ with a 120° angle between them. This concept has proven to be very stable and controllable, which is why most autopilot software has been designed for this circumstance. The alternative T-frame has the same distances between the rotors and $CG$ and therefore is only structurally different. The Y-frame is used as it is sturdier and the front rotors are connected with a thin round rod to ensure simultaneous tilting.

A typical distance between rotors is $L = 600\,\mathrm{mm}$, which has been chosen for this project as well to make autonomous flight code writing easier. The distance between rotors is near optimal as rotors are more efficient closer together in forward flight, but the back rotor should not be influenced by the backwash of the front propellers. Therefore, as the clearance between the front rotors is 372 mm and the diameter of the rotor is 228 mm the placement is considered optimal.

Using formula 2.10, a rough optimal geometry of the flying wing is found (figure 3.12). This approximation satisfies the tricopter central $CG$ location and the static margin requirement of $AC - GC$ = 5%...15% for longitudinal stability (respectively (225.5 mm...195.6 mm) from the aircraft nose). A blunt nose is designed as it makes the tilt-rotor system structurally secure and enables to accommodate batteries and payload up front to better regulate the $CG$ position.

A 1 mm aluminum main frame is glued to the inside of the UAV (figure 3.11)). Bends have been exerted to make it more rigid and the inner filled area has been optimized for minimal weight. The most

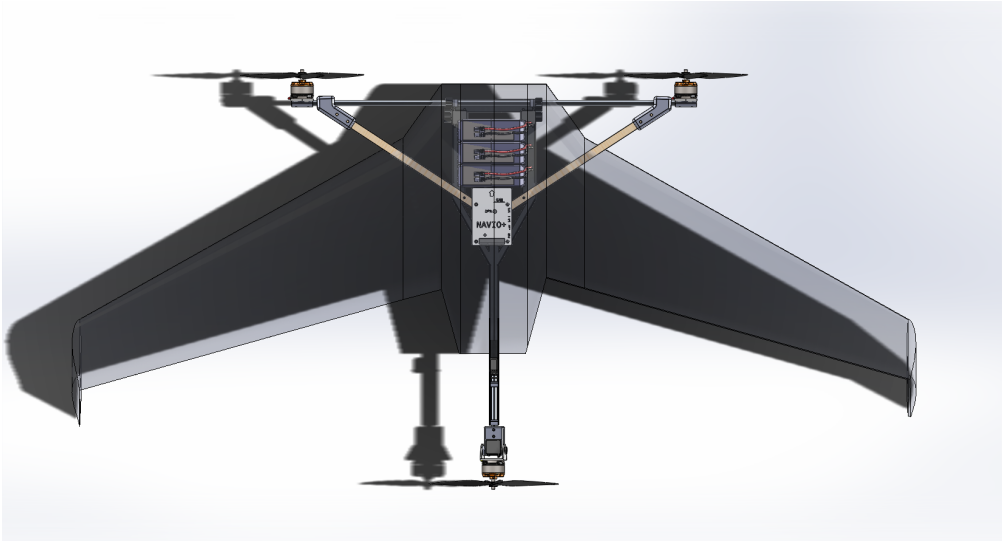


FIGURE 3.11: The decision space of optimization

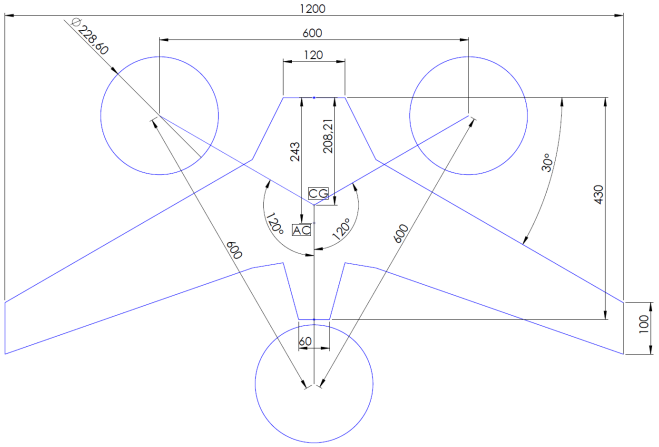



FIGURE 3.12: Preliminary planform of the geometry before optimization.

vulnerable area of the frame is the long canal reaching to the aft rotor. As seen on figure 3.14 stress concentrated near the sharp cuts were ca 30% higher. If the distance was increased the directional deformation at the rotor position started increasing considerably.

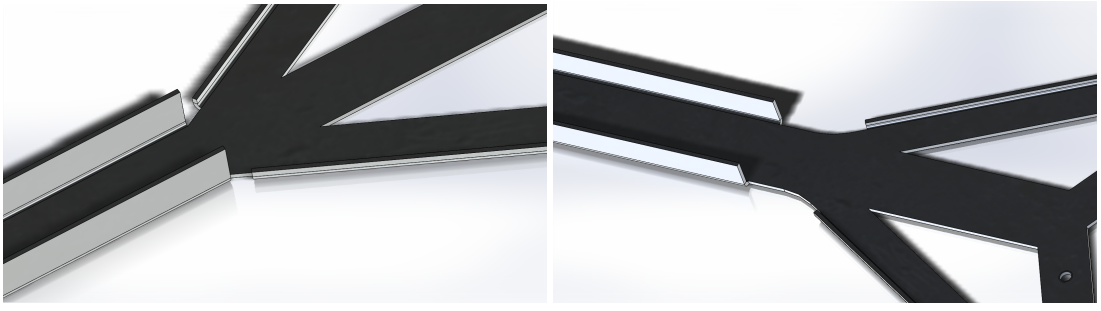


FIGURE 3.13: Main frame before (left) and after (right) modification.

Two pine booms with a cross section of 12 mmx12 mm are bolted to the frame. Wood is preferred over carbon fiber rods as they dampen the vibration. Rubber bushings support the navigation system casing to further dampen the vibration effects. For optimal $CG$ batteries are free to be placed within the

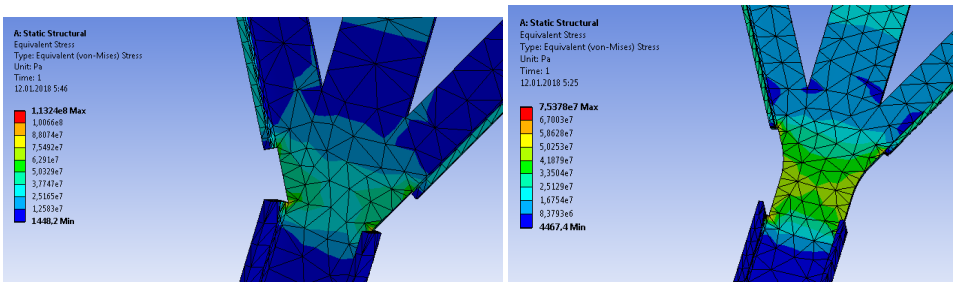



FIGURE 3.14: Stresses on the frame before (left) and after (right) modification.

blunt nose container. Winglets point downwards and act as supports during landing. For a third support point a wheel is unfolded from inside the blunt nose on landing.

### 3.3.2 Tilting Mechanisms

Front propellers are mounted on an 8mm aluminum rod, which is turned by two servo motors. The rod is supported by bearings in four locations: near both servos and in casings attached to booms (figure3.15 ).

