NTNU
Norwegian University of
Science and Technology

Master's thesis

2020

Edvard Frimann Løes Narum

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Engineering Cybernetics

Edvard Frimann Løes Narum

Mission Planning for Fixed-wing UAVs in Wind and Icing Conditions

June 2020

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Mission Planning for Fixed-wing UAVs in Wind and Icing Conditions

Edvard Frimann Løes Narum

Cybernetics and Robotics
Submission date: June 2020
Supervisor: Tor Arne Johansen
Co-supervisor: Richard Hann

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Contents

Abstract

Harsh weather conditions such as wind and icing are a severe curtailment to the operations of unmanned aerial vehicles in terms of performance and reliability. Knowledge of concrete atmospheric parameters in the area surrounding a mission can open many options in terms of how to best traverse said environment. The advent of hybrid electric propulsion systems in such vehicles significantly alter operational range and subsequently the versatility of the platform, but hybrid electric vehicles are equally vulnerable to difficult atmospheric conditions. This dissertation will show that taking meteorological data into account when performing path planning optimisation on such a platform can lead to improvements of up to 50% in terms of both energy efficiency and time efficiency, when compared to a standard, straight cruising flight. Presented here is a model of a hybrid electric, unmanned aerial vehicle equipped with an icing protection system, intended to be employed in wintertime conditions, and a waypoint guidance algorithm realised through particle swarm optimisation.

Sammendrag

Sterk vind og isete forhold er blant de største utfordringene rundt bruk av ubemmanede småfly over lengre distanser. Kjennskap til konkrete værforhold i området der flyvningen skal foregå, gir muligheten til å minimere de uønskede værforholdenes påvirkning på flyet ved å anvende algoritmer for ruteplanlegging og optimalisering. Primærkonfigurasjonen av flyet vil være utrustet med et isbeskyttelsessystem som hindrer is i å danne seg på vingene på bekostning av energi. Hybridelektriske propulsjonssystemer er en måte å drastisk utvide rekkevidden, og dermed sikkerhetsmarginene, til ubemannede småfly på. Dog er hybridelektriske småfly like lett påvirkelige av vanskelige værforhold. Denne avhandlingen vil vise at ved å ta meteorologiske værdata med i betraktningen under ruteplanlegging for slike hybrid-elektriske fartøy kan man nær halvere energiforbruk og tidsforbruk i sammenligning med rett flyvning under konstant marsjfart.

Nomenclature

Units

℃	Celsius
Ω	Ohm
A	Ampere
Ah	Ampere hour
cc	Cubic centimetre
fpm	Feet per minute
g	Gram
h	Hour
hp	Horsepower
Hz	Hertz
J	Joule
K	Kelvin
kg	Kilogram
l	Liter
m	Metre
mol	Mole
Pa	Pascal
s	Second
V	Volt
W	Watt
Wh	Watt hour
Ws	Watt second

Metric Prefix

µ	micro	10^{-6}
m	milli	10^{-3}
c	centi	10^{-2}
k	kilo	10^{3}
M	mega	10^{6}
G	giga	10^{9}

Variables and constants

α	Angle of attack
χ	Course over ground
η_{prop}	Propeller efficiency
γ	Flight path angle
γ_a	Air-mass-referenced flight-path angle
ψ	Aircraft heading
ψ_w	Horizontal wind heading
ρ	Air density
ρ_w	Water density
θ	Pitch angle
C_D	Drag coefficient
C_L	Lift coefficient
e_a	Water vapour partial pressure
e_{sat}	Saturated water vapour pressure
F_{drag}	Drag force
F_{lift}	Lift force
g	Gravitational constant
h	Height
L_0	Standard temperature lapse rate
LWC_c	Liquid water content per volume
LWC_m	Liquid water content mixing ratio
M	Molar mass of air
m	Aircraft mass
N_c	Droplet concentration
p	Pressure
P_0	Static sea level pressure
P_{IPS}	IPS power load
q	Specific humidity
R_d	Specific gas constant for dry air
R_u	Universal gas constant

RH	Relative humidity
S	Wing surface area
T	Temperature
t	Time
T_0	Sea level standard temperature
T_{prop}	Propulsive thrust
V_a	Absolute airspeed
V_g	Absolute ground speed
V_w	Absolute wind speed
w_d	Downwards wind
w_e	Eastbound wind
w_n	Northbound wind
xx	ENU east position
yy	ENU north position

Abbreviations

AOA	Angle of attack
CFD	Computational fluid dynamics
COG	Course over ground
CPU	Central processing unit
DSD	Droplet size distribution
DTM	Digital terrain model
ECEF	Earth-centered, earth-fixed
EM	Electric motor
ENU	East-north-up
FAA	Federal Aviation Administration
GPU	Graphics processing unit
HEPS	Hybrid electric power system
HEUAV	Hybrid electric UAV
ICE	Internal combustion engine
IPS	Icing protection system
ISA	International standard atmosphere
LiPo	Lithium polymer
LWC	Liquid water content
MAV	Micro aerial vehicle
MET	Norwegian Meteorological Institute
MTOW	Max take-off weight
MVD	Median volume diameter
NED	North-east-down
NetCDF	Network Common Data Form
PSO	Particle swarm optimisation
RAM	Random access memory
RH	Relative humidity
UAV	Unmanned aerial vehicle
UTM	Universal transverse mercator
WCS	Web coverage service
WRF	Weather Research and Forecasting

List of Figures

List of Tables

1 Introduction

1.1 Background and Motivation

Unmanned aerial vehicles (UAVs) have seen a surge in development and research over the past few years. Globally, the UAV market is expected grow from \$25.59 billion in 2018 to \$70.28 billion by 2029 [1], indicating the enormous commercial and scientific potentials of the technology. A source of this growth could be in the enormous versatility of the UAV as a scientific or transportational platform, as they are cheap, quick to deploy and often more expendable than the alternatives. Typically, UAVs are operated by simple flight planning software, where waypoints and altitudes are manually entered and subsequently flown to in order. Although this method is sufficient for shorter or highly specific missions, it is unable to use advantageous or avoid disadvantageous atmospheric conditions. The operational environment of the UAV plays a significant role in determining flight time and energy consumption, and should be taken into account in path planning to truly realise the potential of UAV as a versatile platform. This work aims to illustrate the potential gains in in-flight performance through path planning optimisation.

Atmospheric winds and icing conditions are two of the primary disturbances to UAV operations. Both can be critical to the stability and efficiency of the aircraft. Taking wind maps into account when performing path planning optimisation can significantly improve flight times without increasing energy expenditure. Icing conditions, on the other hand, are typically best avoided, but the emergence of icing protection systems (IPS) have provided a way of safely traversing such conditions at a cost of power. These two hazards are the main topic of this dissertation.

Development of battery powered electric UAVs has been motivated by a desire to reduce emissions, to lower the sound profile of the vehicle and to shrink the size of the system, due to batteries' high specific power. A major drawback to this design is the operational range of the vehicles, where most medium sized fixed-wing designs, despite being significantly more efficient than their multirotor counterparts, are limited to around 2 hours of active flight time. This is a severe restriction to mission profiles and the applicability of drones for commercial, scientific and surveillance based purposes. Installing a hybrid electric power system mitigates the low specific energy of the battery by adding energy rich fuel that can be used to recharge the battery cells during operation, while still benefiting from the available power from the battery. Hybrid electric UAVs (HEUAV) are in their relative infancy, but are emerging as promising alternatives to established technologies [2]. HEUAVs can typically provide upwards of a five-fold increase in flight-time at a minimal cost to system weight, which will be explored here.

1.2 Previous Work

This thesis begun as an unpublished project report from the autumn of 2019 as [3], and was expanded upon as a master's thesis the following semester. As there were few fundamental changes to the method of the project through this transition, certain sections of the thesis will be more or less unchanged to that which could be found in the project report. A more condensed paper based on this work was also produced, set to be published in September of 2020 as [4].

The project is a continuation of the work *Long range path planning using an aircraft* *performance model for battery powered sUAS equipped with icing protection system* from chapter 3 of [5]. The thesis in question is a thorough work on the effects of harsh weather on UAVs and how they can be mitigated through path planning methods.

Unfavourable weather conditions are another severe restriction to UAV operations. Work has been done on the effects of wind [6] and performing guidance therein [7], but much less so on operations in icing conditions. This has only recently become an active area of research, such as [8]–[10] exploring stability and general operations, and remains a significant challenge for mission profiles in colder climates.

HEUAVs is a relatively new area of research [11], although modeling of general hybrid electric vehicles is a more well-established topic as in [12]. Energy optimisation of a HEUAV through path planning was done in [13] although the focus here was more motor control and short distance waypoint guidance. There has also been focused research on combustion engine control of a HEUAV, such as a neural network based approach in [14].

1.3 Objective

The overarching objective of the following work is developing algorithms for optimal path planning of a hybrid electric UAV given realistic meteorological input data. To realise this, a sufficiently accurate performance model of the aircraft and the hybrid electric powertrain must be developed, but the system should be customisable and modifiable enough to be employed with minimal effort to different platform configurations. The platform will be outfitted with an icing protection system using a model of the required energy expenditure for ice mitigation dependent on the corresponding atmospheric conditions.

Optimisation will be made both with respect to energy consumption, and to overall time usage. To demonstrate the potential savings in these regards, the algorithm will be deployed on multiple data sets with varying severity in weather conditions, on a realistic HEUAV configuration. A model of aerodynamic degradation as a result of airframe icing will also be employed. This is to compare the effects of icing to the power loads of an icing protection system. Further, the results will be discussed, as well as giving an outline of what work will be done in the future to improve on the realism and reach a system that could feasibly be deployed in a real-life scenario.

1.4 Structure

The report is divided into the following sections. Section 2 gives an overview of the required knowledge and equations for performing accurate weather estimations, the acquisition of weather and elevation data as well as a touch on the relevant coordinate frame conversions. Section 3 gives an outline of an aircraft performance model including the relevant equations of motion, force estimations, power requirements and the icing protection system. Additionally, a model for estimating performance losses as a result of airframe icing on a no-IPS configuration is presented. Section 4 explains the design of a hybrid electric powertrain, including the options in configurations, combustion engines and a battery performance model. Section 5 explains the implementation of the algorithm, including the optimisation method, cost functions and specific HEUAV configuration to be used in simulations. Section 6 presents a potential commercial application for the platform, and specifies some concrete mission parameters representing such an application. Section 7 presents the results obtained by employing mission parameters on the specific weather data in the simulation environment, as well as discussing the implications of the found results. Section 8 gives suggestions to further improvements that could be made to the system to increase its usability and applicability, and Section 9 provides a brief conclusion.

2 Atmospheric weather modeling

Estimations and projections of the geographical distributions of atmospheric weather parameters are at the core of the optimisation problem. The efficiency of a given path is determined by the aircraft's performance relative to the surrounding meteorological conditions. Acquiring these parameters is a matter of employing mathematical models describing the parameter's relations, and downloading spatial weather maps, in this work provided by the Norwegian Meteorological Institute (MET). The following section will explain the atmospheric models employed in this work, as well as the weather conditions considered to be icing conditions, coordinate frame conversions and the process of data acquisition.

2.1 Atmospheric estimations

2.1.1 Barometric pressure

Barometric pressure is a measure of the air pressure at a point within the atmosphere. This value is typically approximated by the hydrostatic pressure at a point, meaning the weight of the air column directly above the given area. In this work, pressure is assumed to be uniformly spread at a given height. The pressure p is estimated in pascal through the International Standard Atmosphere (ISA) barometric formula, which relates height over sea level h to pressure by:

$$p = P_0 \cdot \left(\frac{T_0}{T_0 + L_0 \cdot h} \right)^{\frac{g \cdot M}{R_u \cdot L_0}} \tag{2.1.1}$$

where P_0 is the static sea level pressure of 101325 Pa, T_0 is the sea level standard temperature of 288.15 K, L_0 is the standard temperature lapse rate which is around -0.0065 K/m within 11000 metres of altitude, R_u is the universal gas constant of 8.3144598 J/(mol·K), g is the gravitational acceleration and M is the molar mass of air at 0.0289644 kg/mol. Atmospheric pressure has significant effects on the induced drag, lift, ice accumulation and general performance of the aircraft, and is therefore of interest in this application.

2.1.2 Air density

Also playing a major role in the aerodynamic forces on the vehicle is the air density ρ in kg/m^3, being a measure of the mass of air per unit volume in the atmosphere. The density is calculated by the ideal gas law and varies with surrounding temperature T and pressure p:

$$\rho = \frac{M \cdot p}{R_u \cdot T} \tag{2.1.2}$$

where M and R_u are as specified in 2.1.1, temperature T is in Kelvin and pressure p in pascal as described in Equation 2.1.1.

2.2 Icing conditions

Icing conditions are combinations of atmospheric conditions that lead to the accumulation of ice on an unheated structure. In practical terms, this occurs due to pure, uncrystallised water in clouds existing in a *supercooled* state, meaning the water has a temperature below 0℃, but is still not frozen. This phenomenon occurs due to the surface tension in water droplets suspended in air. Supercool liquid droplets are the main component of clouds between 0 and -10℃, while between -10 and -20℃ coexists with ice crystals [15]. In temperatures below -20℃ clouds generally consist of primarily frozen crystals, except for certain types of clouds like the cumulonimbus, which may contain supercooled liquid water at temperatures approaching -40℃.

Icing on the wings of a UAV then occurs through exposure to these droplets, where the rigid airframe acts as a nucleus to the crystallisation of the water. Ice will then gradually build up on the parts of the airfoil exposed to the liquid water, which can be critical to flight operations. Ultimately, this may lead to stalling and crashing the UAV, and has therefore recently become an area of study for operations in colder climates. The performance-wise effects of icing on an aircraft will be explained in greater detail in Section 3.5.

In this work, an aircraft is considered to be in icing conditions when operating in a temperature below 0°C, experiencing a *relative humidity* above 0.99 and a *liquid water content* above 0.01 g/m^3.

2.2.1 Relative humidity

Relative humidity (RH) is the ratio of the partial pressure of water vapour e_a to the saturated water vapour pressure at a given temperature e_{sat}. Relative humidity is thus expressed as:

$$RH = \frac{e_a}{e_{sat}} \tag{2.2.1}$$

where e_a in Pa is given by:

$$e_a = \frac{q \cdot p}{0.622 + 0.378q} \tag{2.2.2}$$

where p is surrounding air pressure in Pa and q is the specific humidity, which is ratio of the mass of water vapour to the total mass of the air parcel. The saturated water vapour pressure e_{sat} is dependent on the temperature T in ℃ and is given by:

$$e_{sat} = 2 + 10^{\frac{0.7859+0.03477\cdot T}{1+0.00412\cdot T}} \tag{2.2.3}$$

Intuitively, the partial pressure e_a is a measure of the pressure the gas exerts had it been alone in a volume. The saturated vapour pressure e_{sat}, however, is the pressure the gas in thermodynamic equilibrium exerts with its liquid phase. A RH above 0.99 then implies that the air parcel's water vapour content is saturated, that is it holds as much water vapour as it is capable of. A relative humidity of 0.99 and above occurs in fog and clouds.

2.2.2 Liquid water content and median volume diameter

Liquid water content (LWC) is a measure of the mass of water within a body of air, typically in the form of a cloud, and is of interest as one of the input parameters to the icing protection system regression models. Specifically, the LWC parameter in the regression models is in g/m^3, and can be extracted from the mixing ratio LWC_m in kg/kg by:

$$LWC_c = \frac{LWC_m p}{R_d T} \cdot 10^3 \tag{2.2.4}$$

where p and T are pressure in Pa and temperature in K respectively, and R_d is the specific gas constant for dry air of 287.058 J/(kg·K). LWC_m is one of the available meteorological data sets, hence the need for the above conversion.

Another parameter used by the icing protection regression models is the water droplet median volume diameter (MVD) in µm. MVD is, as the name implies, the median of the diameters at the largest points of a set of droplets in a cloud, indicating the expected size of the droplets one encounters. As in [16] the value can be calculated by:

$$MVD = \left(\frac{3.672 + \mu}{\lambda}\right) \tag{2.2.5}$$

where μ is a size distribution shape parameter given by:

$$\mu = min\left(15, \frac{1000}{N_c} + 2\right) \tag{2.2.6}$$

where N_c is the droplet concentration set to 100 cm^{-3}, meaning that a cubic centimetre of air is expected to contain a constant 100 droplets. λ from 2.2.5 is the slope parameter given by:

$$\lambda = \left[\frac{\pi}{6}\rho_w \frac{\Gamma(4+\mu)}{\Gamma(1+\mu)} \frac{N_c}{LWC}\right]^{1/3} \tag{2.2.7}$$

where ρ_w is the density of water and Γ is a gamma distribution.

The proposed relationship between LWC and MVD seems to vary a bit throughout the literature. In particular, there are two widely used models that seem contradictory; that of [17] adapted by [16], presented above, to that of the engineering standard 14 CFR, Part 25 App. C of the Federal Aviation Administration (FAA) of the USA. Although only the model presented above will be used in the simulations of this work, it is deemed valuable to provide an overview of the two models, and give a brief interpretation to the nature of their contradiction. The first model will from now be referred to as the Weather Research and Forecasting (WRF) model, while the second will be referred to as the FAA model. The inverse proportionality of the models in question is illustrated in Figure 2.1 and Figure 2.2. In most FAA models an increase in MVD corresponds to a decrease in LWC, while the WRF model proposes an increase in MVD leads to an increase in LWC.

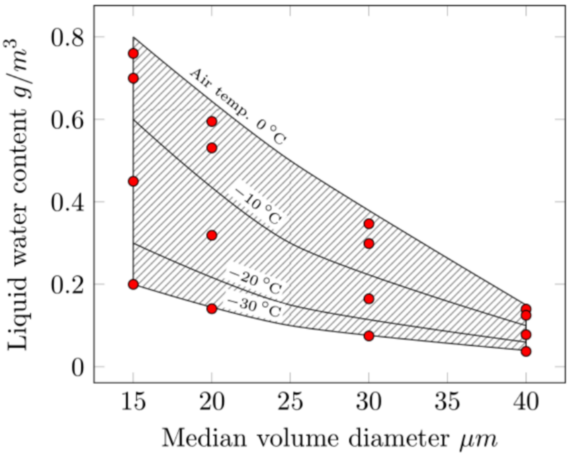

Figure 2.1: Example FAA model of LWC to MVD, from [18], modified from App. C to 14 CFR 25

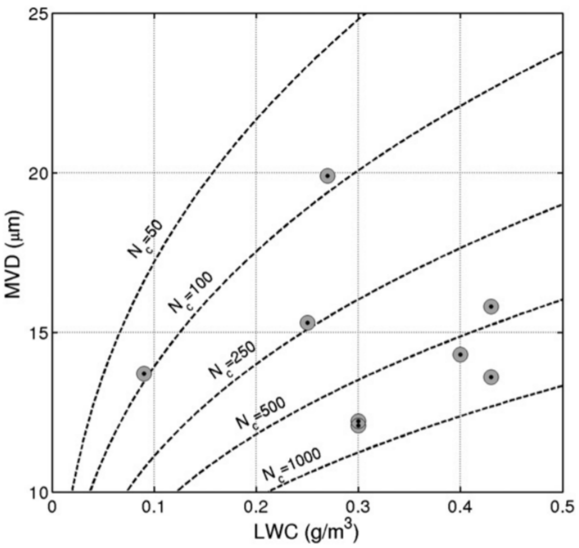

Figure 2.2: WRF model of LWC to MVD under different N_c, from [16]

The WRF model is typically employed with a constant droplet concentration dependent on the environment, where 100 cm^{-3} is a typical value for clean or maritime air, while more polluted continental air can see a concentration of 250 cm^{-3} or higher. A key assumption to the validity of this model is that the droplet size distribution (DSD), indicating the range of expected droplet sizes, is monodisperse in a cloud, meaning it contains a uniform distribution of droplet sizes. In well-mixed clouds the LWC will increase from close to zero around cloud base, increasing to a maximum value around the top of the cloud. Assuming a constant N_c throughout the cloud, the DSD will then intuitively shift to the right moving upwards, corresponding to an increase in MVD (which can be seen as a median value of the DSD). This is because if the liquid water content is low, and the cloud is thought to have 100 droplets per cubic centimetre, the median size of the droplets must be very low. If the LWC increases, and the droplet concentration remains constant, the only way to follow such an increase in LWC is by the MVD increasing.

The situation above presents only a subset of potential icing clouds. For instance, a cloud could be so deep that the amount of large drops towards the top would force a collision-coalescence, leading to a bi-modal DSD. For example, in a deep cloud with a high LWC at the top, the droplets might initiate a cascade of collisions severely increasing MVD in the area, and possibly starting a rain mode limited to the top of the cloud. Thus, the DSD of the cloud is no longer monodisperse, and the WRF model collapses. The FAA models are based on a wide range of cloud types, covering a large range of possible DSDs from varying climates. Determining which distribution and mode is present in a given cloud, which would be required to employ these models, is considered outside of the scope of this work. Only the WRF model presented above will be used in this project's simulations, as it seems as a large number of well-mixed wintertime clouds follow this distribution.

2.3 Wind

Wind is one of the primary disturbances in UAV operations, and plays a significant factor in energy expenditure and flight times. Given sufficient wind speeds, flying a stretch in headwind can easily double flight times, as will be demonstrated in Section 7. For this reason, there are large potential energy and time savings by optimising a path taking geographical wind maps into account. Spatial maps of horizontal wind, separated into east-bound and north-bound maps with magnitudes varying with altitude are available for download. These must be properly interpreted to be of use in the system. Not considered in this work are adverse winds such as gusts or variable wind, as all simulations are based on static data sets. The implications of this assumption, and an overview of alternatives will be presented in Section 8.2.

In addition to horizontal wind, vertical wind is also present in most UAV applications. Although the magnitude of vertical wind vectors is lower than that of zonal and meridional wind, the effect of updraft on a UAV can lead to gaining "free" lift, which could play a factor in the calculation of an optimal path. Several factors can lead to vertical wind. One is the pressure gradient force, forcing air parcels from high pressure to low pressure, which is usually the case in a vertical column as air pressure is highest at ground level as discussed in Section 2.1.1. Ridge lift is another source of vertical wind, which occurs when horizontal wind meets an obstacle such as a cliff face, forcing the wind upwards. As the air parcel rises the subsequent decrease in pressure causes the air to expand, lowering its temperature. With this cooling, clouds and precipitation may occur, leading the windward sides of cliffs and mountains to have higher rates of icing conditions [15]. The wind may then either descend on the leeward side, or return the opposite way from the cliff side at a higher altitude. Thermal columns are another potential source of updraft, stemming from pressure differentials as a result of uneven heating of Earth's surface. Vertical wind is used by a variety of birds, as well as glider planes, and utilising this phenomenon in a UAV application has significant potential.

Let V_w^i be the wind speed vector with respect to an inertial north-east-down (NED) frame such that

$$V_w^i = \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix} \tag{2.3.1}$$

where w_n and w_e are the north- and east-component of the wind speed vector respectively and w_d is the downwards facing component. The magnitude of the wind vector thus becomes

$$V_w = \sqrt{{w_n}^2 + {w_e}^2 + {w_d}^2} \tag{2.3.2}$$

and the horizontal projection of the heading of the wind vector becomes:

$$\psi_w = \arctan 2(w_n, w_e) \tag{2.3.3}$$

These are of great importance to the resulting ground speed of the aircraft, which will be further explained in Section 3.1.

2.4 Data acquisition

Although forecasts would be needed for an operational planning system, historical data are preferred for testing, and used in this work. Historical meteorological data of the Scandinavian peninsula is available through a webapp called THREDDS Data Server hosted by the Norwegian Meteorological Institute. Here, a multitude of weather data from 2016 and onwards is available for download, where one can specify a desired date, time and geographical coverage and receive relevant data in the Network Common Data Form (NetCDF). Fully processed data is currently available at 3 hour intervals, i.e. at 00, 03, 06, etc. The service also provides a subset of parameters hourly, but these are not archived and will therefore have to be downloaded by the user within 24 hours of being processed.

Relevant to this application are the following downloaded maps and parameters. Two separate three-dimensional wind maps, detailing horizontal wind, indicate longitudinal and lateral wind. They are structured as a cube matrix, where each slice indicates a certain altitude, and the maximum and minimum altitude can be specified before downloading. Another wind map, detailing upwards wind, stored in the same format as the horizontal maps, is also acquired. To get a map of the relative humidity, a map of the spatial temperature, pressure based on altitude and a map of the specific humidity is downloaded. This data is then used to generate the RH following the relations presented in Section 2.2.1. A map of the LWC_m is also available, which is subsequently processed into LWC_c using Equation 2.2.4. All weather maps are indexed with matrices associating values of latitudes and longitudes to cells in the map matrix. All the mentioned data is provided with a zonal and meridional resolution of 2,5 km while the resolution of the vertical slices are roughly 100 m, but the resolution is decreasing with increasing altitude, meaning that more detailed information is present close to ground level. Assuming constant and evenly distributed parameters throughout every cell is a simplification, but one deemed necessary.

Kartverket, or *The Norwegian Mapping Authority*, supply a number of detailed maps in different categories for the Norwegian peninsula through their service *Kartkatalogen* (= *the map catalogue*). Included are *digital terrain models* (DTM) that illustrate elevation at a set resolution for a specified geographical area. For this application a DTM map for a section of the Universal Transverse Mercator (UTM) zone 33N was downloaded at a resolution of 10 metres, but later further downscaled to 200 metres for computational and practical purposes. The data can be downloaded as a Web Coverage Service (WCS) which is most easily done by using a given WCS URL in a dedicated program such as QGIS [19]. A graphical representation of one of the DTMs used for simulations in this work can be seen in Figure 2.3. The blacked out section in the bottom right is the Finnish border, which is data not distributed by Kartverket for legal reasons.

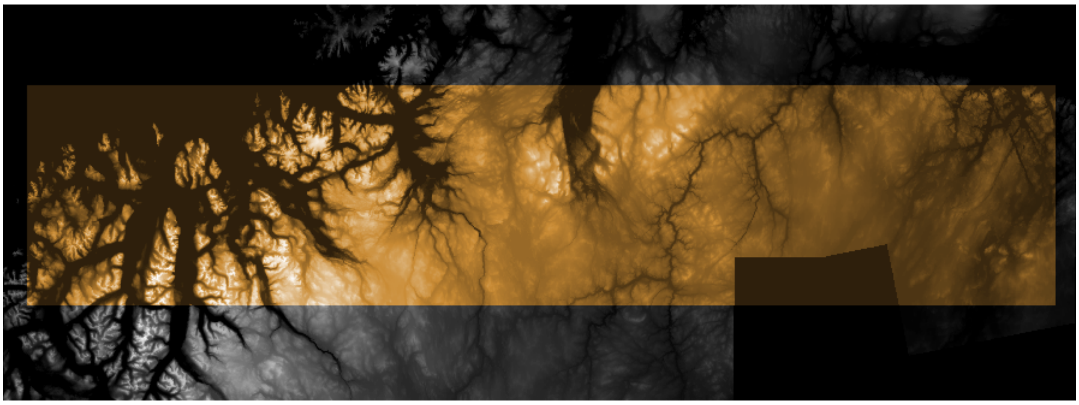

Figure 2.3: Partial UTM 33N DTM

2.5 Coordinate and reference frames

To relate the world and the UAV within it, coordinate frames are needed. In this work the east-north-up (ENU) frame is used as the local frame. ENU coordinates are represented in a tangential plane with some self defined origo, using the x-axis to represent east, y to represent north and z to represent height over origo. Here, origo is placed at sea level on the midpoint between the start and the end of a mission. As all weather data here is represented in geodetic coordinates (latitude, longitude), proper conversions between the two frames need to be performed. To accomplish this one needs to go through the earth-centered, earth-fixed (ECEF) coordinate system which has its origo in the center of the earth, and the x-axis pointing out towards a longitude of 0°. The relation between the two frames can be seen in Figure 2.4.

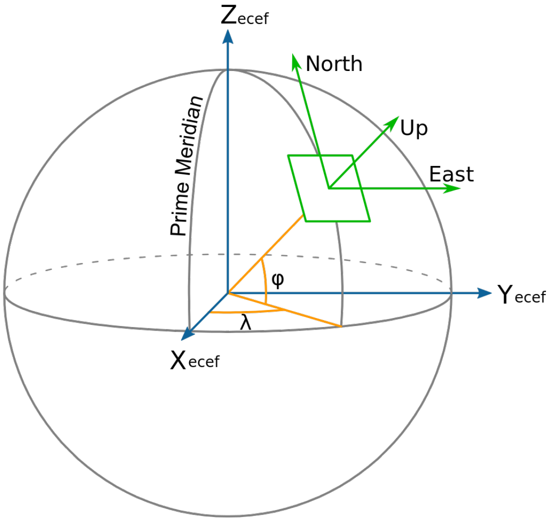

Figure 2.4: ENU frame represented by ECEF, from [20]

ENU coordinates x, y and z can be found from ECEF coordinates X, Y and Z with ENU origin X_r, Y_r and Z_r by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\sin\lambda_r & \cos\lambda_r & 0 \\ -\sin\phi_r\cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r \\ \cos\phi_r\cos\lambda_r & \cos\phi_r\sin\lambda_r & \sin\phi_r \end{bmatrix} \begin{bmatrix} X - X_r \\ Y - Y_r \\ Z - Z_r \end{bmatrix} \tag{2.5.1}$$

where λ_r and ϕ_r is the origin longitude and latitude respectively. Conversely, going from ENU to ECEF is the opposite operation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -\sin\lambda_r & -\sin\phi_r\cos\lambda_r & \cos\phi_r\cos\lambda_r \\ \cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r\sin\lambda_r \\ 0 & \cos\phi_r & \sin\phi_r \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} \tag{2.5.2}$$

Geodetic coordinates ϕ, λ and h, or latitude, longitude and height over sea level respectively can be converted to ECEF through:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (n+h)\cdot\cos\phi\cos\lambda \\ (n+h)\cdot\cos\phi\sin\lambda \\ (n\cdot(1-e)+h)\cdot\sin\phi \end{bmatrix} \tag{2.5.3}$$

where:

$$n = \frac{a}{\sqrt{1 - e \cdot \sin\phi \sin\phi}} \tag{2.5.4}$$

and a and e are constants valued at $a = 6378137.0$ and $e = 6.6943799901377997 \cdot 10^{-3}$. To convert from ECEF to Geodetic coordinates, Ferrari's solution is applied. This method is expressed in code form in Appendix A.1.

3 Aircraft performance model

A mathematical model of the aircraft is essential to determining and subsequently improving upon operational performance. Following is an overview of the relevant equations to express the UAVs performance in terms of aerodynamic forces and moments and equations of motion. It is worth noting that in more control oriented algorithms, one would need higher accuracy and resolution than the models presented here, as this work is primarily focused around long-distance guidance. Also in the following chapter are the proposed models for propulsive power and an icing protection system.

3.1 Aircraft equations of motion

The equations of motion of the aircraft propose the behaviour and state of the vehicle at an instant. Most states are interdependent, as well as affected by the surrounding environment.

3.1.1 Airspeed

All aerodynamic forces acting on an aircraft are directly proportional to the vehicle's *airspeed* V_a. Airspeed is a measure of the aircraft's velocity relative to the surrounding air, and is a combination of the vehicle's *ground velocity* vector V_g^i and the surrounding wind velocity vector V_w^i, given in an earth-fixed inertial frame by

$$V_a^i = V_g^i - V_w^i \tag{3.1.1}$$

where V_g^i is the vector describing the UAV's motion with respect to the ground below, V_a^i describes the projections of the UAV's relative air velocity vector into the same NED frame. V_a and V_g are the magnitudes of the vectors V_a^i and V_g^i respectively. Most UAVs are rated for a specified max airspeed, but that does not imply a maximum speed over ground in a practical application. For instance, if a UAV operates in its maximum V_a of 28 m/s and flies directly against a wind of 24 m/s, it would be seen as slowly creeping forward at 4 m/s from the ground. Wind speeds typically constitute between 20-50% of operational airspeed for a UAV [21]. This has great implications for the potential increases in efficiency by optimal planning for predicted wind speeds.