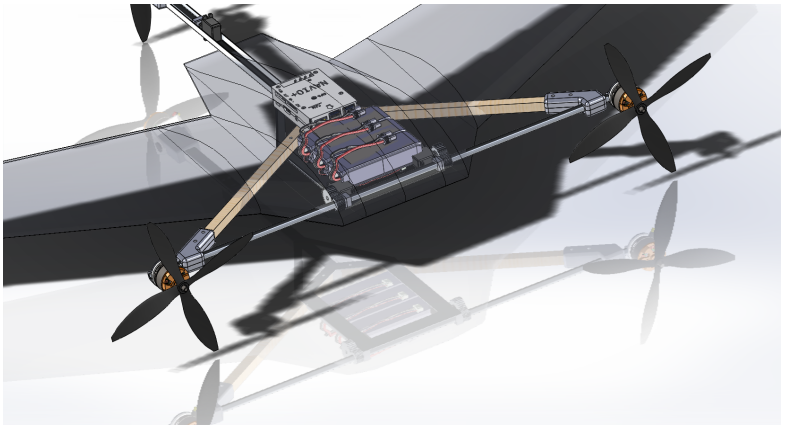


FIGURE 3.15: Front rotor tilting mechanism.

After the $CG$ towards the aft the aluminum main frame turns into a C-channel accommodating a servor motor connected to an 8mm aluminum rod for the hover mode tilting (motion sketched on figure 3.17). This motion is necessary for the tricopter to stay stable and acts simlarly to a helicopters tail rotor.

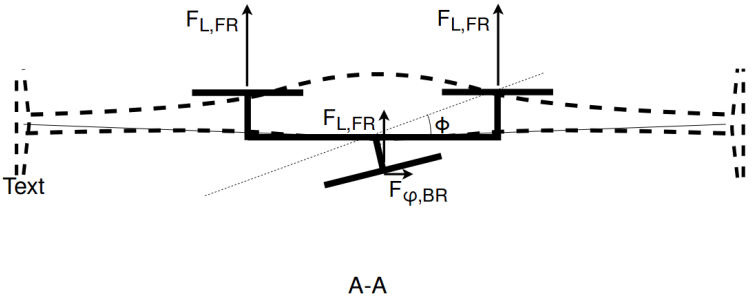



FIGURE 3.16: The front rotor tilting mechanism force components diagram

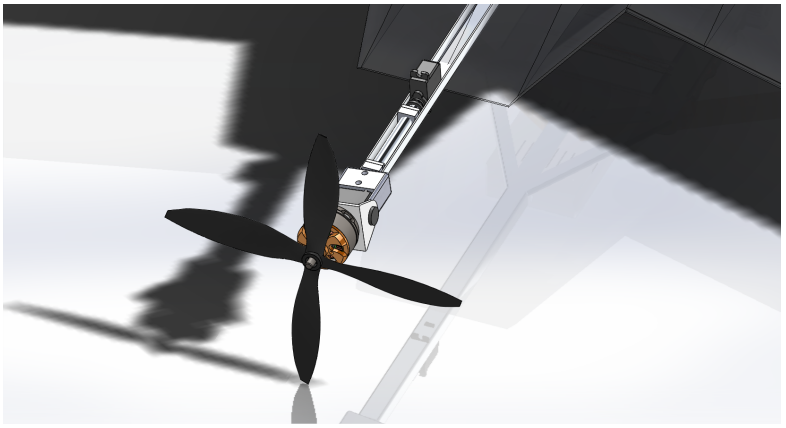


FIGURE 3.17: The aft rotor tilting mechanism.

## 3.4 Aerodynamic optimization setup

Flying wing configuration aircraft have high potential to outperform conventional airplanes. They provide many design perks: eliminating the tail reduces energy loss and instead of a long high drag coefficient fuselage the aircraft have an efficient space in the middle, which acts as a wing.

However, to optimize the shape hundreds of variables could be tuned, which are analytically difficult to solve and iterative simulations are CPUh costly. This curse of dimentionality can be tackled with effective Bayesian MOO optimizers. During research Predictive Entropy Search based optimizers were not found to have been used in aerodynamic optimization cases.

#### 3.4.0.1 Physics and geometry setup

The physics environment of the optimization task is set standard sea level (SSL) conditions: air at $15°$C (288.15 K), 1 atm (101 325 Pa), air density $1.225kg/m^3$ and air viscosity $1.789 * 10^{-5}Pa * s$.

Within this aerodynamic optimization assignment the Bayesian optimization MOO with a PESM acquisition function was handed a set of 15 constrained variables. Eight of the variables with constraint limits are shown on figure 3.18. The other 9 variables were airfoil variables for the NACA 6-series airfoils. These airfoils have very nice qualities and are at the same time more flexible than the NACA 4-series airfoils. Airfoil geometric parameters were assigned in three sections, where the wings changed the shape most: from centerline to the main wing, the main wing and the winglet. The NACA 6-series airfoils have the capability to both form reflex on the back of the wing providing good stability for the flying wing. On the other hand it is flexible enough to create high lift wings with camber in the aft.

Kulfan CST and Bezier curves were also considered as they are potentially more versatile.

Most of the constraints are due to the rotor placement and the maximum wingspan. Some were set to prevent the physics solvers from producing unrealistic (e.g the flat plat problem discussed in the physics solvers section) or NaN answers.

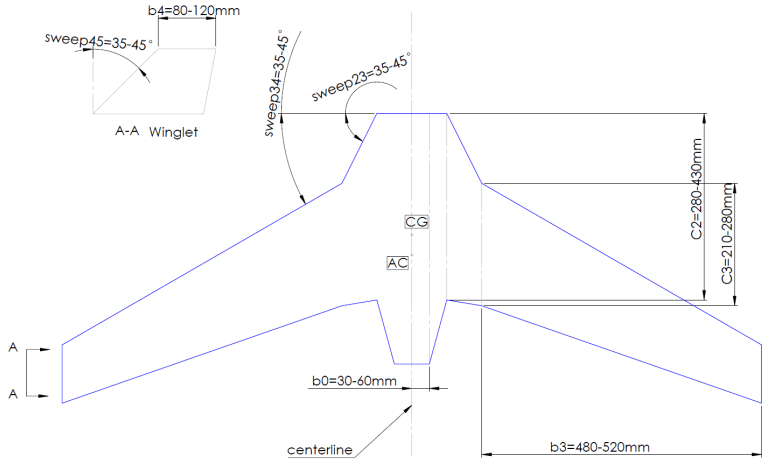



FIGURE 3.18: The optimization decision space.

The optimization task focused on minimizing energy consumption and maximize stability of the aircraft at 12 m/s - the cruising speed selected in previous sections. The objective space variables were $C_D$, $C_{My\alpha}$ and $C_{Mx\beta}$.

To analytically connect the decision space variables to the objective space variable analytically is close to impossible. Seasoned aerodynamics and CFD experts consulted at the Technical University of Denmark relied more on learned experience and rule of thumb laws than direct formulations. Many of

them gave examples of how sometimes good solutions are found by mistake in the aerodynamics domain. This might be the case of the Prandtls bell shaped lift distribution, which migt have been ahead of its time due to the structural engineering limitations of that time period.

Therefore, the hypothesis is that the Bayesian based optimization algorithm develops am intuition. One of the authors of the PESM approach describes the algorithms method of finding the next test point as ”hallucinating” for lack of a better word.

As the optimization task provided is much more difficult to solve for humans, another non-convex optimization task was set up to benchmark the MOO to a human engineer. 16 loosely constrained parameters and a similar objective space to the current task were proposed. After 50 iterations both PESM and the human engineer converged near the same geometry as shown on figures 3.19 and 3.20 respectively.