Defining γ as the *flight path angle*, which is the angle between the horizontal plane and the ground velocity vector, we can express the time derivative of the UAV's altitude h by:

$$\dot{h} = V_g \sin\gamma \tag{3.1.2}$$

The flight path angle γ will be one of the optimisation variables indicating climbing or descending along the path. V_g^i can be expressed in element form as:

$$V_g^i = V_g \begin{pmatrix} \cos\chi\cos\gamma \\ \sin\chi\cos\gamma \\ -\sin\gamma \end{pmatrix} \tag{3.1.3}$$

where χ is the *course over ground* (COG), calculated as the angle with respect to north between two desired discrete positions. The airspeed V_a^i can be expressed similar to V_g^i

as:

$$V_a^i = V_a \begin{pmatrix} \cos\psi\cos\gamma_a \\ \sin\psi\cos\gamma_a \\ -\sin\gamma_a \end{pmatrix} \tag{3.1.4}$$

where ψ is the aircraft heading and γ_a is the *air-mass-referenced flight-path angle* defined as the angle from the horizontal plane to V_a^i. Eq. (3.1.1) can now be rewritten:

$$V_a \begin{pmatrix} \cos\psi\cos\gamma_a \\ \sin\psi\cos\gamma_a \\ -\sin\gamma_a \end{pmatrix} = V_g \begin{pmatrix} \cos\chi\cos\gamma \\ \sin\chi\cos\gamma \\ -\sin\gamma \end{pmatrix} - \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix} \tag{3.1.5}$$

Squaring the norm of each side of Eq. (3.1.5) gives:

$$V_g^2 - 2V_g \begin{pmatrix} \cos\chi\cos\gamma \\ \sin\chi\cos\gamma \\ -\sin\gamma \end{pmatrix}^T \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix} + V_w^2 - V_a^2 = 0 \tag{3.1.6}$$

which can be solved for V_g to give a measure of the actual ground speed of the aircraft. The horizontal component of V_g can then be used to get a measure of the flight time t_i of a discretised step i:

$$t_i = \frac{L_i}{V_g \cos\gamma} \tag{3.1.7}$$

where L_i is the horizontal distance between two desired positions. This is then used to calculate altitude changes based on Eq. (3.1.2), as well as energy consumption from expended power. From Eq. (3.1.5) one can also obtain:

$$\gamma_a = \arcsin\left(\frac{V_g \sin\gamma + w_d}{V_a}\right) \tag{3.1.8}$$

which is useful in determining pitch and aerodynamic forces. Pitch angle θ relates to γ_a and the angle of attack α through:

$$\theta = \gamma_a + \alpha \tag{3.1.9}$$

which is a measure of the rotation of the aircraft about the transverse axis in relation to the horizontal plane of the ground below.

3.1.2 Heading

Heading is a measure of the direction in which the nose of the aircraft is pointing in the horizontal plane. Under no wind this will equal the direction of the ground speed, but when wind occurs the heading ψ can be determined by:

$$\psi = \chi - sin^{-1}\left(\frac{V_w}{V_a sin(\psi_w - \chi)}\right) \tag{3.1.10}$$

where total wind speed V_w is defined as in Equation 2.3.2, V_a is the total airspeed, ψ_w is the wind direction from Equation 2.3.3. χ is the COG which is the direction of the ground speed calculated at discretisation step i by:

$$\chi_i = \text{atan2}(xx_{i+1} - xx_i, yy_{i+1} - yy_i) \tag{3.1.11}$$

where xx_i and yy_i are ENU positions at step i on the x-axis and y-axis respectively. There is a big advantage to defining the four optimisation variables as V_a, γ, xx and yy. Because if these four variables are known, and provided knowledge of the wind vector is present, all values above (except for θ and α) are deterministic.

3.2 Aerodynamic forces and moments

There are multiple forces acting on a UAV during flight. The four main forces are the thrust from the propulsion system, the gravitational weight of the vehicle and the two aerodynamic forces, lift and drag, all illustrated in Figure 3.1. These last two stem from the pressure differential generated by the airfoil as it moves through the surrounding air. Multiple variables affect this property, such as air density, pressure, airfoil shape and the attitude of the UAV.

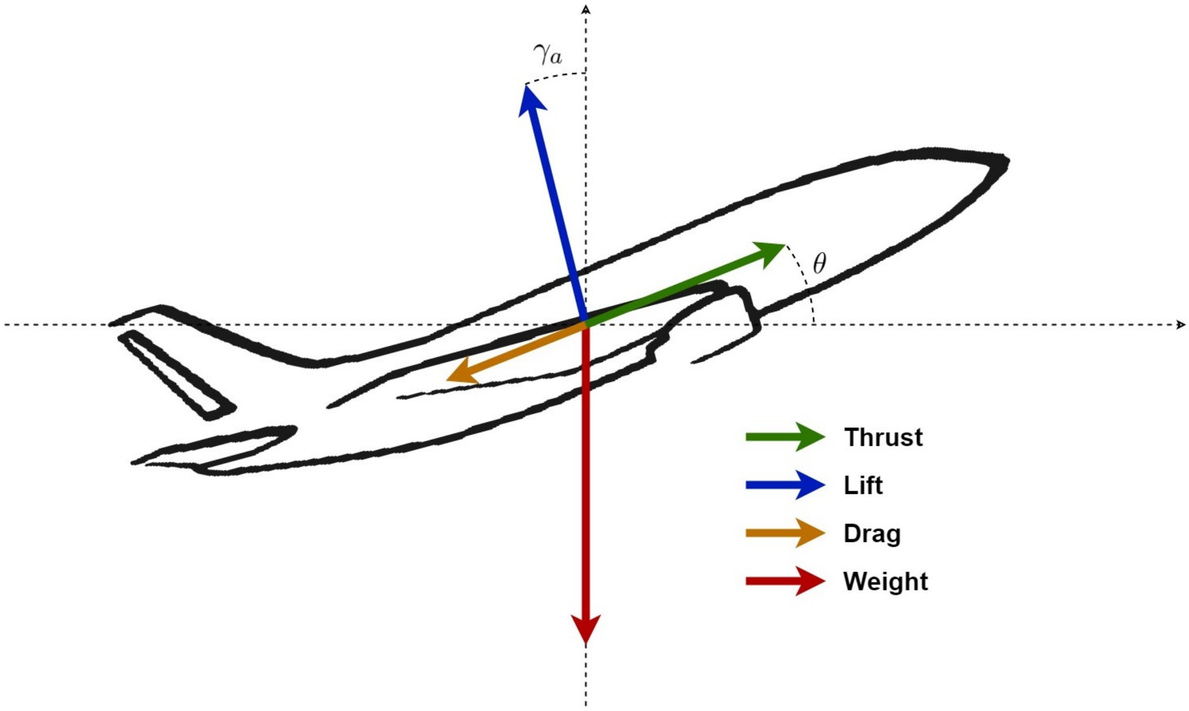

Figure 3.1: Forces acting on an aircraft during climb

To maintain completely level, cruising flight, the generated lift force must equal the weight while the horizontal thrust force compensates for the drag. However, during climb and descent the matter is slightly more complicated. Somewhat counter-intuitively, the generated lift force, here defined as acting perpendicular to the frame of the airspeed, actually decreases when climbing. As the thrust and drag vector work parallel to each other, the excess thrust still has a vertical component. This, together with the vertical component of the lift force, must together equal the weight force to achieve a constant, steady climb. The total lift force is given by:

$$F_{lift} = \frac{1}{2}\rho V_a^2 S C_L \tag{3.2.1}$$

where ρ is the air density (see 2.1.2), V_a is the airspeed, S is the surface area of the wings and C_L is the aerodynamic lift coefficient. Similarly, the absolute drag is the force acting horizontally backwards in the body frame of the UAV, given by:

$$F_{drag} = \frac{1}{2}\rho V_a^2 S C_D \tag{3.2.2}$$

where C_D is the aerodynamic drag coefficient.

3.2.1 Aerodynamic coefficients

The aerodynamic coefficients of a UAV relates the forces acting on the vehicle to the vehicle's state at an instant. Determining C_L and C_D is a complicated procedure, and is an important step to quantify the performance of an aircraft. In this work, a desired lift coefficient in a single step is calculated based on γ_a and the weight of the vehicle:

$$C_L(\gamma_a, \rho, V_a) = \frac{2mg\cos\gamma_a}{\rho S V_a^2} \tag{3.2.3}$$

where m is the total mass of the aircraft in kg and g is the gravitational constant. Using a model adapted from [18], the desired C_L is then used to determine the appropriate angle of attack (AOA) to achieve said lift. An advantage of this approach is that it gives a deterministic estimate of the pitch θ from Equation 3.1.9. With the appropriate angle of attack, an associated drag coefficient is also extracted from the model. Continuous relations between AOA, C_L and C_D with a clean airfoil were achieved using a simple linear interpolation between simulated values under integer AOAs. The relations can be seen, together with their icing-degraded counterparts, as the green curves in Figure 3.5 and Figure 3.6. One thing to note is that the simulations were performed on an RG-15 airfoil, which has a larger chord than the platform used in this work. Therefore, the actual coefficients are likely to be less efficient for the PX-31, that is the drag-to-lift ratio, $\frac{C_D}{C_L}$, is likely to be somewhat higher. However, there is no clear conversion between the efficiency of the different airfoils, so the curves for the RG-15 will be used in this work, despite providing slightly lower energy expenditure than what would be expected. Obtaining accurate estimations of $C_L(\alpha)$ and $C_D(\alpha)$ for the PX-31 would require new CFD simulations or wind tunnel tests for the relevant airfoil.

3.3 Propulsion force

The UAVs considered in this work are all driven by a single electric motor (EM) powered by a battery, a generator or a combination of the two. Attached to the EM is a propeller, which produces the desired thrust at a given moment. The thrust stems from the pressure differential created between the two sides of the propeller, similar to how the wings create the lift of the vehicle.

3.3.1 Propeller efficiency

At the centre of propeller design is *propeller efficiency*, a measure of the propulsive power generated by the propeller given an amount applied by the EM:

$$\eta_{prop} = \frac{P_{propulsive}}{P_{shaft}} \tag{3.3.1}$$

where P_{prop} is the generated propulsive power and P_{shaft} is the power applied to the motor shaft. Propeller efficiency is typically dependent on multiple factors such as the angle of attack of both the aircraft and the propellers themselves, as well as airspeed. However, in this work, η_{prop} will be considered static.

3.3.2 Required power

Essential to the evaluation of a planned path is the required propulsive power at each discretisation step. This, combined with the length of time of a step indicates the amount of energy being used at every section of the path. By assuming the lift force is high enough to counteract the weight of aircraft, it is just required to provide sufficient power to overcome the drag component and propel the aircraft at the desired airspeed. Thus, the required propulsive power in W is proportional to the required thrust and the desired airspeed, expressed as:

$$P_{prop}(T_{prop}, V_a) = T_{prop}V_a \tag{3.3.2}$$

Desired thrust is then determined by the sum of the drag force and the horizontal component of the weight force seen in the body frame. The required thrust T_{prop} in Newtons is thus:

$$T_{prop}(F_D, \theta) = F_{drag} + mg \cdot sin(\theta) \tag{3.3.3}$$

where θ is determined from Equation 3.1.9. During descent, θ will generally be negative, implying a lower required thrust due to assistance from the gravity. Further, by inspecting Equation 3.1.8 we can see what vertical wind can do for the thrust requirements. Sufficiently positive upwards wind (negative w_d) can cause a negative γ_a despite a positive climbing angle γ. This implies that despite climbing in relation to ground, the UAV actually descends in the relative air, which eases thrust requirements. Further assuming perfect power transmission from the battery or the rectifier to the EM, the expended work in Ws becomes:

$$W_s = P_{EM} \cdot t \tag{3.3.4}$$

for some duration t in seconds. This model assumes constant required power over the time step, which is a necessary simplification due to the size of the calculations and length of the path. In the case of negative thrust requirements (i.e. $T_{prop} < 0$), power requirements are assumed to be zero. In such a case, only the icing protection system accounts for energy expenditure. There are UAV configurations that can spin the propeller in reverse to generate power, but that is not assumed to be the case here.

3.4 Icing protection system

As the operational domain in the simulations of this work concern icing conditions, the UAV is equipped with an icing protection system (IPS), as seen in [22], [23]. IPSs typically work by detecting icing conditions during operations and airframe icing, and then mitigating this either by preemptive measures called anti-icing, or by de-icing the wings after the fact. This is done by activating electro-thermal panels installed on the airframe either melting the ice, or preventing ice from forming. When implementing a model of an IPS into a discretised simulation environment, two factors play a major role; power requirements for the electro-thermal system, and the degraded drag-coefficient, should the IPS allow temporary airframe icing. The power loads for the two ice mitigation strategies are using a model adapted from [24], and follow relation:

$$P_{ips}(V_a, LWC_c, T) \tag{3.4.1}$$

where V_a is the absolute airspeed in m/s, LWC_c is in g/m^3 and temperature T in ℃. The proposed explicit models in kW are:

$$\begin{aligned} P_{deice} = (-0.7551 \cdot T - 0.1122) \cdot (0.0211 \cdot V_a + 0.4722) \cdot (0.1211 \cdot LWC_c + 0.9596) \\ \cdot (1.3277 - 1.0366 \cdot (1 - e^{0.3260 \cdot T})) \cdot A \end{aligned} \tag{3.4.2}$$

for the de-icing power load and:

$$P_{antiice} = (-0.7551 \cdot T - 0.1122) \cdot (0.0211 \cdot V_a + 0.4722) \cdot (0.1211 \cdot LWC + 0.9596) \cdot A \tag{3.4.3}$$

for the anti-icing power load. A is the active heat surface area, which in this work will be set to 0.05 m times a wingspan of 2.1 m. Figure 3.2 is a graph comparing de-icing and anti-icing power loads under various configurations. Here, the LWC is considered constant at 0.4 g/m^3. The experimental data creating the models was originally only gathered between 0℃ and -10℃, so power consumptions for colder temperatures than that are assumptions. The graph plots the expected power consumption between -20℃ and 0℃, which encompasses most practical icing temperatures one can expect in UAV operations.

Work surrounding [5] generated an estimation for performance penalties when experiencing airframe icing through the simulation tool FENSAP-ICE, intended to provide a performance penalty for allowing the de-icing strategy. Following determination of the drag coefficient as explained in Section 3.2.1, C_D is further altered when using de-icing through relation:

$$C_D(C_D^0, LWC_c) = C_D^0 \cdot (0.0785 \cdot LWC_c + 1.4973) \tag{3.4.4}$$

where LWC_c is the liquid water content discussed in section 2.2.2. To decide which IPS strategy to use in a given step, the algorithm compares the power consumption from $P_{antiice}$ with a clean C_D^0 to using P_{deice} with a degraded C_D, picking whichever has the lowest consumption.

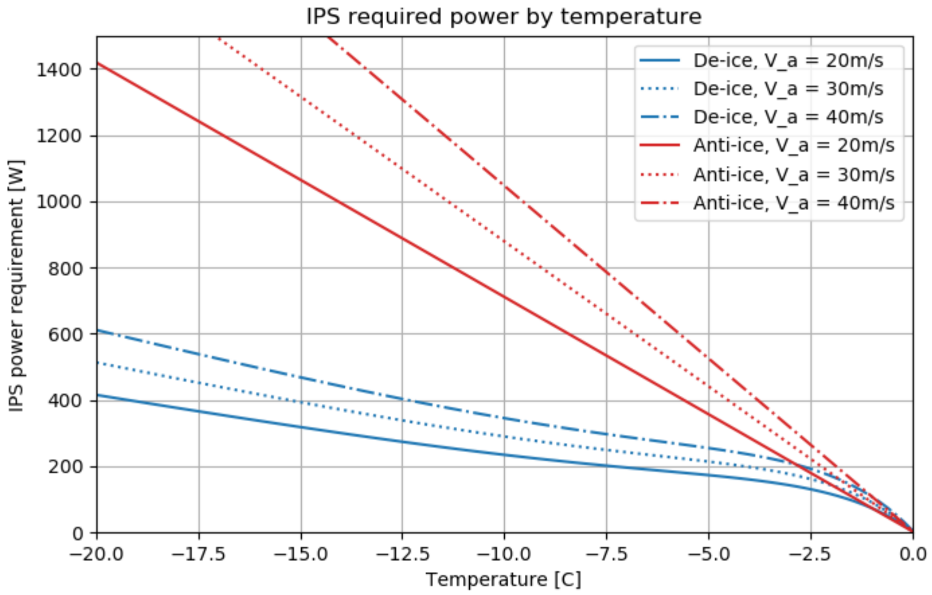

Figure 3.2: Power requirements for IPS under different conditions, $LWC_c = 0.4$ g/m^3

3.5 Unfettered airframe icing

As discussed in Section 2.2, airframe icing occurs as a result of being exposed to supercooled liquid droplets. The collision of a water droplet with an airframe breaks the surface tension of the droplet causing the contained supercooled water to freeze. Allowing ice to form on an airframe increases the drag force, reduces the generated lift and makes the aircraft more susceptible to stalling. Both to demonstrate why an IPS is desirable in long-distance UAV operations in icing conditions, and to explore alternatives, a model of cumulative performance degredation as a result of airframe icing is deployed.