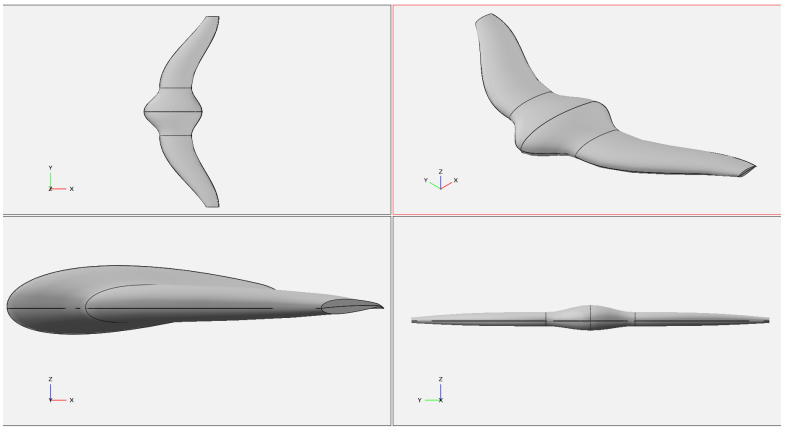


FIGURE 3.19: PESM converged solution.

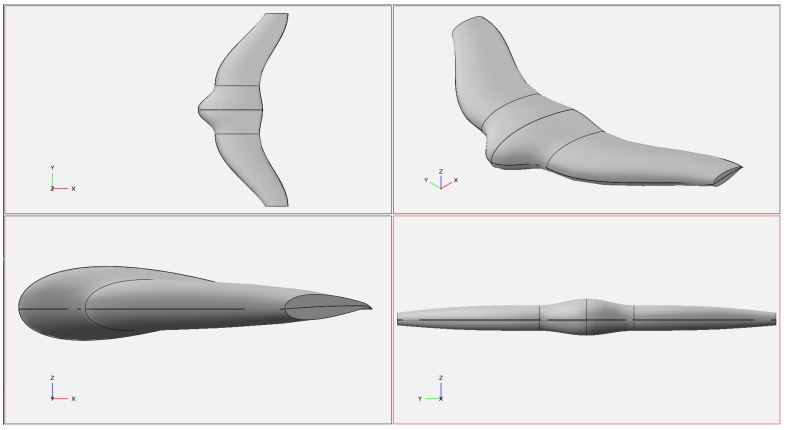


FIGURE 3.20: Human converged solution.

### 3.4.1 Implementation

For the purpose of using this MOO software a special configuration of Linux Ubuntu 14.04 LTS was set up to satisfy all the dependencies of the physics solvers and the optimizers. Pyhton was used as the main scripting language. Code written within the scope of this thesis are handed in separately as meta data.

Error handling of physics solvers became crucial as the optimizers get influenced by every objective space result and indirectly by not retrieving a result as well. Physics solvers can give exponentially inaccurate results if input geometry has bad quality or physics environment is not suitable for that

solver. The geometry solvers also crashed due to some algorithmic bugs. To tackle these problems some test points decision space values were altered by one tenth of a millimeter in order to bypass this technical problem. Also, thresholds were set to rate ridiculous values as low results. This is not an ideal solution as it does not reflect real-life physics. It is the best however to overcome the issue temporarily.

# 4 Results & Validation

In this chapter the validation of the methods used and results are discussed:

- The propulsion system estimates were tested with a thrust measuring stand and a constrained tricopter mock-up.
- The VLM solver OpenVSP was validated with thin plate experiments in regions between $\alpha$ = ±5 deg and a validated CFD analysis of a S5010 airfoil based wing.
- Ansys Fluent CFD solver was validated at $\alpha$ = 0...10 deg with the thin plate experiment.
- Results of the optimization are reviewed with Ansys Fluent.

## 4.1 Propulsion validation experiments

Testing the Turnigy L2210 using a thrust measuring stand (figure 4.1) confirmed the thrust and power given by users, but also revealed a design flaw (figure 4.2). The grub screw tightening the winding to the fixed inner shaft had insufficient grip or vibrated loose. A workaround would have been to add grub screws and fix it with thread-locking glue. However, to minimize risks another motor (Turnigy Park450 1200kV) was chosen instead, as it had a lower kV rate and was deemed more reliable by users. The tests confirmed the prediction of an average 150W power draw when used a 8''x5'' carbon prop between 7.5 N and 8 N.

FIGURE 4.1: Thrust measuring stand.

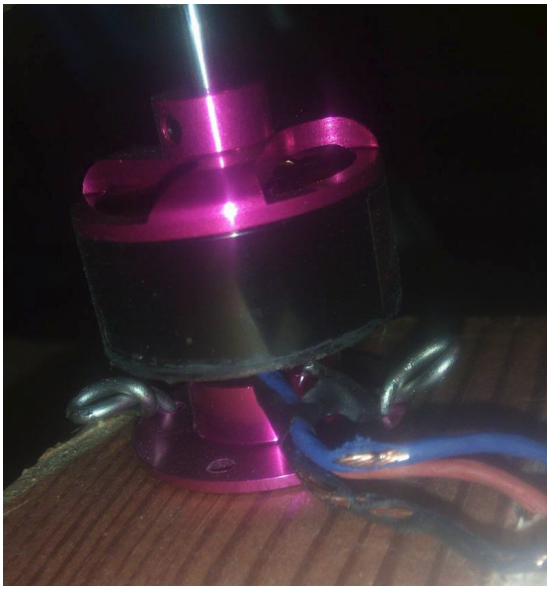


FIGURE 4.2: Broken wires due to grub screw failure.

For the second test a frame was composed of 11 mmx50 mm pine board, which weighed 1.5kg with added weights (figure 4.3). It was loosely attached to a table stand with long bolts and kept in hover

mode for an avearge of 7.5 minutes using one 2200 mAh 35C LiPo battery. After that the battery was critically low and had to be charged. Reflecting on this, the 15% shorter hover time might be due to poor hover control when attached to the stand and peculiarities of LiPo batteries as they cannot be discharged to less than having 3.0 volts per cell.

FIGURE 4.3: Tricopter test stand.

## 4.2 VLM and CFD validation

**Validation cases**

Flat plate experiment AR = 3, Q = 20 m/s, Re = 80 000 SG5010 rectangular wing validated analysis AR = 3, Q = 20 m/s, Re = 232 000

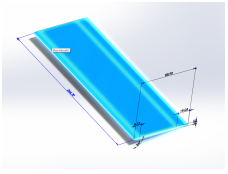


FIGURE 4.4: Flat plate geometry.

Figures/Tests/SG5010.png

FIGURE 4.5: SG5010 rectangular wing geometry.

**Validation results**

The VLM solver OpenVSP and CFD CVM solver Ansys Fluent physics environments were benchmarked. As can be seen from figure 4.6, between $-5°$ and $5°$ the VLM results are very similar to the experimental results. However, when leaving that zone the $C_L$ values stay on course, whereas $C_D$ values drift off. This can be attributed to the fact that the lifting line theory does not take into consideration the growing frontal area of the wing causing drag and an air bubble on top of the wing. Furthermore, the as the angle of attack increases dramatically flow rotation intensifies as well, which makes the flow even more complex. Fortunately this should not affect the results of the optimizer as they are mostly related to very thin wings and secondly most preferred $C_L$ values of the aircraft are expected to lie between $\alpha =$ .

Ansys Fluent was validated between $\alpha =$ to be within 5% of the experimental value. Therefore, to a reasonable degree of certainty OpenVSP results can be validated and further examined using Ansys Fluent.

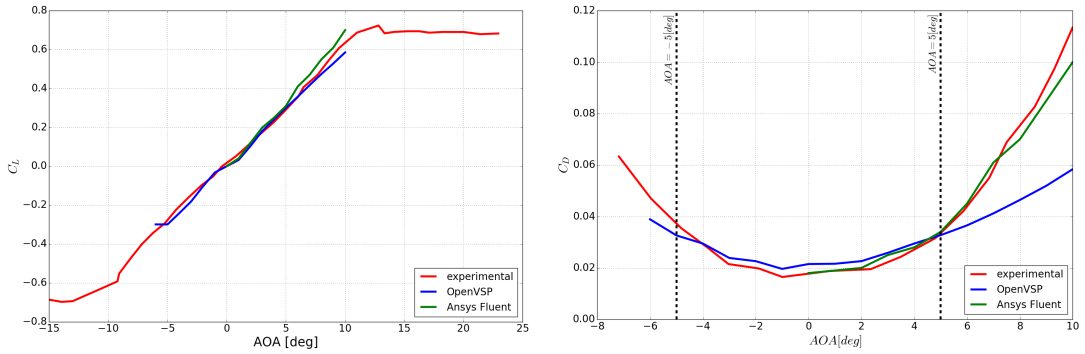



FIGURE 4.6: Flat plate validation results.

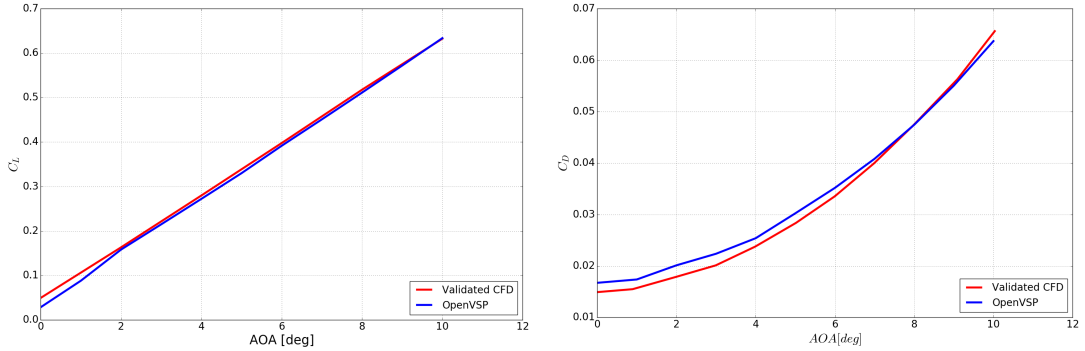



FIGURE 4.7: SG5010 rectangular wing validation results.

## 4.3 Analysis of optimal result

Comparing the B-MOO solvers results it is evident that for this particular run PESM achieved some of the closest values to the underlying real Pareto optimal solution. This was not a persistent trend: in some test tasks e.g SUR performed superior to PESM, so did EHI. At the same time, SUR and EHI were more prone to crashing and had to be restarted many times. This was the case during this assignment as well.

ParEGO and SMSego, as was discussed in section 2.3.2 performed on average slightly worse, but more persistently drew out the Pareto front for some reason (see figure 4.8 the SMSEgo and ParEGO patterns in comparison to a hectic SUR ). In essence this is the difference between an exploitation (SMSego) vs exploration (SUR) type acquisition function characters.

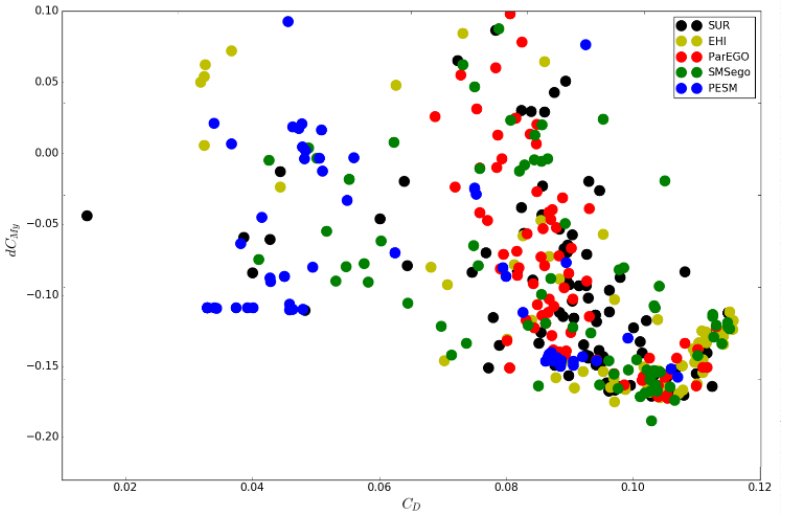



FIGURE 4.8: Comparison of B-MOO optimizers in the $dC_{M_y}$, $C_D$ objective space.

A Pareto optimal point from the results of all acquisition functions was selected (figure 4.9). According to VLM OpenVSP this optimal solution has the following properties: $C_D = 0.036$, $C_{M_y,\beta} = -0.12$ and $C_{M_x,\alpha} = 0.04$.

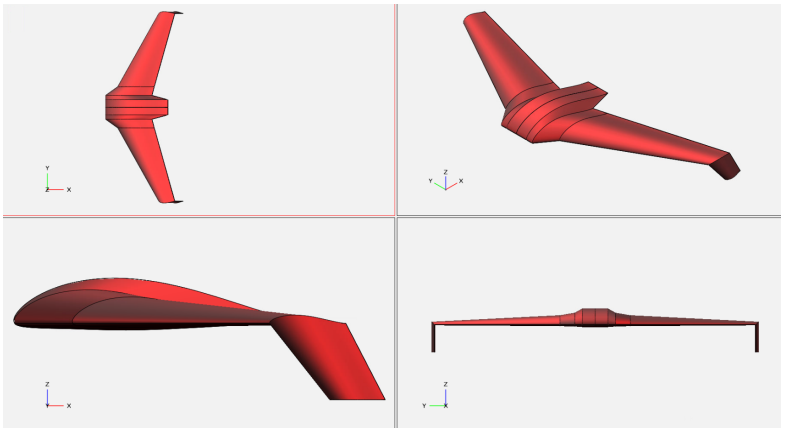


FIGURE 4.9: The selected Pareto optimal geometry.

The lift distribution (figure 4.10) resembles roughly an elliptical form. As the rotors will increase the airflow over the wing the higher lift near the nose will help maintain the smoothness and decrease the potential rotation of flow if otherwise.

### 4.3.1 Analysis of optimal result

[Add mesh pictures and explanations here]

Converged surprisingly well thanks to very low Re. Also, new Ansys Fluent meshes more efficiently. Figure 4.12

Pressure plot looks nice and smooth - this is good for stability, max lift and structure as well. The pressure at nose is not very high. Figure 4.13

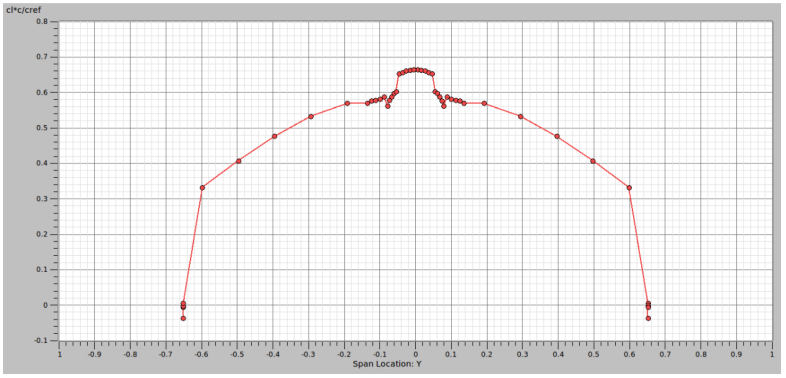



FIGURE 4.10: Lift distribution over the span of the Pareto optimal solution.