Figure 3.3 shows the shape of accumulated ice after 20 minutes in icing conditions, under different LWC values. As seen in the figure, icing for LWC < 0.01 g/m^3 can be considered negligible, which it will in this work. As ice horns form on the airfoil, the weight of the ice formed will eventually become substantial enough to have an effect on aircraft performance.

3.5.1 Ice accumulation

Ice forms on an airframe with different shape and appearance based on the droplet diameter, temperature and LWC of the surrounding area. Engine icing, such as ice forming on internal surfaces of the carburetor and hindering air supply to the engine, will not be considered in this work. The three main categories of structural icing are *rime ice*, *glaze ice* and *mixed ice* [15]. Rime ice occurs in colder temperatures with smaller droplets, when the water forms an opaque, even coating of ice around the airfoil. Glaze ice, or clear ice, forms in higher temperatures, when parts of the liquid water striking the airfoil flows backwards along the structure, forming dense, uneven bumps. Glaze ice has a greater effect on airflow than rime ice, making it the most hazardous category of the two. Mixed ice is a combination of rime and glaze ice.

Estimates of expected icing can be acquired using computational fluid dynamics simula-

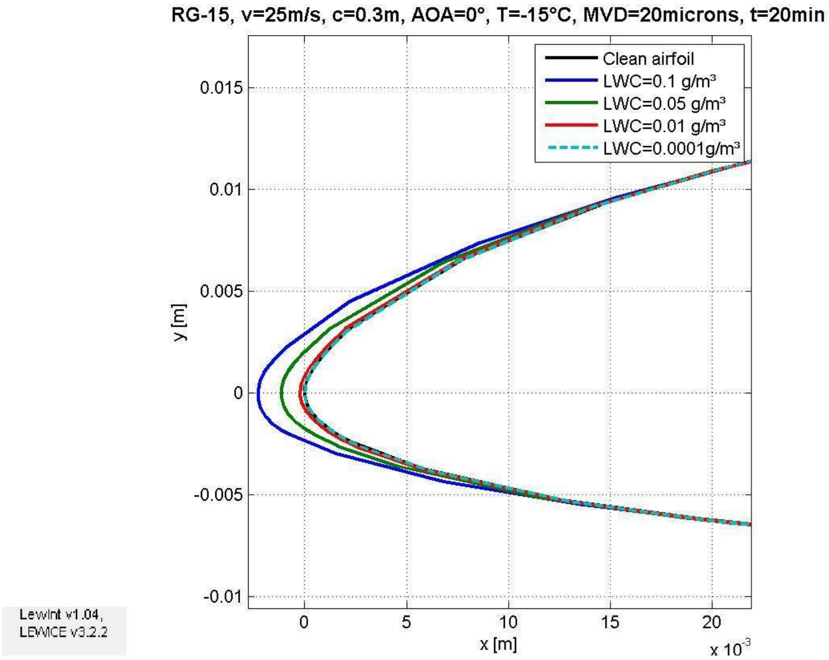

Figure 3.3: Simulated ice horns after 20 minutes of exposure to different LWC, courtesy of Richard Hann

tions. A tool such as ANSYS FENSAP-ICE can be used to simulate water dispersion and icing on a geometric model of an airfoil. This can then be used to simulate performance degradations and total ice accretion. All data and models presented here are adapted from [18].

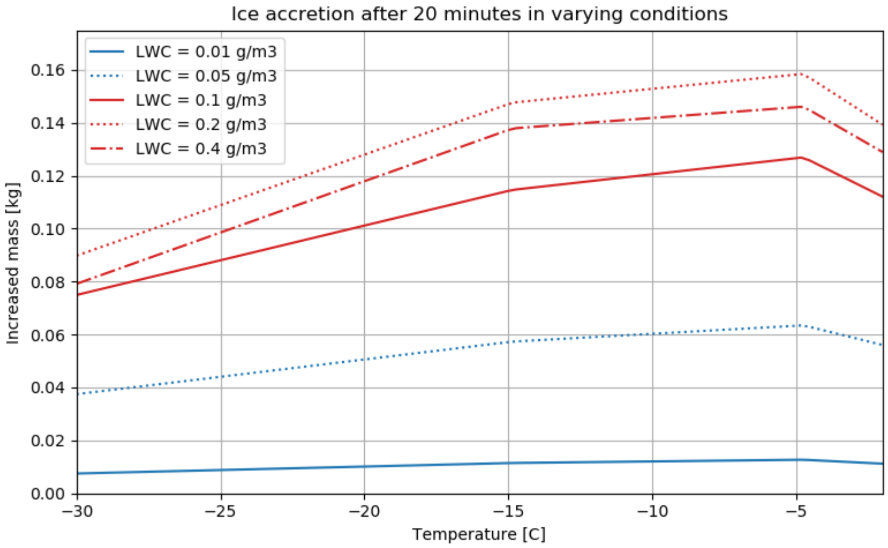

Figure 3.4: Weight of ice accretion after 20 minutes, adapted from [18]

The original model gives estimates in added ice mass kg/m on an RG-15 airfoil. This can be converted to the relevant airfoil of the PX-31 (presented in Section 5.4) by dividing the estimate by the aerodynamic chord of the RG-15 of 0.450 m and multiplying by the correct chord of 0.275 m. Figure 3.4 displays estimates of the total added mass to the airfoil of the UAV platform when multiplied by the wingspan of the PX-31 of 2.1 m. All values are estimates after 20 minutes in a given condition, and are considered to scale linearly for durations shorter and longer than this. The model is initially generated from MVD values of 15 µm and over, and is thus scaled linearly for LWC values from 0.01 g/m^3 up to the LWC value corresponding to this MVD threshold, following the relations presented in Section 2.2.2.

3.5.2 Aerodynamic performance degradation

In addition to the added weight from ice on the airframe when operating in icing conditions, the aerodynamic performance of the UAV changes. Generally, the drag coefficient C_D increases while the lift coefficient C_L decreases, their respective magnitude dependent on atmospheric variations, time spent in the conditions and on the angle of attack at the time of interest. Degradations occur by the shifting of curves relating the angle of attack to lift and drag coefficients.

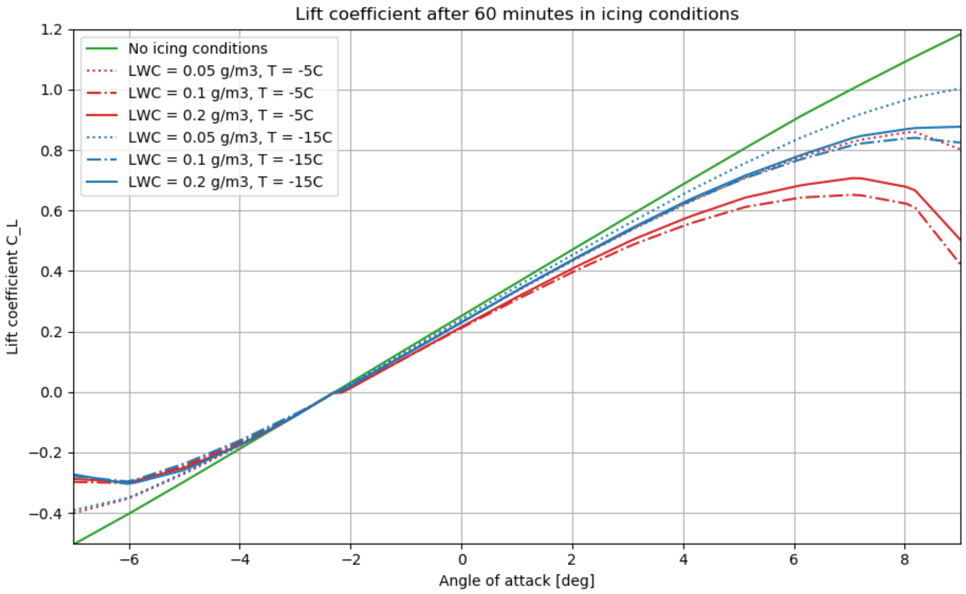

Figure 3.5: Lift coefficient C_L after 60 minutes of icing conditions, adapted from [18]

Figure 3.5 shows C_L as a function of angle of attack after experiencing icing conditions for 60 minutes. The weather parameters, LWC and temperature, are considered constant throughout the icing process. Note the deviation from C_L is larger the higher (or lower under zero) the angle of attack is. As an example of the implications of this model during actual flight, consider a UAV after experiencing an average of 0.2 g/m^3 LWC and -5℃ for 60 minutes. If it is, through Equation 3.2.3, determined that a step requires a C_L of 0.6, what would normally require an AOA of 3.2° now require an AOA of roughly 4.3°. The drag force thus equals that corresponding to a higher AOA, significantly increasing the drag to overcome.

Figure 3.6 displays the same relation for the drag coefficient C_D under similar conditions. Note the significant increase in C_D for all angle of attacks, compared to C_L, seeing almost

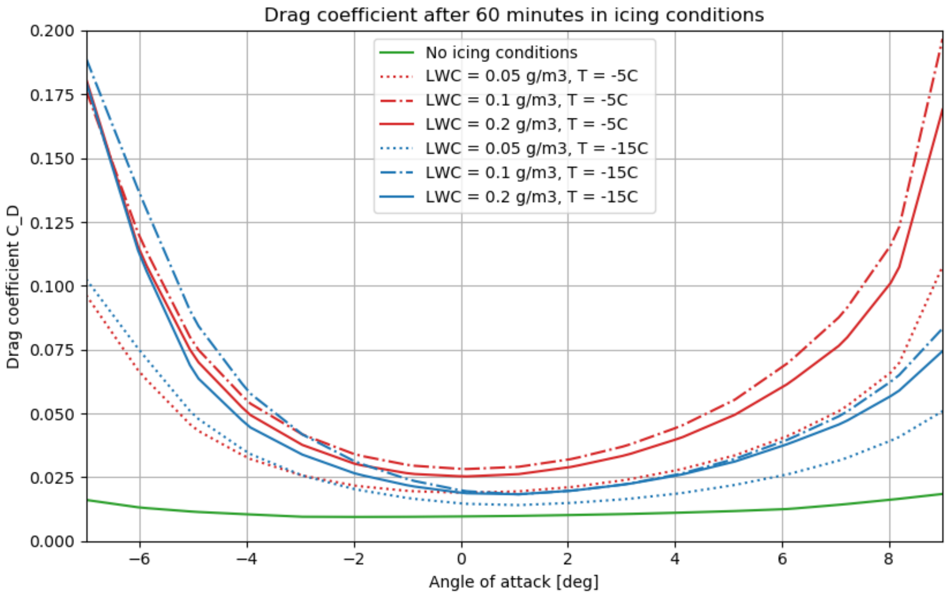

Figure 3.6: Drag coefficient C_D after 60 minutes of icing conditions, adapted from [18]

a threefold increase of C_D under even low AOAs. Also of note is that the degradations, both for C_L and C_D, are significantly larger with temperatures closer to the freezing point, which is emblematic of the formation of glaze ice, discussed above. Inspecting Equation 3.2.2, we can see that such a large increase in C_D causes an enormous increase in the drag force the UAV needs to overcome, which has indications of just how critical sustained airframe icing can be to the efficiency of a mission.

Originally simulation values for the coefficients are provided for a range of integer angle of attacks under different conditions for MVD (converted to LWC in this adaptation) and temperature. Continuous values for C_D and C_L were obtained through trilinear interpolation on the ensuing three-dimensional matrix. The process of such an interpolation can be seen in Appendix A.2.

As the coefficient curves shift in different manners for different weather conditions, and needs to be calculated for all angle of attacks, there is no good way of "summing" degradations. To avoid storing every weather condition for every step with icing conditions to calculate a degraded coefficient at every step, the LWC_c and temperature is averaged out for all icing steps up to the given point. This is done using a cumulative moving average for the relevant weather conditions, for example in the case of the accumulated average temperature at step i T_i^{avg}:

$$T_i^{avg} = \frac{t_i \cdot T_i + T_i^{avg,prev} \cdot \sum_{j=0}^{i-1} t_j}{t_i + \sum_{j=0}^{i-1} t_j} \tag{3.5.1}$$

where T_i is the temperature at step i, and $t_j = 0$ for all j where icing is not present. The same operation is done for the LWC.

4 Hybrid electric powertrain

The following section will give an overview of the different design characteristics a hybrid electric powertrain might have, and the governing equations that will be used in the simulation environment. Included is also a model of the battery intended to be used in this work. Hybrid electric power systems (HEPS) in UAVs are a relatively new concept, being pioneered by the Diamond DA36 E-Star in 2011 [25]. Motivation surrounding the development of HEPS for UAVs is primarily increasing operational range without sacrificing the benefits of a smaller aircraft design. As the operational limitations of the vehicles are determined in large by the weight and available power, specific energy density of the power sources arise as an important factor. Specific energy density refers to the energy per unit mass of a medium, and is typically denoted by Wh/kg.

4.1 Internal combustion engines

Most HEPS use a fuel tank feeding an internal combustion engine (ICE) connected either mechanically to the drivetrain, or its generated power is converted to electrical energy through a generator. ICE based HEPS typically get their energy from gasoline, having a specific energy of around 13 kWh/kg [26]. Much of this energy will be lost during operation due to inefficiencies in coupling, thermal losses and combustion, but one can still expect significant amounts of energy from relatively little amounts of fuel.

Some hybrid configurations can suffer from not being able to operate the ICE in the optimal region. This typically occurs mostly in HEPS with direct transmission coupling, meaning mechanical power is directly applied from the ICE to the propulsion system. Most of this can be mitigated through using for instance continuously variable transmission, but some losses will occur through the ensuing clutch system.

4.2 Fuel cells

An alternative to ICE in the HEPS is a fuel cell. Fuel cells convert chemical energy of a fuel and an oxidiser into electrical energy. Using a fuel source such as hydrogen would produce no emissions, as well as being silent compared to a combustion engine. Hybrid fuel cells for use in UAVs can provide significant advantages in terms of energy densities over batteries [27]. The specific energy of Hydrogen is 39.4 kWh/kg [28], significantly more than that of gasoline and LiPo batteries. Typically, an efficiency of up to 60% can be expected from a fuel cell [29], giving it a significant advantage to that of a gasoline based ICE.

There are potential issues with a fuel cell configuration such as the dependency on temperature, where variations in temperature significantly affect operations, and many fuel cells are not rated to operate in freezing temperature [30]. Another potential drawback is the price point, where a 1kW hydrogen fuel cell can cost upwards of 10.000$, which could effectively double the production cost of a mid-range, mid-sized commercial UAV. For these reasons, an ICE configuration will be considered for the rest of the thesis.

4.3 Hybrid electric configurations

Hybrid electric powertrains come in multiple configurations with advantages and disadvantages for each, but generally consist of many of the same components. Choice in configuration typically depends on design space, weight limitations, power demands and desired efficiency.

4.3.1 Series powertrain

Series hybrid powertrains are distinguished by that the propeller is driven purely by the electric motor, as seen in Figure 4.1. The ICE is fed by a fuel tank and produces torque which a generator converts to electrical power. This can directly power the electrical motor as well as charge the battery with the excess power during cruise. Should the EM desire more power than the generator produces, the battery supplies the rest.

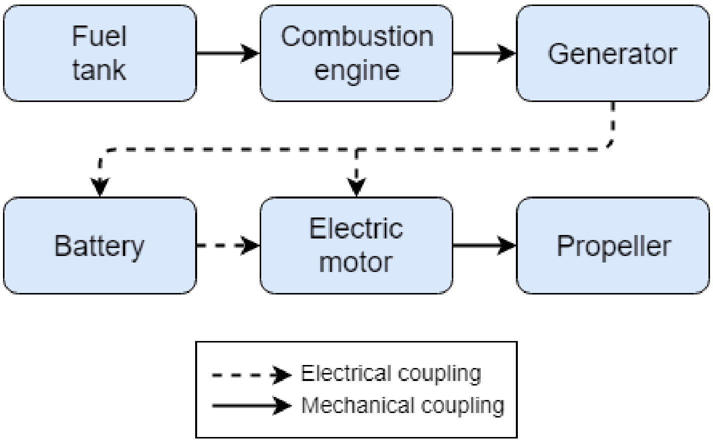

Figure 4.1: Hybrid series configuration

The primary advantage of the series hybrid configuration is the complete decoupling of the ICE from the mechanical drivetrain, which allows the engine to be run with optimal loads [11]. Motor control for this configuration is rather simple, but it comes a significant weight penalty due to the inclusion of a generator [14].