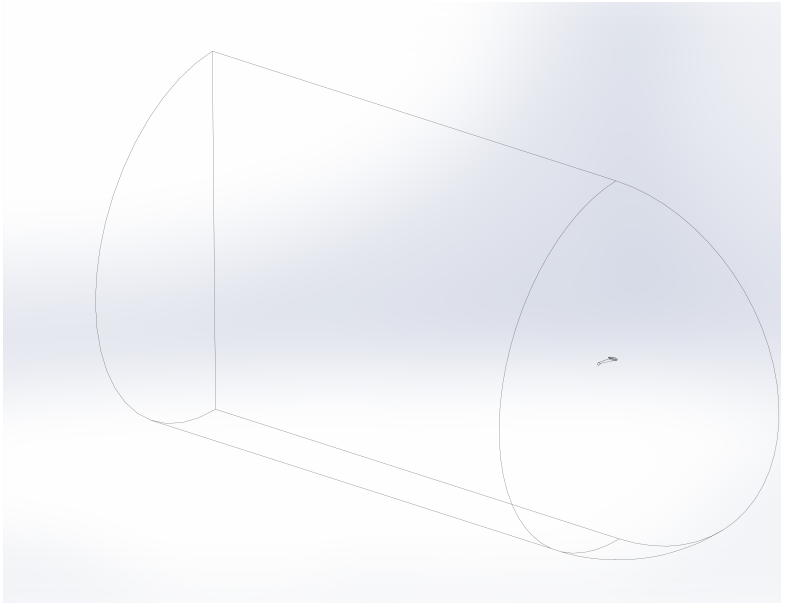


FIGURE 4.11: Boundary conditions for the Ansys Fluent CFD analysis.


FIGURE 4.12: Convergence of the CFD simulation.

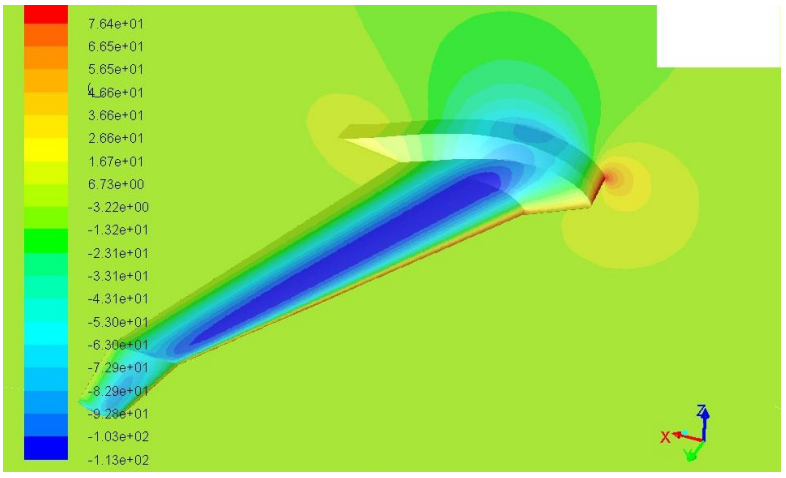



FIGURE 4.13: Pressure plot of the CFD simulation

## 4.4 Final assembly of drone

FIGURE 4.14: Pressure plot

FIGURE 4.15: Pressure plot

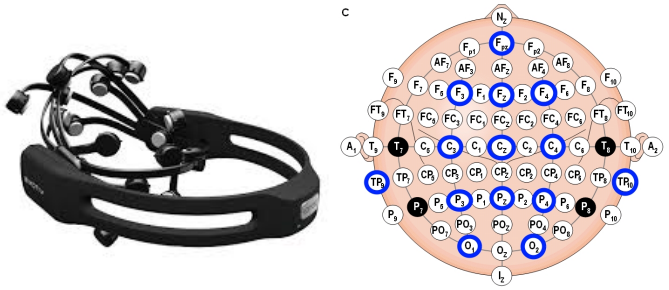
# 5 Brain-Computer Interface

This chapter contains the methods, design and validation of the nice-to-have requirement of the product: a brain-computer interface to communicate with the drone. The goal of training a policy to distinguish motor movement was accomplished.

Brain Computer Interface (BCI) constitutes a mean of communication between a human brain and a computer using the recording of brain signals. BCI technology is highly probable to integrate to our daily life within the next decade. According to Gartner Hype Cycle for Emerging Technologies 2017 the plateau of productivity will not be reached in less than ten years. In combination of highly sophisticated machine learning and signal processing algorithms and rapidly evolving electronics there is an ocean of neurons waiting to be discovered.

Drone pilots during missions (e.g surveillance, rescue, conflict situations etc) have a complex task to resolve: while having hands filled with controllers they need to assign other commands besides maneuvering. As the motion of the drone using FPV could be inherently similar to the intuition of moving ones limbs and creating similar patterns in the brain signals. One study revealed even that expert action video game players had a more developed sensorimotor network. Therefore it is likely that future unmanned vehicle pilots conducting critical missions will be connected with an integrated BCI system.

Work described in this chapter assesses the concept to control a drone in a three class system (move left, no movement and move right) using real and motor imagery data collected with a modified consumer EEG headset (figure **??**). The EMOTIV EPOC+ is the property of the Department of Applied Mathematics and Computer Science at Technical University of Denmark. The 14 electrodes and the amplifier had been detached from the plastic headset and placed in an EEG cap according to the layout shown in figure **??**. The signal from those electrodes goes to the EMOTIV EPOC+ wireless amplifier, which sends the data over Bluetooth to the laptop. On the drone there is a Raspberry Pi computer directly connected to the navigation and control system of the drone and over Mavlink to the laptop. This does not only provide autonomous flight capabilities to the drone, but also a fills the loop of the BCI drone system.



(A) .

(B) New placement of electrodes indicated by blue circles.

FIGURE 5.1: .

BCI systems consist of five main steps in a closed loop (figure 5.2)[16]:

- acquisition of brain signals

- pre-processing raw data
- extraction of features
- classification
- application
- feedback

BCI systems using EEG signals are currently favored in many applications due to good temporal resolution, easy portability, low-cost and non-invasive nature. Limitations, such as low spatial resolution of EEG signals and an uncomfortable cap/device, still need to be overcome [1]. Amplifier-integrated probe electrodes have proven successful to retrieve high quality $\alpha$ and $\beta$ wave signals with no physical contact to the scalp (at distances of ca 1-5mm from the skin). Using similar methods heartbeats have been recorded at a 1m distance and there is evidence that with better spacial analysis techniques EEG brainwaves could be mapped at a distance or with an inconspicuous device mounted on the head.