4.3.2 Parallel powertrain

Hybrid parallel powertrain configurations are distinguished in that both the ICE and the EM provide mechanical power to the propeller as seen in Figure 4.2. The power is coupled mechanically through a clutch to which the propeller is connected.

When the ICE contributes more power than the propulsion controller requires the clutch reroutes the excess mechanical power to the EM which in turn works as a generator to charge the battery. Should the ICE fail, or the controller prefers to use the EM, the battery can power the entire drivetrain. This design is highly beneficial in terms of redundancy, as well as relative lightness due to the lack of a generator for the ICE. An issue with the design is the flexible load demands issued on the ICE, which can in turn lead to sub-optimal motor efficiency.

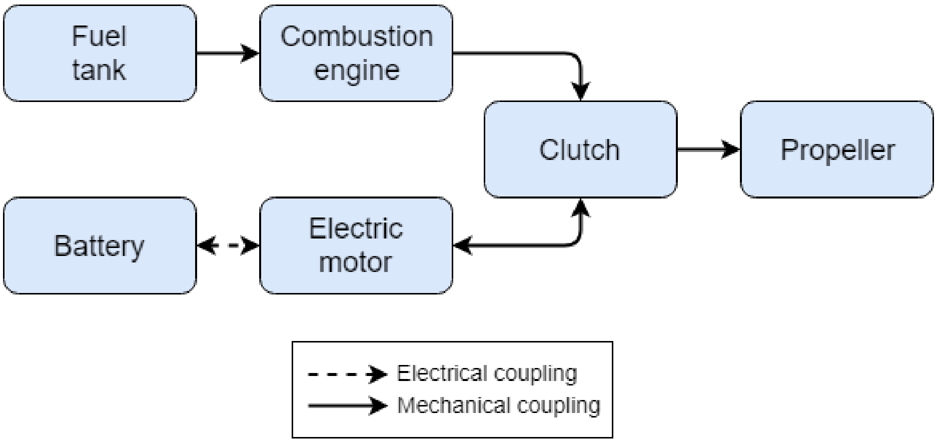

Figure 4.2: Hybrid parallel configuration

4.3.3 Series parallel powertrain

Somewhere between the two aforementioned configurations falls the series parallel powertrain. Here the ICE and the EM are mechanically separated as in the parallel configuration, but they are coupled in a planetary gear which drives the propeller as well as a generator that can charge the battery, as seen in Figure 4.3.

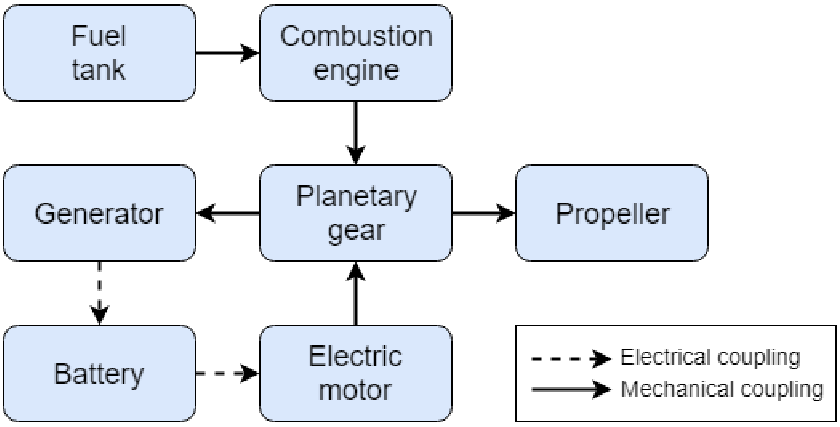

Figure 4.3: Hybrid series parallel configuration

The ICE can thus be driven in its optimum operating point, which can lead to gains in terms of efficiency over the parallel configuration. However, it is worth noting that the planetary gear and the generator adds a weight penalty over said structure.

4.4 Batteries

Batteries have become commonplace in UAV development. Not only are there multiple advantages over pure combustion powertrains in terms of emissions and operational noise, but also that of potential power. Although batteries are lacking in terms of energy density compared to aircraft fuel, they make up for it in *specific power*, which is the ratio of available power to the weight of a power source in W/kg and *power density* which is the ratio of power to volume in W/m^3. In practice, this opens up options for the miniaturisation of the vehicles, as smaller micro aerial vehicles (MAV) with limited available weight and design space can still supply sufficient power to an EM. Lithium polymer (LiPo) battery cells are the most commonly used in UAV applications, with a specific power of up to 430 W/kg and a specific energy density of up to 0.22 kWh/l, the highest of the common battery types. In this work the dynamic model of the battery is based on the Tremblay's model from [12] and supplemented by [31]. Define:

$$\begin{aligned} A &= V_{full} - V_{exp} \\ B &= \frac{3}{C_{exp}} \\ C_r &= C_{cut} - C \end{aligned} \tag{4.4.1}$$

where V_{full} is the voltage at maximum capacity, V_{exp} is the voltage at at the end of the exponential range of the battery, C is the discharged capacity in Ah, C_{cut} is the discharged capacity at the cut-off limit in Ah. Thus, the battery potential V_{oc} in V can be represented by:

$$V_{oc}(C, V_0) = V_0 - \left(\frac{\kappa C_{cut}}{C_r}\right) + Ae^{-BC} \tag{4.4.2}$$

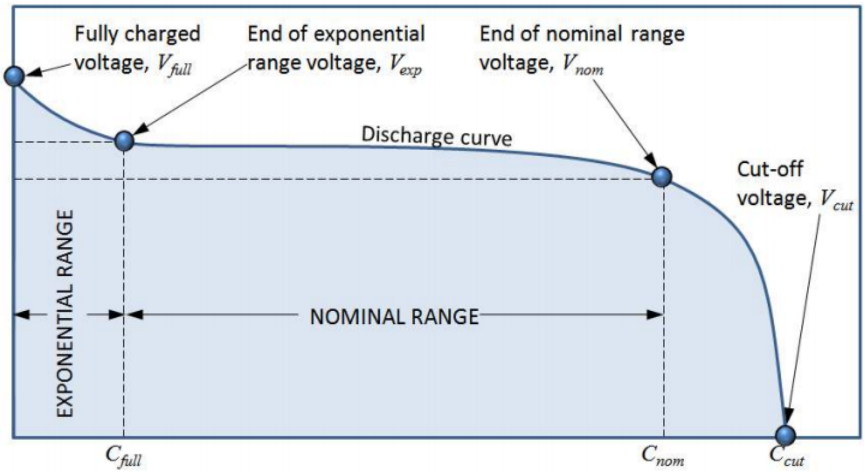

Figure 4.4: Battery discharge curve, from: [31]

The relation is illustrated in Figure 4.4. V_0 is defined as:

$$V_0 = V_{full} + \kappa + (R_c \cdot I_{eff}) - A \tag{4.4.3}$$

where R_c is the constant internal resistance in the battery in Ω and I_{eff} is the effective current in A given by:

$$I_{eff} = \frac{P_{eff}}{V_{oc}} \tag{4.4.4}$$

Using I_{eff} as a variable rather than the constant specified discharge current as in the regular Tremblay model. This is due to the power consumption being considered constant during a discretisation step. Further, κ is the polarisation voltage in V given by:

$$\kappa = \frac{\left(V_{full} - V_{nom} + A(e^{-BC_{nom}} - 1)\right)(C_{cut} - C_{nom})}{C_{nom}} \tag{4.4.5}$$

Thus, a nonlinear equation representing V_{oc} is obtained as:

$$V_{oc}^{n+1} - \left(V_{full} + \kappa - A - \frac{\kappa C_{cut}}{C_r} + Ae^{-BC}\right) V_{oc}^{n} - R_c I_{rated}^{1-n} P_{eff}^{n} = 0 \tag{4.4.6}$$

where n is the Peukert's constant for the specific battery, I_{rated} is the rated maximum discharge current for the battery in A. The equation can be solved with any simple nonlinear solver. Among the assumptions for this model are:

- Constant internal resistance
- Charging and discharging characteristics are each other's reciprocal
- No temperature effects
- No self-discharge
- No memory effects or cycle life

Some of these assumptions could be worth looking into removing, especially the temperature effects, considering the primary operational environment. However, this would require some robust temperature dissipation models between the interior of the UAV and the external environment combined with heat production in the batteries and the ICE/generator.

5 Simulation environment

A simulation environment for the meteorological data, UAV model, mission parameters and optimisation algorithm is needed to quantify the potential energy and time-wise improvements that can be made. This section will explain the method of optimisation, structure of the code base, the used resources and specify the platform configuration for the simulations.

5.1 Particle swarm optimisation

For this work, the optimisation algorithm is based on the *particle swarm optimisation* (PSO) method from [32]. PSO is based on the concept of *swarm intelligence* which is accumulating knowledge or learning from decentralised, self-organised systems [33]. Generally, the optimisation works by randomising a population of *particles*, which is a quantifiable candidate solution. These particles move iteratively and semi-randomly through the viable search-space through a changing *particle velocity* which indicates which direction a better solution might lie.

The aforementioned particles are in essence relatively simple suggestions of a potential path, designed to seed a more detailed solution. Each particle contains separate vectors of waypoint positions in an ENU frame, desired airspeeds and climbing angles. Each particle has an associated particle velocity which has vectors of similar sizes and physical properties, but these values instead indicate how much each of the particle vectors should be updated in an iteration.

After initialising the world, mission and UAV model, the PSO algorithm is started. To begin, a generic baseline solution is generated through a relatively hard coded particle. In this initial solution, the aircraft begins by climbing up to a feasible height, then maintains this height flying straight towards the desired ending point at a constant cruise velocity before descending to the appropriate final altitude. The reasons for this baseline solution is twofold. It yields a viable initial solution to push the particles towards feasible solutions in the first very random steps, and it serves as a quantifiable measure of improvement after the optimisation is complete.

Further, a number of particles (given by a set population size) are randomly generated within the confounds of the mission space, and their associated solutions and subsequently costs are calculated. All property vectors of all particles (including the initial generic solution) are then updated every iteration with a change given by the associated particle velocities *vel*:

$$\begin{aligned} W_x &= W_x + vel.W_x \\ W_y &= W_y + vel.W_y \\ A &= A + vel.A \\ \Gamma &= \Gamma + vel.\Gamma \end{aligned} \tag{5.1.1}$$

where W_x and W_y are the x and y components of ENU waypoints W, A is the vector of airspeeds and Γ is the vector of climbing angles. After an update, the particles are clamped to fit within the predefined confounds of the state-space. Prior to the above operation, the particle velocities are updated by:

$$\begin{aligned}
vel.W_x &= w \cdot vel.W_x + c1 \cdot r_1 \cdot (W_{x,cb} - W_x) + c2 \cdot r_2 \cdot (W_{x,gb} - W_x) \\
vel.W_y &= w \cdot vel.W_y + c1 \cdot r_1 \cdot (W_{y,cb} - W_y) + c2 \cdot r_2 \cdot (W_{y,gb} - W_y) \\
vel.A &= w \cdot vel.A + c1 \cdot r_1 \cdot (A_{cb} - A) + c2 \cdot r_2 \cdot (A_{gb} - A) \\
vel.\Gamma &= w \cdot vel.\Gamma + c1 \cdot r_1 \cdot (\Gamma_{cb} - \Gamma) + c2 \cdot r_2 \cdot (\Gamma_{gb} - \Gamma)
\end{aligned} \tag{5.1.2}$$

Equation 5.1.2 indicates changes in velocity based on current states as well as local and global bests. Vectors with subscript cb or gb indicate the values of the best local and global particle respectively. Local best means the best value the current particle has been historically while global best is the single best particle value that any solution has been valued at. $c1$ and $c2$ are tuneable parameters that change how desirable it is to learn from local and global bests respectively while r_1 and r_2 are randomly generated positive vectors of appropriate dimensions, and are regenerated every iteration for every particle. Finally, w indicates the inertial weight of the particle, which affects how aggressively the particle should move in its search space. The initial inertial value is set to 1, encouraging larger searches, but w then linearly decreases through the iterations, to let all particles converge towards well-established solutions and focus the search more locally, defined by:

$$w_i = \frac{w_{initial}(maxIt - i)}{maxIt} \tag{5.1.3}$$

where w_i is the inertial weight at iteration i, $w_{initial}$ is the initial inertial weight and $maxIt$ is the maximum number of iterations in the PSO algorithm.

If a particle reaches the pre-defined spatial domain-limits a *velocity mirroring* is performed. This is done by flipping the particle velocity vector to face the opposite direction, which would imply the particle is moving away from the limit it just reached and back into the appropriate search space.

The optimisation variables have different limits imposed on them. Restrictions on the spatial domain in which the algorithm is allowed to search is important to ensure convergence towards a feasible optimal solution. These limits are sat in the ENU frame to impose restrictions on the allowed values of the optimisation variables xx and yy. The limits are defined by:

$$\begin{aligned}
x_{min} &= min(startENU.x, stopENU.x) - L/5 \\
x_{max} &= max(startENU.x, stopENU.x) + L/5 \\
y_{min} &= min(startENU.y, stopENU.y) - L/5 \\
y_{max} &= max(startENU.y, stopENU.y) + L/5
\end{aligned} \tag{5.1.4}$$

where L is the absolute distance between the two points $startENU$ and $stopENU$ defined by:

$$L = \sqrt{(startENU.x - stopENU.x)^2 + (startENU.y - stopENU.y)^2} \tag{5.1.5}$$

When a candidate solution goes outside of this area, a velocity mirroring is performed. Dividing L by 5 in Equation 5.1.4 is an arbitrary and tunable choice.

Limits on the climbing angle γ are dependant on the flight dynamics of the UAV, and are often specified by the manufacturer in climbing rate over time. Here, the maximum and minimum climbing angle is set to 10° and -10° respectively. A justification for these values, as well as limits on the airspeed, is given in Section 5.4.

5.2 Cost functions

To realise the optimisation algorithm, a proper cost function is needed. The purpose of the cost function is to indicate the successfulness of a given solution, and to compare multiple solutions and determine the best one. Infeasible solutions are given infinite cost the moment they are deemed infeasible, and are therefore not possible for the optimisation algorithm to learn from. Solutions are deemed infeasible for infractions such as breaching the highest set altitude (in this case 2300 m), going below the topography at any point or reaching the cut-off capacity in the battery.

For energy optimisation, a measure of expended energy is required, preferably based on the cumulative remaining energy both in terms of battery capacity, and of fuel. The cost function is, with that in mind, expressed as a measure of spent energy in Ah:

$$COST_{energy} = C_{cut} - C_{end} + \frac{2(F_{full} - F_{rem})\eta_{tot}\rho_{fuel}E_{fuel}}{V_{exp} + V_{nom}} \tag{5.2.1}$$

where C_{cut} is the maximum discharged capacity from the battery, C_{end} is the remaining battery capacity at the end of the mission, F_{full} is the initial fuel level, F_{rem} is the remaining fuel level, η_{tot} is the total efficiency in the system, ρ_{fuel} is the density of fuel, E_{fuel} is the total energy in fuel per kg and V_{exp} and V_{nom} is the voltage at the end of the exponential range and the nominal range respectively. In a physical sense, the final term in Equation 5.2.1 expresses how much battery capacity one could expect to regenerate with the remaining fuel given an average nominal battery voltage.

In some situations, it is desirable to optimise the time of mission completion rather than expended energy. The cost function is thus simply the sum of each time step through a solution:

$$COST_{time} = \sum_{i=0}^{N-1} t_i \tag{5.2.2}$$

where t_i is the time spent at step i and N is the total number of discretisation steps in a single solution. A further complication to this cost function could be to penalise energy expenditure close to the cut-off value, so that one is guaranteed a solution with enough remaining capacity to perform for instance landing maneuvers. However, through simulations, this has not been deemed necessary as all time optimisations has caused reduced energy expenditure as will be shown in Section 7.

5.3 Algorithm structure

The system was constructed in an object-oriented manner in C++ for reasons of running time, modularity and accessibility in structure. A flow chart of communications and structure of the system can be seen in Figure 5.1.