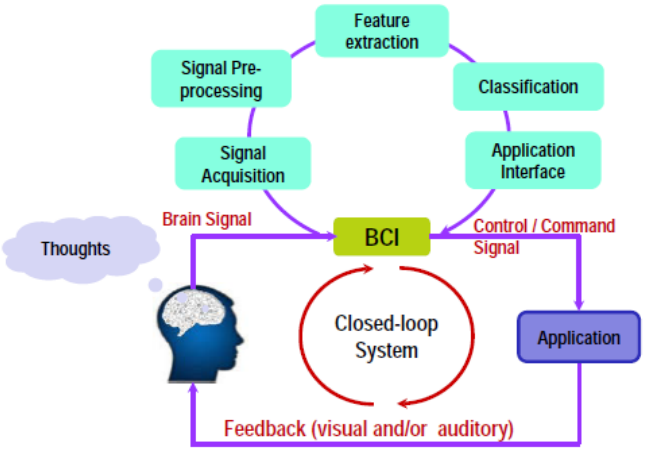


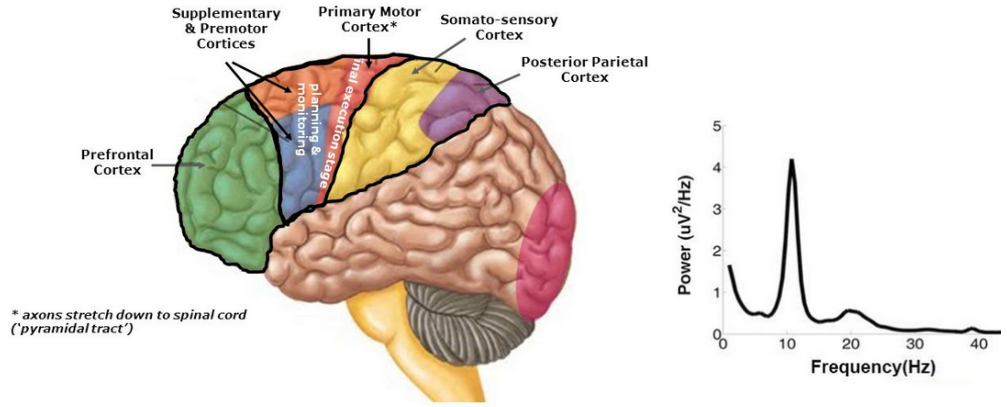

FIGURE 5.2: BCI system diagram of the main steps [1].

Sensorimotor brain activity within 7.5 – 12.5 Hz ($\mu$ rhythms) is most prominent in the sensorimotor cortex (figure **??**) [17]. A fraction of a second before actual motor movement, there is increased in potentials in the premotor cortices. Data from this region connected to a command system would therefore give faster reflexes to pilots. $\mu$ rythms are also triggered even when the movement is planned but not executed. Although these rhythms It has been experienced that sensorimotor rhythms present power decrease in the low frequency components, or event-related desynchronization (ERD), and an amplitude increase in high frequency components, or event-related synchronization (ERS) [17].

## 5.1 Data recordings

As mentioned, data was received from 14 electrodes as shown on figure **??**. The subjects sat comfortably behind a desk in front of a laptop, where commands would show up on the screen to either actively hold an open hand or fist or relax (figure 5.4). Therefore five brain states were recorded. Pictures were chosen to be aesthetically pleasing and not too bright to ease the recording process and help subject blink less during the recording. Also, the room was kept silent and dim.

The picture commands came up in a random sequence and data was labeled accordingly within a Python script, which received channel signals over EMOTIV EPOC+ Bluetooth transmitter. The commercial headset uses saline water for better conductivity, whereas the modified EEG-cap used a viscous saline gel. The process of getting subjects mentally ready for recording and having good conductivity was crucial as focusing on actively forming a gesture was demanding for subjects. To




(A) Sensorimotor related functional areas in the brain anatomy.

(B) $\mu$ frequency response of planned movement of limbs [**?** ].

FIGURE 5.3: .

clarify: actively forming a gesture does not imply squeezing, but rather mindfully making a gesture. The EEG signals were first referenced to the baseline between sessions: eyes open, blank dark screen on the laptop. A maximum of 4-5 five minute sessions were recorded as the process is tiring. Six subjects data was recorded as summarized in the results subsection.

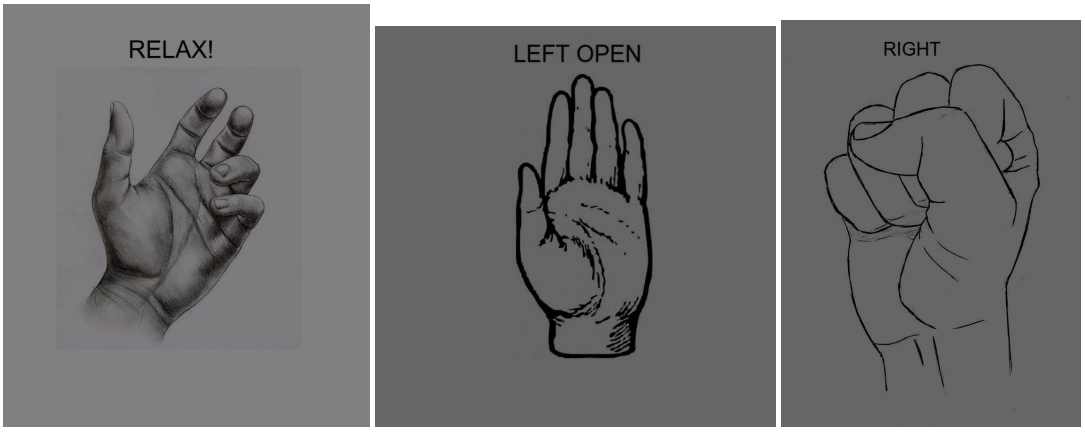



FIGURE 5.4: .

## 5.2 Signal processing and classification methods

The raw data was first pre-processed (cleaned), transformed before classification algorithms were trained. For the first two tasks were a Python EEG analysis and visualization libraray, MNE was used**??**. For classification machine learning algorithms from Sklearn were used**??**.

First, the baseline was subtracted (eyes open recording) from each channel and ”noisy” electrodes (electrodes that had transmitted unrealistic values) were removed from data. To further reduce noise in the data a bandpass filter (Butterworth) was applied to separate the 11-30 Hz band, which involves $\mu$ and $\beta$ waves. This also eliminates external signals coming from power lines (50 Hz) and frequencies from electronics, the human body and the Earth (Schumann resonance). Finally, ICA (Independent Component Analysis) was applied for artifacts removal. ICA is based on the assumption that the multi-channel data is constituted by linear independent basis (or components) in temporal domain. This hypothesis allows to extract components such as eye blinking and muscle artifacts.

The data was transformed from the time/potential domain into the frequency domain using Fast Fourier Transformation. As data was scarce (especially in the beginning of the project), the FFT function moved over the data set one time step at a time taking 256 data points sequentially. Classifiers were then trained and cross-validated to distinguish right-hand movement from no-movement (resting state) and left-hand movement. The following five classifiers were considered: Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Logistic Regression (LR), Multilayer Perceptron (MLP) and Gradient Boosting Classifier (GBC). Their pros and cons are further discussed under the classifier selection subsection.

### 5.2.1 Classifier selection

**Linear Discriminant Analysis (LDA)**

LDA is a much-used method for EEG-based BCI classification tasks. To distinguish between classes it predicts two normal density functions by applying least squares estimation. This method is therefore computationally effective and robust. LDA does not work well with non-linear separation and is sensitive to outliers because of its method of projecting data points.