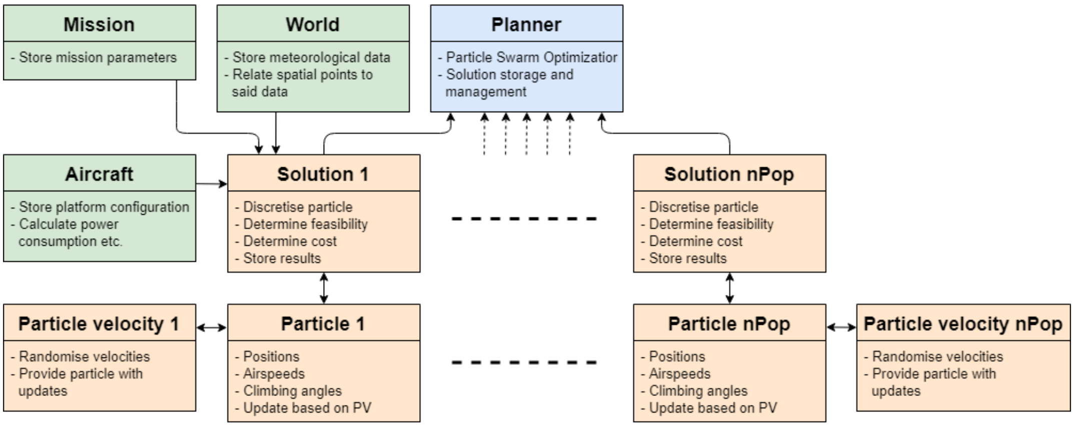

Figure 5.1: Software structure

Planner is the central module and can be considered the "main" function of the system. From Planner, a *Mission* is created with specified parameters. Further it spawns a *World* object, which reads the relevant meteorological data from a folder and stores this in spatial matrix form with read-only member functions to extract the data at run-time. An *Aircraft* model is created, which keeps track of platform parameters such as physical constraints, powertrain specifications and functions to calculate power requirements and energy consumption at given time intervals.

Planner then initialises as many *Solutions* as the specified population size, each with their own *Particle*. Solutions keep track of simulation results for each Particle at a single iteration, through information such as remaining fuel, battery capacity, altitudes, time expenditure etc. Particles have specified ENU-frame positions, velocities and climbing angles, which are updated every iteration based on its own *Particle Velocity*, as explained in Section 5.1. A copy of the globally best particle and solution, as well as a list of the historically best particles for each member in the population, is kept track of in Planner to properly update the Particle Velocities. All simulations are run with 256 unique particles and 256 iterations.

Not illustrated in the flow chart are the multiple supporting functionalities, such as coordinate frame conversions and helper functions.

5.4 Drone platform

In these simulations, the UAV specifications are based on the *PX-31*, pictured in Figure 5.2. The PX-31, developed by the Norwegian company *Maritime Robotics* [34], was designed with winter and arctic climates in mind, making it highly suited for this application. Its most important parameter specifications are:

- Empty weight: 15.0 kg
- Max take-off weight (MTOW): 22.0 kg
- Wing surface area: 0.81 m^2
- Vcruise: 25 m/s
- Vmax: 40 m/s

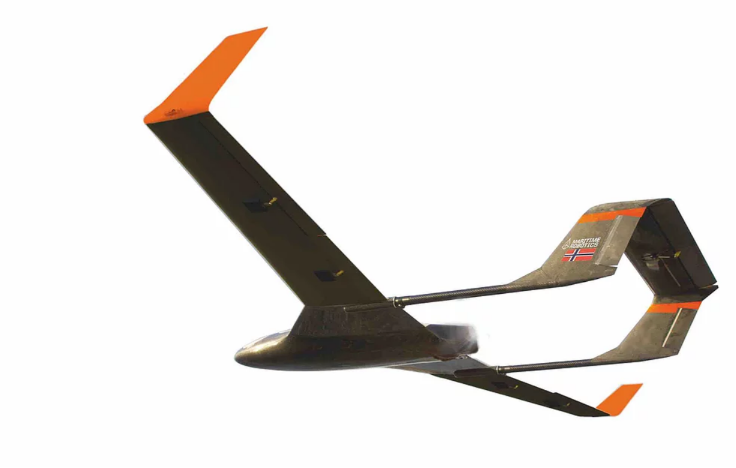

Figure 5.2: Maritime Robotics' PX-31 UAV platform

- Vstall: 14 m/s
- Max climb rate: 1200 fpm

As these parameters are based on a fully electric configuration there's a remaining capacity of 7.0 kg to the hybrid powertrain, fuel and payload. The changes in design space this subsystem might require and the aerodynamic performance changes that will cause is outside the scope of this work. Calculating propeller efficiency is a difficult procedure based on many variables. In these simulations the propeller efficiency will be simplified to a constant $\eta_{prop} = 0.5$. The maximum climbing rate of 1200 fpm (feet per minute) corresponds to 6.096 m/s. Assuming this climbing rate occurs when the airspeed equals Vmax of 40 m/s, we can separate into a maximum horizontal airspeed of 33.904 m/s and a vertical airspeed of 6.096 m/s. In no wind (such that ground speed equals airspeed) this corresponds to a climbing angle of $\arctan(\frac{6.096}{33.904}) = 10.19°$, hence the climbing angle limitation discussed in Section 5.1 of ±10°.

5.4.1 HEPS configuration

The HEPS used in this work is a variant on the series hybrid configuration described in Section 4, where there is no mechanical coupling between the combustion powertrain and the battery, as illustrated in Figure 5.3. Instead, a brushless AC-motor is driven in reverse by the ICE to generate electricity. A rectifier converts the alternating current to direct current, which can drive the EM and subsequently the propeller, or charge the battery.

The ICE is a carburetor engine such as the one from UAV Factory's *Penguin B* UAV. It is a 2,5 hp engine, corresponding to a maximum output power of 1860 W. Desired output power from the generator is around 1 kW to drive the propeller and slowly charge the battery during cruise in normal conditions. An efficiency in the generator of $\eta_{gen} = 80\%$, and the efficiency in the ICE of $\eta_{ice} = 15\%$ would correspond to a fuel expenditure of around 0.65 kg/h based on the specific energy of fuel from Section 4.1. This would give the entire combustion powertrain an efficiency of $\eta_{gen} \cdot \eta_{ice} = 12\%$, which is reasonable.

It is worth noting more complex models for combustion and power generation processes exist, but due to the large discretisation steps in the algorithms of this work, they can be considered relatively constant and controllable, as complex behaviours are likely to

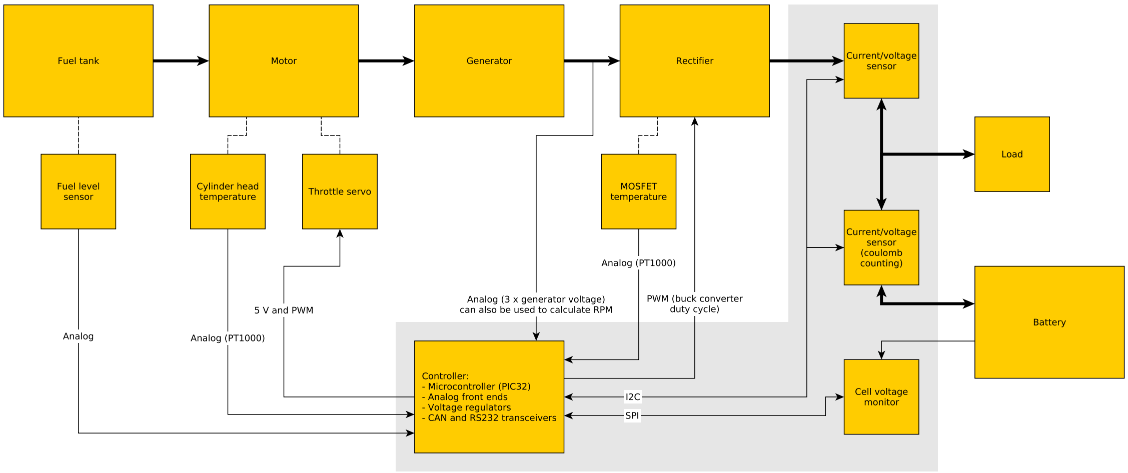

Figure 5.3: PX-31 hybrid controller and powertrain

disappear over longer horizons. This is further justified by the structure of the HEPS, in that it can be driven separately from the EM and the battery, and thus be ran at a constant, ideal operating point as discussed in Section 4.3.

Using a 4000 cc fuel tank, corresponding to around 3 kg of fuel at take-off plus an additional 0.5 kg for the actual weight of the tank as well as 2 kg for the ICE and generator would leave the total weight of the platform 20.5 kg, leaving 1.5 kg for the payload.

The battery is modelled as described as in Section 4.4, with specific parameters specified below. The battery is charged and discharged in the same manner, so the battery capacity at time step i+1 is given by:

$$C_{i+1} = C_i - \dot{C}_i(P_i - P_{gen}, C_i) \cdot t_i \tag{5.4.1}$$

where P_i is the required power at step i, P_{gen} is the generated power, t_i is the length of the time step and $\dot{C}$ is a function describing rate of change of capacity at a certain voltage given a supply or demand in power. $\dot{C}(P, C)$ is designed such that:

$$\dot{C}(P, C) = -\dot{C}(-P, C) \tag{5.4.2}$$

which implies that the remaining capacity of the vehicle increases if $P_{gen} > P_i$ in Equation 5.4.1. $\dot{C}_i$ is defined by:

$$\dot{C}_i = \frac{I_i}{3600} = \frac{P_{toti}}{3600 \cdot V_{oci}} \tag{5.4.3}$$

where P_{toti} is the total power drawn at step i and V_{oci} is the battery potential voltage at step i.

In terms of control strategy for when to use the HEPS, a simple rule-based approach is used here. When the remaining battery capacity hits C_{nom} the HEPS will turn on, and remain on until the battery capacity hits C_{exp} at which point it will turn off. This is due to the nominal battery range being easier to charge, i.e. less generated power equals more gained battery capacity than in the exponential ranges. In other words, the current HEPS control strategy is waiting to charge the battery until the battery capacity is sufficiently low. A potential issue to this approach is if the low battery capacity coincides with an area requiring significantly more power consumption than nominal, such as a stretch of icing conditions. This may render the HEPS unable to recharge the battery at all, which might drain the battery and cause a system failure. A mitigation to this would be to ensure maximum battery capacity when the UAV hits icing conditions, and thus preemptively charging the battery when necessary. However, to avoid employing another path planning strategy that conflicts with the PSO, HEPS-on commands should be a fifth optimisation variable to the algorithm. But since only a few cases encounter the aforementioned problematic situation, this has been deemed to do more harm than good as it would clutter the optimisation process, and is therefore not included as a strategy.

5.4.2 Battery configuration

The battery parameters are based on a 10-cell LiPo battery with assumptions as stated in Section 4.4. The battery is considered fully charged during mission start, and has configuration:

- C_{cut} = 26.400 Ah (discharged capacity at cut-off)
- C_{exp} = 2.640 Ah (discharged capacity at end of exponential range)
- C_{nom} = 20.400 Ah (discharged capacity at end of nominal range)
- V_{full} = 41.8 V (fully charged potential)
- V_{exp} = 39.67 V (potential at end of exponential range)
- V_{nom} = 37.67 V (potential at end of nominal range)
- R_c = 0.015 Ω (internal resistance)
- I_{rated} = 660 A (rated maximum discharge current)
- n = 1.05 (Peukert constant)

5.5 Discretisation

Three parameters indicate the discretisation of the desired paths, and have implications on the potential resolution of a solution and the details in calculation. A constant determines the number of waypoints, while the total number of discretised steps N is determined by the number of airspeed and climbing angle changes K and some multiplication factor F, giving:

$$N = KF - 1 \tag{5.5.1}$$

These are a determining factor in the running time of the algorithm, and are ideally high enough to properly use the resolution of the weather data. The total number of waypoints is not a factor in the total number of discretisation steps, but each waypoint will be placed in a single step along the path. All simulations were run with with 10 changeable waypoints and 150 discretisation steps, by $K = 5$ and $F = 30$ in Equation

5.5.1. This means that every 5 consecutive discretised steps will have the same airspeed and climbing angle, and that 30 such segments of 5 identical values will be present.

6 Missions

To demonstrate the practical viability of the system it is desirable to employ use case parameters that could mirror a realistic application of the platform. Therefore a theoretical mission for this system with a concrete societal value is presented, from which explicit configurations are extracted. Ignoring the obvious challenges surrounding the legality, funding and interdisciplinary expertise required to implement and routinely complete such missions can seem somewhat reductive. However, this approach is to give a sense of the full potential of HEUAVs as delivery platforms.

6.1 Long distance hybrid electric flight

Drone deliveries between hospitals has recently been green-lit and funded as a joined research effort in Norway [35]. The project is entirely focused on Oslo, delivering blood samples and supplies between the capital's four hospitals. This would improve the efficiency of the cooperation between the hospitals and would allow samples, tests and results to be quickly delivered, regardless of the current traffic conditions. However, these hospitals are within relative vicinity of each other, and travel between them are typically less than 20 minutes by car. As important and exciting the project is, it does not explore the full potential of drone delivery platforms.

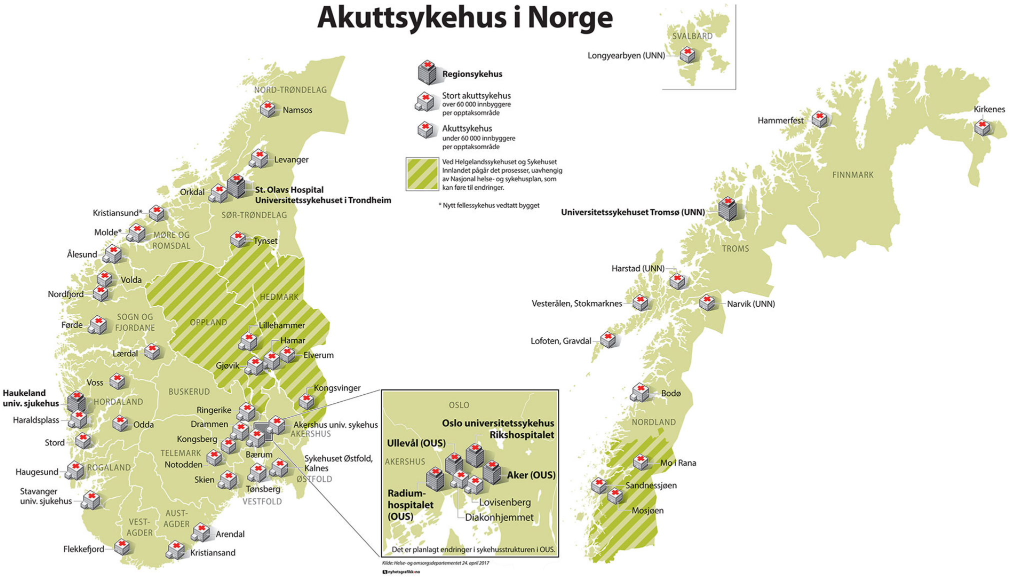

Figure 6.1: Hospitals and emergency departments in Norway, from: [36]

Figure 6.1 illustrates a map of regional hospitals, and larger and smaller emergency departments in Norway. The four Oslo based departments mentioned above are showed in the expanded square in the centre middle. Although there's a relatively high frequency of medical departments in some of the coastal population centres, certain institutions, particularly further north, are required to serve populations of vast geographical areas.

Inhabitants in many of the sparsely populated municipalities of Norway raise the issue of degrading medicinal services for the local populations, as the bigger hospitals in the cities are far better funded due to larger populations. A way of improving the quality and efficiency of health clinics in remote areas in the country could be through drone deliveries. This would for instance allow blood samples to be taken at a local clinic, and subsequently delivered to the closest hospital with equipment better suited to perform analyses on said sample. Routes in the northern parts of the country, between different towns' medical departments, will be the backdrop for simulations in this dissertation.

This area is characterised by harsh winters with heavy wind and plenty of precipitation, making it an ideal candidate for this work. Similar mission profiles are applicable to most of the country's emergency departments and hospitals, and switching between them is merely a matter of changing waypoint coordinates and downloading the proper data sets. A map showing the relevant locations is shown in Figure 6.2, and the respective geographical coordinates are presented in Table 1.

Firstly, a route between the two cities of Bodø and Tromsø will be simulated. This route has fairly complex topography considering the ground below, as it follows the fractal coastline. Using generic direction planning software, driving this distance by car would take about 8 hours 23 minutes, including one ferry, over a distance of 536 km.

Secondly, a route between Tromsø and the small town of Kirkenes near the Russian border will be considered. The related topography of this area is somewhat less mountainous than the previous route, but it is heavily affected by wind partially due to being situated on the polar front. This trip would take well over 10 hours by car due to the vast distances and poor road infrastructure in the north of the country, resulting in a distance of almost 800 km. Most optimal UAV routes this mission would likely pass through Finnish borders, which would in practical terms require some cooperation with the Finnish aviation authorities. An alternative is modifying the DTM to list all coordinates within Finnish borders as infinite elevation, which would invalidate all paths going through Finland and force the algorithm to search only in Norwegian airspace.

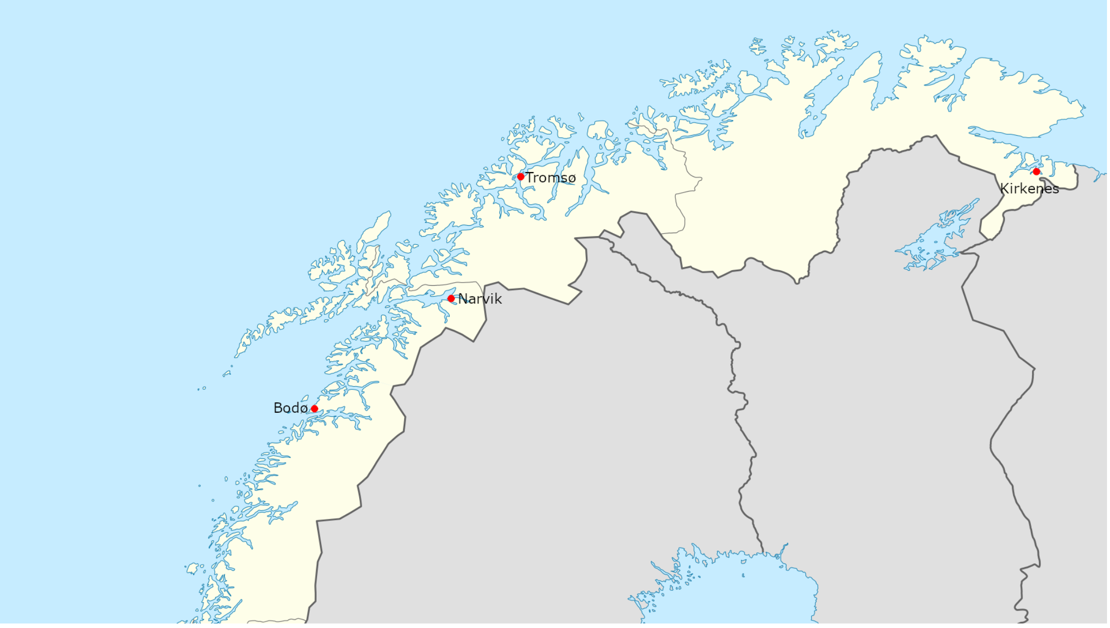

Figure 6.2: Map of northern Norway with mission waypoints marked

Currently, the algorithm can explore upwards of 2300 m above sea level, but it is worth noting that the Norwegian Civil Aviation Authorities typically disallow activities above 120 m, which would imply that these types of missions need special permissions to be

Location	Latitude	Longitude
Bodø	67.280111°	14.398633°
Tromsø	69.682484°	18.982639°
Kirkenes	69.708466°	30.034848°
Narvik	68.440818°	17.415258°

Table 1: Mission waypoint coordinates

run.

6.2 No-IPS flight

As electro-thermal icing protection systems are relatively new to the UAV market, many operations the next few years will be performed without an IPS. Therefore, it is interesting to employ the algorithm to a platform expecting ice accumulation, and seeing how it performs. The platform will experience degradation of its aerodynamic coefficients following the relations explained in Section 3.5.2.

The no-IPS configurations will mostly be employed on shorter mission profiles. This is because it would be unrealistic to launch a mission spanning multiple hours and hundreds of kilometres when one expects icing. Prolonged icing eventually causes large increases in drag force, requiring ever more propulsive power to compensate for. Even without considering the stability issues this may cause, the power consumption by the propulsion system may exceed that which the HEPS can provide. Thus the system may exhaust the batteries (implying a crash) despite having plenty of fuel left.

The no-IPS configuration will be employed between the towns of Bodø and Narvik, as well as two limited missions on the same data sets as used in the first mission type. The Bodø - Narvik route is roughly half the distance of Bodø - Tromsø, and is a practical alternative as one can use the same weather data, and the same DTM as the latter.

6.3 No icing conditions

To investigate the applicability of the system in nominal weather conditions as well is interesting, as a lot of the institutions utilising UAV services globally seldom have to worry about airframe icing. Therefore, a limited amount of simulations will be made in which the default solution is not affected by icing conditions. Expected from such a mission is energy optimisations of slightly smaller magnitude, as increases in efficiency can't be made by avoiding icing conditions. Time optimisations should theoretically be of the same magnitude as the other missions.

7 Results and discussion

The following section will present the simulation results of employing the algorithm on multiple mission configurations. Discussion regarding the results, as well as plots illustrating the algorithm's behaviour, is also contained within this section. Three subsections will present results from three different mission configurations, the first being the main long-distance mission with an IPS configuration in wind and icing conditions. The second will compare results from an IPS configuration to that of a no-IPS configuration actively experiencing aerodynamic degradation. The third will present limited results from the algorithm employed in no icing conditions.

By browsing historical weather data of Norway, one can easily pick out dates and subsequently data sets with the weather conditions most relevant to this application. Associated meteorological and elevation data was downloaded as explained above, simulations were run and selected data was collected. In this section, a *profile* will refer to a single route on a specific data set. A boldface number (**1**, **2**, etc.) will in this section also refer to such a profile, unless otherwise indicated. Table 2 shows indices of these different result profiles. Table 3 shows tags each profile can be given, indicating the specifics of a single simulation on a profile.

Profile	**Route**	**Date**
1	Tromsø - Bodø	14.02.2020
2	Bodø - Tromsø	14.02.2020
3	Kirkenes - Tromsø	10.02.2020
4	Tromsø - Kirkenes	10.02.2020
5	Tromsø - Bodø	06.03.2020
6	Bodø - Tromsø	06.03.2020
7	Narvik - Bodø	05.02.2020
8	Bodø - Narvik	05.02.2020
9⋆	Tromsø - Bodø	11.03.2020
10⋆	Bodø - Tromsø	11.03.2020

Table 2: Indices of Profiles for result tables
⋆ Default route is without icing conditions

Note that the dates and data sets are not picked entirely at random, as specific conditions are required to produce interesting results. For instance, data representing entirely clear skies are not relevant to a mission intended to perform path planning for an IPS configuration, although the algorithm would run just fine. One can expect lower total optimisations if the conditions are not heavily influenced by icing weather and wind.

In all default solutions, the aircraft climbs to a constant appropriate altitude based on the highest point of the expected topography, and then cruises at a constant airspeed of 28 m/s. This is slightly higher than the proposed cruise velocity by Maritime Robotics to remove some of the inflation in optimisation from just increasing the desired airspeed.

7.1 IPS configuration

Results from the long-distance flights all equipped with an IPS are presented in Table 4. The data set from which Profile **1** and **2** is created is clearly influenced by heavy wind

Descriptor	Name	Explanation
D	Default solution	Straight cruising flight at constant airspeed
E	Energy optimised	Route minimising energy expenditure through cost function 5.2.1
T	Time optimised	Route minimising time expenditure through cost function 5.2.2
N	No-IPS	IPS pulling no power and aerodynamic coefficients degrade as in Section 3.5.2

Table 3: Descriptors for individual results

favouring a route flying from Bodø to Tromsø, but with relatively light icing conditions. Profile **3** and **4** represent a longer route in heavy icing conditions. Profile **5** and **6** represent geographically identical routes to **1** and **2** but with lighter wind and more evenly spread icing conditions.

Profile	**Expended energy**	**Time spent**	**Fuel left**	**Battery cap left**	**Tot. time in ice**	**Avg. height**	**Tot. path length**
1 - D	116.739Ah	6h14m29s	0.518L	14.873Ah	1h18m51s	1320m	328.0km
1 - E	63.713Ah	4h30m19s	2.365L	12.075Ah	0h12m20s	1059m	349.9km
1 - T	68.805Ah	3h36m22s	1.923L	20.337Ah	0h42m34s	1199m	342.7km
2 - D	47.503Ah	2h15m56s	2.779L	15.771Ah	0h24m55s	1320m	328.0km
2 - E	32.924Ah	2h2m27s	3.155L	19.004Ah	0h0m42s	1237m	333.5km
2 - T	35.986Ah	1h52m3s	3.170L	15.490Ah	0h19m42s	1290m	334.1km
3 - D	124.758Ah	5h57m15s	0.507L	7.170Ah	3h22m45s	881m	428.5km
3 - E	70.590Ah	4h19m55s	1.982L	16.775Ah	0h21m4s	1517m	448.7km
3 - T	79.146Ah	3h32m9s	1.643L	18.462Ah	1h30m2s	1353m	438.1km
4 - D	92.614Ah	4h17m24s	1.236L	17.283Ah	2h14m15s	881m	428.5km
4 - E	54.314Ah	3h17m5s	2.594L	14.576Ah	0h11m14s	1598m	436.0km
4 - T	67.412Ah	2h56m9s	2.388L	7.693Ah	1h15m12s	1405m	436.9km
5 - D	85.600Ah	3h33m57s	1.353L	20.757Ah	2h27m33s	1320m	328.0km
5 - E	57.799Ah	3h0m34s	2.338L	18.807Ah	0h38m51s	1782m	343.9km
5 - T	61.105Ah	2h26m33s	2.476L	11.343Ah	1h9m45s	1965m	334.5km
6 - D	78.024Ah	3h11m19s	1.607L	20.668Ah	2h14m13s	1320m	328.0km
6 - E	53.584Ah	2h44m8s	2.808L	8.814Ah	1h12m43s	658m	340.7km
6 - T	54.776Ah	2h28m30s	2.766L	8.918Ah	1h33m6s	686m	331.9km

Table 4: Results for long-distance flights with IPS with profiles as described in Table 2

Significant energy savings could be made in each of the simulations above. Generally, energy optimisations in these results range between approximately 30% and 45%, while time optimisations range from 17% to 40%. Energy optimisations for Profile **1 - E** and **2 - E** were quantified to 45.42% and 30.69%, respectively. Time optimisations for **1 - T** and **2 - T** correspond to 42.22% and 17.57%. **1** is clearly experiencing headwind, while **2** is experiencing tailwind, as evidenced by an almost threefold difference in default route flight time despite flying exactly the same distance. Intuitively, flying in headwind can leave more potential for optimisation, which is why the headwind route gains proportionally

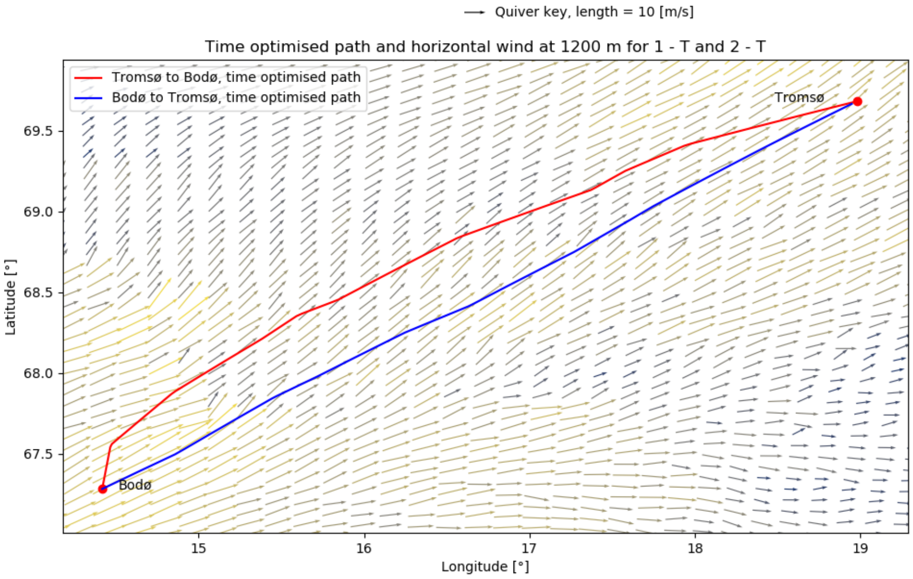

Figure 7.1: Time optimised path against horizontal wind at 1200 m, **1 - T** and **2 - T**

more from the algorithm. The optimal paths for **1** are also significantly longer than the corresponding paths for **2**, indicating that a more active planning approach could be needed to optimise a path in these conditions. Figure 7.1 displays the desired positions for **1 - T** and **2 - T** plotted over a slice of expected horizontal wind at an altitude of 1200 m. The plot clearly shows the slightly longer path required for the configuration in headwind, compared to that in tailwind. Note that both paths also have significant variations in desired altitudes along the path, so actual encountered wind at each discrete step might vary somewhat from the vectors representing wind here.

Profile **3** and **4** saw energy optimisations of similar magnitudes at 43.41% and 41.35% respectively. In all mentioned profiles, there is a clear correlation between the amount of icing conditions that can be avoided and the energy saved. Further, reducing the active flight time reduces total energy consumption by both the IPS and the propulsion system, making it a perfectly viable strategy to save fuel and battery capacity. Hence, the paths for the time optimised and energy optimised route are often correlated. Time optimisations for **3 - T** and **4 - T** are quantified as 40.61% and 31.57% respectively, whose routes also correspond to energy savings of 36.56% and 27.21% compared to the default route, to illustrate the previous point. Figure 7.2 displays remaining battery capacity and fuel for profile **3 - E** compared to **3 - D** over all time steps. Notice the default solution's inability to recharge the battery over time steps 20 to 80, causing a constant drain of fuel over this period. This likely stems from the IPS consuming most of the excess power from the HEPS during this period.

Figure 7.3 displays the altitudes of profile **4 - E** against the presence of icing conditions at different heights. Icing conditions are displayed for a column of all possible altitudes under and over the desired position at a given step. Blue dots indicate an icing condition, that is a cell which has a temperature under 0°C, relative humidity over 0.99 and liquid water content over 0.01 g/m^3. Notice the energy optimised solution's decision to fly over a ridge of icing conditions, exploiting a gap underneath the maximum allowed altitude, and then descending through an unaviodable stretch of icing conditions towards the end of the path. The altitudes of the default path are plotted as a dashed line. Note that since

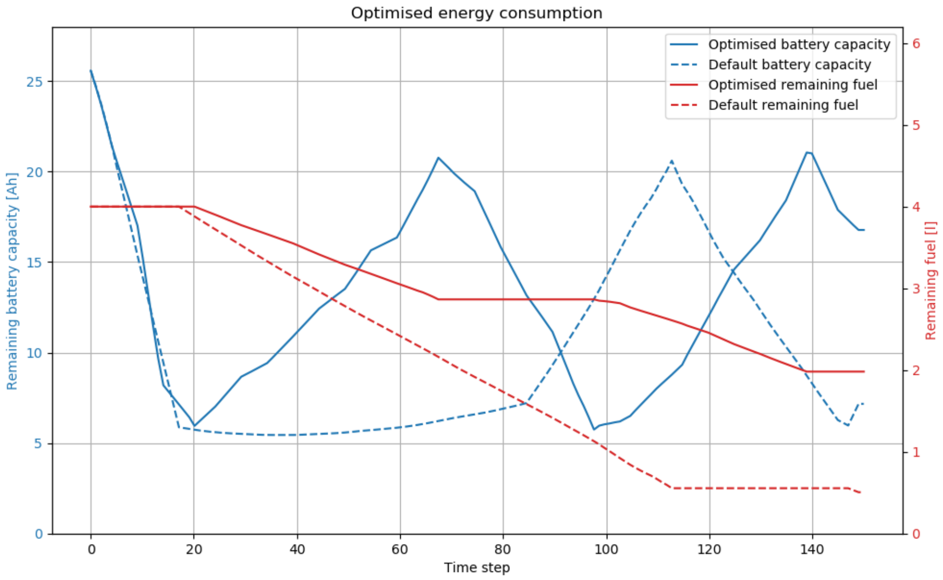

Figure 7.2: Remaining fuel and battery capacity for **3 - E** against default **3 - D**

the conditional columns, including icing and elevation, are determined for the optimised path, the conditions may not necessarily apply 1-to-1 for the default path, but they are likely to be quite similar.