**Logistic Regression (LR)**

LR is widely used to predict cancer, given patient general and habit data. LR provides the probability of a binary output and has similar properties to LDA, but instead of using density functions to distinguish between classes, it uses using log-odd functions. LR is more robust than LDA and usually LDA gets a worse result. LR performs very well with problems with not too many features can give very good predictions to a data set with a binary output and the data is linearly separable. LR is robust, fast and can be regularized to generalize well for new data. However, it has many disadvantages when there are many features and mixed correlations between them; continuous systems and if data is not linearly separable. As output is binary this method might work well, but depending on feature selection there might be problems. For further accuracy, Logistic Regression maybe not the best choice here, but it has good potential to be a weak learner.

**Support Vector Machines (SVMs)**

SVMs are widely used for EEG-based BCI classification problems, disease diagnosis and other problems which are difficult to separate linearly. SVMs allow the 'kernel trick' to separate linearly inseparable data, meaning that it should get a better accuracy when properly tuned than LDA and LR. SVMs are sometimes referred to as a black box as the separating vectors are complex. They are also computationally quite costly and demand more training data when compared to other algorithms. Therefore, this may be a limiting factor for our project. SVM is less robust than LR, which may cause overfitting.

**Gradient Boosting Classifier (GBC)**

Gradient boosting is gaining popularity on Kaggle Machine Learning competition platform as it has proved to be a very efficient way to continuously improve a model, while being computationally light and less volunerable to overfitting. Most notably this method won the Netflix competition[18]. GBC puts more emphasis on harder to classify data, while staying robust. Partly because of this more iterations doesn't make the model overfit. Using multiple weak learners makes it work well with non-linearly separable data. However, it is computationally more costly than LDA and LR, but not merely as costly as SVM-s. The data will be hard to connect to the model results to interpret, because of many weak learners. Many learners and their combinations also create more hyperparameters for tuning.

**Multilayer Perceptron (MLP)**

MLP is considered a form of Deep Learning as it takes advantage of the similar effect of hidden perceptrons. The same concept of LR can be expanded to a feedforward neural nets perceptrons by transforming the value with a sigmoid function. This is a very beneficial method of making it possible to solve very complex problems. Although it is widely used, it has many disadvantages. Similarly to GBC, the model is hard to interpret through logical assumptions of features and labels. It is also computationally rather costly, uses many parameters and has a tendency to overfit. However, as it is difficult to extract features from the 14 input channels while capturing still over 90% of the variance, this might be considered as a good solution.

#### 5.2.1.1 Workflow and metrics

First data is split into separate two groups training (80%) and testing (20%) and are treated as a three class problem distinguishing between left and right hand movement and relaxation. The data can not be shuffled and data from the same person from the same recording day should be used as brains tend to shift their patterns overnight. A Naive modelling method is used for a reality check predicting the biggest class label:

$$NaivePredictor = \frac{tp}{AllData},$$

After the classification models are compared, one model is chosen to be further tuned with an exhaustive grid search. Sklearns GridSearchCV function is used to evaluates all proposed combinations of selected hyperparameters [19]. All results were evaluated using K-fold cross-validation with $F_\beta - score$ metric [20]. $F_\beta$ score takes values between 1 (being the best precision score) and 0 (being the worst) and can be defined as

$$precision = \frac{tp}{tp + fp}, \qquad recall = \frac{tp}{tp + fn}, \tag{5.1}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}, \tag{5.2}$$

where $tp$ (true positive) = ”correct result”, $fp$ (false positive) = ”unexpected result”, $fn$ (false negative): missing result and $tn$ (true negative) correct absence of result.

To finally test the system the classifier should be trained during the recording session and validated with an empirical test right away. For this a virtual environment was created in Python, where a box would assume the direction of movement depending on which limb moved or was imagined moving (figure 5.5).

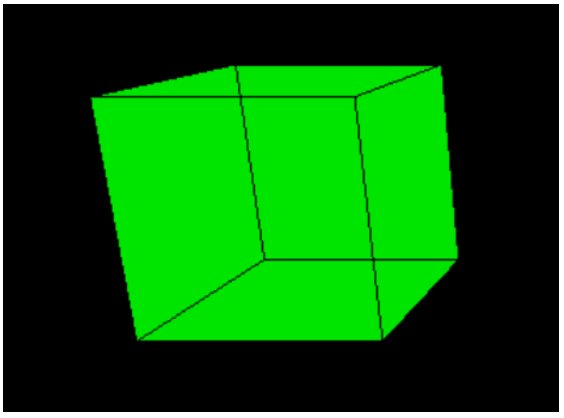


FIGURE 5.5: BCI system diagram of the main steps [1].

### 5.2.2 BCI task results and discussion

After initial classification trials were done on the preliminary recorded data, it was desided that Gradient Boosting Classifier would be the most prominent to be used. The three best classifier types are shown on figure 5.6. It was a hard decision as the graph only shows the weighed average results. For some subjects data one classifier worked better than for others, for another subject it was vice versa. However, the other classifiers (LDA and SVM) scored steadily less than others. We can see that the MLP and GBC require much more computational resources than LR. When we look at how much faster their $F_{beta}$ scores decrease as the training portion grows we can make a safe assumption that LR is much less prone to overfit and is more robust. However, only GBC is close to the naive classifier (dashed line).

GBC was decided to be used not only because it is the medium between the two extremes, but mostly because of its architecture. It can be assumed that fine-tuning the hyperparameters, the weak classifiers will generalize better and has the potential to score higher precision.

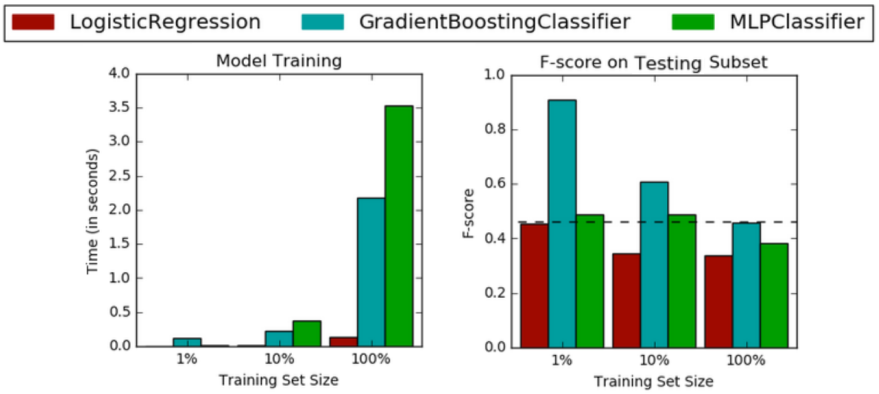



FIGURE 5.6: .

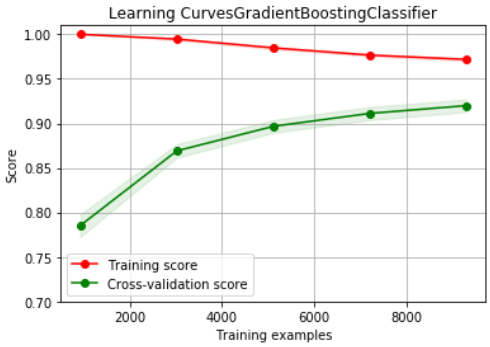
A grid search was conducted and training the classifier extensively during a session, it yielded actual control function. Once all the data was recorded and the classifier sequentially tested and cross-validated over each session, it achieved a consistently good score for both real motor movement and imaginery motor movement (figure **??**).

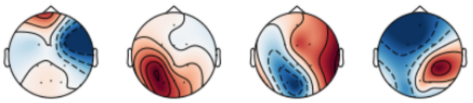
Component analysis was carried out (both PCA and ICA) in order to either join the information of two channels into a single feature or eliminate less telling channels. This might prove beneficial in terms of developing a more robust model that would generalize well to training different subjects and brain states. However, as long as finding one model hyper parameter tuning that is best for a specific session all information was vital.

As mentioned in the data recording subsection, six subjects data was recorded and their final cross-validate results are summarized in table 5.1. To conclude, the brain-computer interface side-task was a success: by connecting the onboard Raspberry Pi computer through a Mavlink to a ground station, it is possible to assign tasks to the drone using the EEG cap as is by pressing right and left arrows on a keyboard or controller on a transmitter.

| Subject | #sessions | f-score | comments |
|---|---|---|---|
| **A** | 30 | 0.94 | in depth classification analysis was done |
| **B** | 10 | 0.53 | ”noisy” channels in motor cortex |
| **C** | 15 | 0.83 | experienced with BCI, ”noisy” channels in motor cortex |
| **D** | 10 | 0.59 | ”noisy” channels in motor cortex |
| **E** | 20 | 0.69 | many distractions during recording |
| **F** | 5 | 0.56 | subject was tense, ”noisy” channels in motor cortex |

TABLE 5.1: Results



(A) $\mu$ frequency response of planned movement of limbs [**?** ].

# 6 Conclusion

# A An Appendix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus at pulvinar nisi. Phasellus hendrerit, diam placerat interdum iaculis, mauris justo cursus risus, in viverra purus eros at ligula. Ut metus justo, consequat a tristique posuere, laoreet nec nibh. Etiam et scelerisque mauris. Phasellus vel massa magna. Ut non neque id tortor pharetra bibendum vitae sit amet nisi. Duis nec quam quam, sed euismod justo. Pellentesque eu tellus vitae ante tempus malesuada. Nunc accumsan, quam in congue consequat, lectus lectus dapibus erat, id aliquet urna neque at massa. Nulla facilisi. Morbi ullamcorper eleifend posuere. Donec libero leo, faucibus nec bibendum at, mattis et urna. Proin consectetur, nunc ut imperdiet lobortis, magna neque tincidunt lectus, id iaculis nisi justo id nibh. Pellentesque vel sem in erat vulputate faucibus molestie ut lorem.

Quisque tristique urna in lorem laoreet at laoreet quam congue. Donec dolor turpis, blandit non imperdiet aliquet, blandit et felis. In lorem nisi, pretium sit amet vestibulum sed, tempus et sem. Proin non ante turpis. Nulla imperdiet fringilla convallis. Vivamus vel bibendum nisl. Pellentesque justo lectus, molestie vel luctus sed, lobortis in libero. Nulla facilisi. Aliquam erat volutpat. Suspendisse vitae nunc nunc. Sed aliquet est suscipit sapien rhoncus non adipiscing nibh consequat. Aliquam metus urna, faucibus eu vulputate non, luctus eu justo.

Donec urna leo, vulputate vitae porta eu, vehicula blandit libero. Phasellus eget massa et leo condimentum mollis. Nullam molestie, justo at pellentesque vulputate, sapien velit ornare diam, nec gravida lacus augue non diam. Integer mattis lacus id libero ultrices sit amet mollis neque molestie. Integer ut leo eget mi volutpat congue. Vivamus sodales, turpis id venenatis placerat, tellus purus adipiscing magna, eu aliquam nibh dolor id nibh. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed cursus convallis quam nec vehicula. Sed vulputate neque eget odio fringilla ac sodales urna feugiat.

Phasellus nisi quam, volutpat non ullamcorper eget, congue fringilla leo. Cras et erat et nibh placerat commodo id ornare est. Nulla facilisi. Aenean pulvinar scelerisque eros eget interdum. Nunc pulvinar magna ut felis varius in hendrerit dolor accumsan. Nunc pellentesque magna quis magna bibendum non laoreet erat tincidunt. Nulla facilisi.

Duis eget massa sem, gravida interdum ipsum. Nulla nunc nisl, hendrerit sit amet commodo vel, varius id tellus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ac dolor est. Suspendisse ultrices tincidunt metus eget accumsan. Nullam facilisis, justo vitae convallis sollicitudin, eros augue malesuada metus, nec sagittis diam nibh ut sapien. Duis blandit lectus vitae lorem aliquam nec euismod nisi volutpat. Vestibulum ornare dictum tortor, at faucibus justo tempor non. Nulla facilisi. Cras non massa nunc, eget euismod purus. Nunc metus ipsum, euismod a consectetur vel, hendrerit nec nunc.

# Bibliography


[1] https://www.campusnet.dtu.dk/cnnet/filesharing/download/447d61d5-23f8-4764-80ea-9879f19a4c3a. 31567$_{L}ecture_{1}{}_{I}ntroduction, slide20and27.$

[2] Robert H. Liebeck. A class of airfoils designed for high lift in incompressible flow. *Journal of Aircraft*, 10(10):610–617, 1973.

[3] Greenshields, C., J. Openfoam user guide. 2017.

[4] Drela, M. Xfoil: An analysis and design system for low reynolds number airfoils. *Mueller T.J. (eds) Low Reynolds Number Aerodynamics. Lecture Notes in Engineering*, 54, 1989.

[5] 3d vortex lattice method.

[6] Brochu, E., Cora, V.,M. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. 2010. doi: https://arxiv.org/pdf/1012.2599.pdf.

[7] Hernandez-Lobato, D., Gelbart, M.,A. Predictive entropy search for bayesian optimization with unknown constraints. 2015. doi: https://arxiv.org/pdf/1502.05312.pdf.

[8] Ponweiser, W., Wagner,T. Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. 2008. doi: https://doi.org/10.1007/978-3-540-87700-4_78.

[9] Picheny, V. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. 2013. doi: https://arxiv.org/pdf/1310.0732.pdf.

[10] Hernandez-Lobato, D., Hernandez-Lobato, J.,M. Predictive entropy search for multi-objective bayesian optimization. 2016. doi: https://arxiv.org/pdf/1511.05467.pdf.

[11] Chilenski, M. gptools documentation. 2017. doi: https://media.readthedocs.org/pdf/gptools/latest/gptools.pdf.

[12] Rasmussen, C., E., William, C.,K.,I. Gaussian processes for machine learning. 2006. doi: https://media.readthedocs.org/pdf/gptools/latest/gptools.pdf.

[13] . Tutorials Point (I) Pvt. Ltd. Genetic algorithms. 2016.

[14] Deb,A. A fast and elitist multiobjective genetic algorithm: Nsga-ii. 2002. doi: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.542.385&rep=rep1&type=pdf.

[15] http://deap.readthedocs.io/en/master/api/tools.html?highlight=nsga2#deap.tools.selNSGA2, 2017. NSGA2 code Documentation.

[16] Shih JJ, Krusienski DJ, Wolpaw JR. Brain-computer interfaces in medicine. *Mayo Clinic Proceedings*, 87(3):268–279, 2012.

[17] Han Yuan, Bin He. Brain-computer interfaces using sensorimotor rhythms: Current state and future perspectives. *IEEE Trans Biomed Eng.*, 61(5):1425–1435., 2014.

[18] http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/.

[19] http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[20] http://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-and-f-measures. 3.3. Model evaluation: quantifying the quality of predictions, 3.3.2.8. Precision, recall and F-measures.