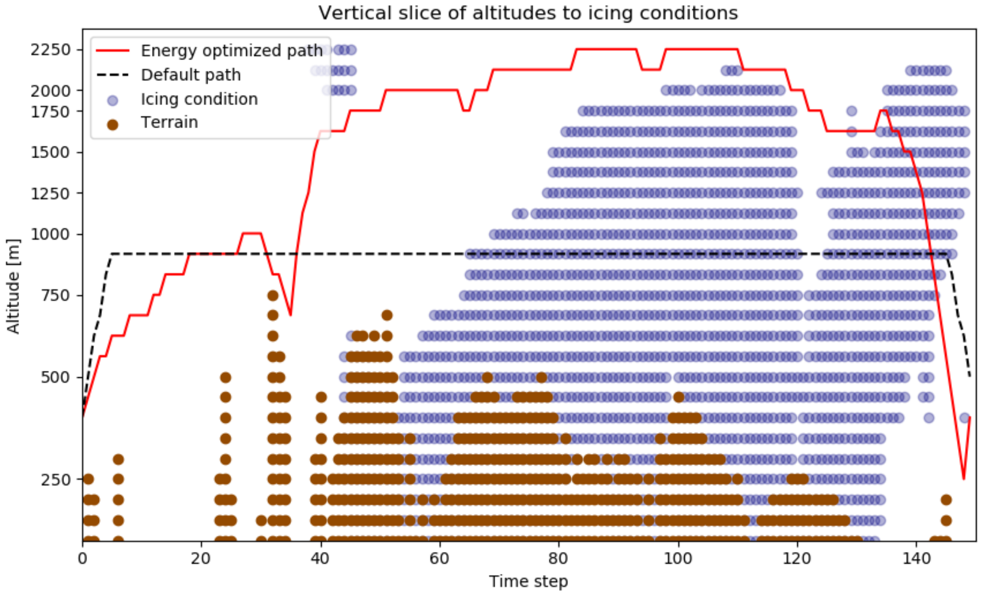

Figure 7.3: Flight altitudes to vertical columns of icing conditions for **4 - E**

Profile **5** and **6** saw energy optimisations of 32.48% and 31.32% respectively, and time optimisations of 31.50% and 22.38%. Of note in these profiles is that the time optimisations **5 - T** and **6 - T** only uses about 3 Ah and 1 Ah more than **5 - E** and **6 - E** despite saving over 30 minutes in the first case and 16 minutes in the second case. This indicates that it could be beneficial to perform both optimisations and choose the most fitting in a practical setting. Another option is constructing a cost function that combines time and energy expenditure. Figure 7.4 displays projections of the wind vectors span along the course of profile **5 - T**. In mathematical terms, the horizontal component of a wind vector in time step i is determined by:

$$w_{horizontal} = \frac{w_e \cdot \Delta\lambda_i + w_n \cdot \Delta\phi_i}{\sqrt{\Delta\lambda_i^2 + \Delta\phi_i^2}} \tag{7.1.1}$$

where $\Delta\lambda_i$ and $\Delta\phi_i$ are the changes in longitude and latitude respectively at step i, w_e is the eastbound wind and w_n is the northbound wind. The vertical components of the vectors in the figure equal the upwards wind at the point.

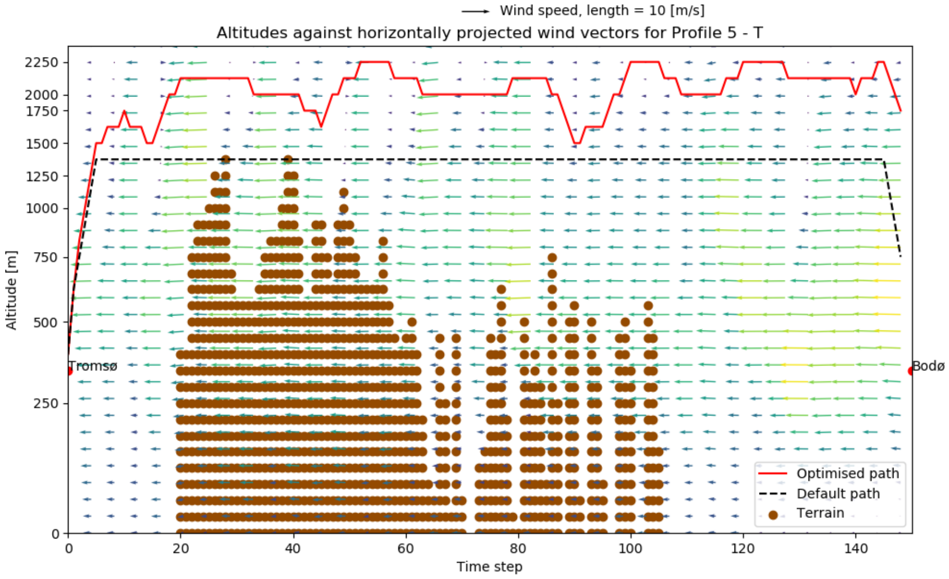

Figure 7.4: Horizontal projections of wind vector against altitudes for **5 - T**

Typically, individual desired airspeed values for energy optimised routes range from 30 m/s to the maximum airspeed of 40 m/s. Time optimised solutions usually prefer values close to 40 m/s in every step, with some exceptions. For instance, when the sum of propulsive power and IPS required power exceed that which the HEPS can generate over multiple consecutive steps the battery might entirely discharge, causing an infeasible solution. In situations like that the algorithm will prioritise either high airspeed around the relevant area of icing conditions, or reduce airspeed to cruise through the area with sufficiently low power requirements to stay flying. Interestingly, in all solutions above, performing a time optimisation has reduced the amount of time spent in icing conditions. The running theory is that, like explained above, avoiding icing conditions is a necessary step in order to expend the power required to fly close to max airspeed without draining the battery with additional de-icing.

In general, the de-icing algorithm requires around 60% less power than the anti-icing algorithm. As explained in Section 3.4 using the de-icing strategy leads to a penalty in terms of an increased drag coefficient, but it turns out that very seldom will the sum of the power required for de-icing and the increased power usage from airframe icing exceed the power required to run anti-icing. The immediate consequence of this is that the algorithm will, at the vast majority of steps with icing conditions, prefer to use use de-icing over anti-icing. Whether this is a fair assumption is unknown at this point, but it would be highly beneficial to include further penalties on for instance stability when allowing airframe icing, but that would require more accurate models, such as from [8].

7.2 No-IPS configuration

Table 5 displays results for the second mission type, with the intent of comparing results for configurations with and without an IPS. The algorithm was also run with an IPS-free configuration on Profile **5** and **6** from the previous table, presented here as variations on **5 - N** and **6 - N**. Profile **7** and **8** represent a shorter route than above, roughly half the total length, but with quite heavy wind.

Profile	Expended energy	Time spent	Fuel left	Battery cap left	Tot. time in ice	Avg. height	Tot. path length
5 - DN	100.846Ah	3h33m57m	1.240L	8.952Ah	2h27m33s	1320m	328.0km
5 - EN	50.650Ah	2h56m54s	2.645L	16.697Ah	0h40m57s	1810m	339.0km
6 - DN	83.232Ah	3h11m19s	1.860L	7.819Ah	2h14m13s	1320m	328.0km
6 - EN	50.658Ah	2h56m6s	2.995L	6.097Ah	0h45m0s	1411m	345.3km
7 - D	81.316Ah	4h2m13s	1.897L	8.619Ah	1h40m8s	1108m	181.4km
7 - DN	104.305Ah	4h2m13s	0.876L	16.487Ah	1h40m8s	1108m	181.4km
7 - E	34.469Ah	2h8m52s	3.197L	16.191Ah	0h30m26s	363m	186.0km
7 - EN	28.773Ah	2h3m41s	3.273L	19.579Ah	0h8m17s	1001m	185.9km
7 - T	36.202Ah	1h51m37s	3.017L	19.894Ah	0h26m50s	377m	188.9km
8 - D	29.771Ah	1h22m33s	3.443L	13.471Ah	0h32m14s	1108m	181.4km
8 - DN	26.531Ah	1h22m33s	3.605L	11.818Ah	0h32m14s	1108m	181.4km
8 - E	16.442Ah	1h18m31s	4.000L	9.958Ah	0h6m50s	548m	197.0km
8 - EN	17.665Ah	1h14m54s	4.000L	8.735Ah	0h19m43s	610m	185km
8 - T	23.374Ah	1h4m59s	3.885L	6.501Ah	0h26m30s	598m	183.7km

Table 5: Results for medium-distance flight with and without an IPS

The no-IPS default solution **5 - DN** saw an energy expenditure of over 100 Ah, approximately 15 Ah more than the corresponding IPS default solution **5 - D**. There was an estimated 421.3 g of ice accumulated in this solution. However, after performing energy optimisation, estimated energy expenditure was reduced by almost half, resulting in a slightly more energy efficient route than the IPS solution **5 - E**. **5 - EN** accumulated an estimated 75.5g of ice. **6** saw similar results, where **6 - DN** had slightly higher energy consumption than **6 - D** and was subsequently optimised to a route **6 - EN** comparable to **6 - E**. Ice accumulation for this optimisation went from 380g to 61.8g. It is important to remember that the only degradation in aerodynamic coefficients is considered in the no-IPS simulations, while issues relating to for instance stability is disregarded. Therefore it is somewhat misleading to claim that the energy optimised no-IPS flights would be more efficient than the optimised IPS flights. Altitudes of **5 - EN** compared to the corresponding icing conditions are displayed in Figure 7.5.

The most dramatic reductions in energy expenditure occurred in profile **7**. Heavy headwind forced the solution to stay in icing conditions for a prolonged time, resulting in large increases in drag force for **7 - DN**. Figure 7.6 compares the power consumption of **7 - DN** and **7 - D**, where **7 - D** is decomposed into required propulsive power and IPS power. Notice that propulsive power requirements are initially mirrored by the two solutions, but **7 - DN** gradually develops a significant gap to that of **7 - D**, illustrating the additional drag force the no-IPS configuration has to overcome. The spikes in power requirements likely stem from conditions that require a high (or low) angle of attack. As discussed in

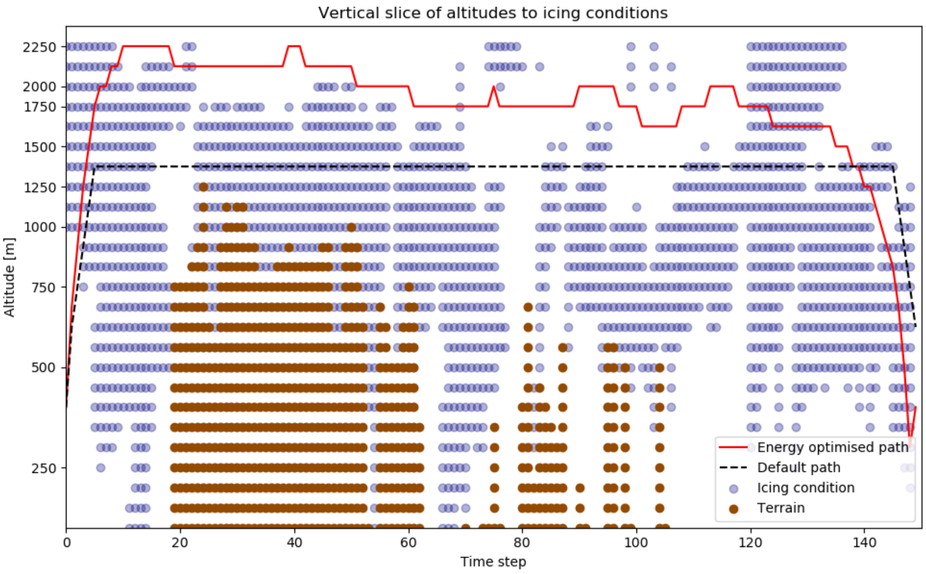

Figure 7.5: Icing conditions compared to altitudes of **5 - EN**

Section 3.5.2 the resulting drag coefficient in larger angle of attacks after icing increase way more than those around zero. This is also witnessed in the final step of this solution, where a steep descent is expected, forcing a low angle of attack and subsequently high drag force to overcome.

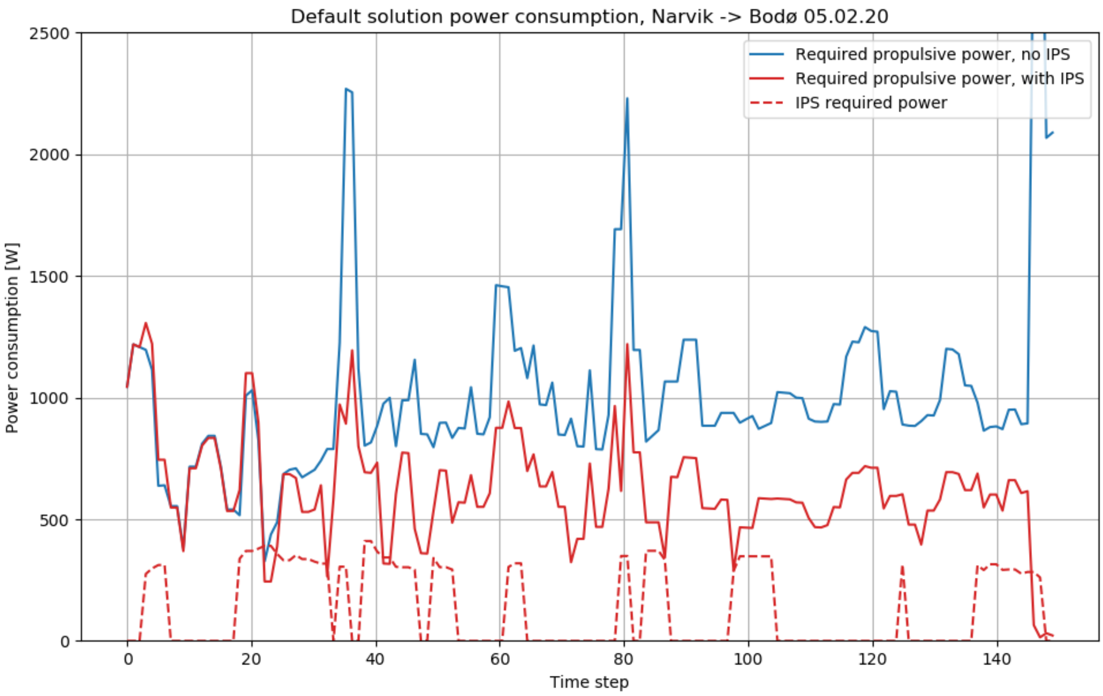

Figure 7.6: Default solution power consumption with and without IPS for **7**

After optimisation energy consumption for **7 - EN** is reduced by a hefty 72.41%, and reducing the weight of the accumulated ice from 423g to 25g. The dramatic reduction in energy expenditure is achieved by reducing the flight time by almost 50% and minimising the time in icing conditions to an almost negligible amount without increasing the total travelled distance by more than 5km. Judging by the properties of **7 - EN** compared to **7**

- E, it seems as though **7 - EN**'s route should be superior, also with an IPS configuration, indicating that a globally optimal route might not be guaranteed with the current setup. The differences in results are minuscule, and likely negligible in a practical situation with dynamic weather, but it is something to be aware of. Time optimisation was only performed on an IPS configuration, as the results for the two configurations are expected to be identical. Still, the time optimised route **7 - T** achieved the largest increase in performance, reducing the active flight time by 53.92%.

The opposite route, profile **8** in tailwind, saw somewhat smaller optimisations. **8 - EN** reduced energy expenditure by 33.42% while **8 - E** reduced by 44.77%. **8 - E** and **8 - EN** were almost identical in expenditure, but of note is that **8 - E** is significantly longer than its opposite counterpart **7 - E**. Time expenditure for this route was optimised by 21.28%, and the lack of significant increases in path length indicate that this was primarily done by increasing desired airspeed.

7.3 No icing conditions

Table 6 displays results for Profile **9** and **10**. These are unique in that neither profile's default solution is considered to be in icing conditions at any time during flight. This scenario was deemed worthy of inclusion to illustrate the potential for optimisation when there are no icing conditions to avoid.

Profile	Expended energy	Time spent	Fuel left	Battery cap left	Tot. time in ice	Avg. height	Tot. path length
9 - D	81.030Ah	4h46m47s	1.658L	16.118Ah	0h0m0s	1320m	328.0km
9 - E	49.661Ah	4h1m42s	2.525L	21.298Ah	0h0m0s	938m	343.0km
9 - T	62.307Ah	3h10m2s	2.237L	17.351Ah	0h41m20s	1939m	332.3km
10 - D	46.399Ah	2h36m8s	3.104L	7.083Ah	0h0m0s	1320m	328.0km
10 - E	31.696Ah	2h28m0s	3.256L	17.174Ah	0h0m0s	1152m	337.5km
10 - T	42.511Ah	2h9m43s	3.062L	12.225Ah	0h12m55s	1796m	332.5km

Table 6: Results for route without default icing conditions

Profile **9** and **10** still showed a high potential optimisation, despite not being able to reduce energy expenditure by avoiding icing conditions. **9 - E** reduced energy expenditure by 38.71%, and **10 - E** by 31.68%. This promises well for the usefulness of weather optimisation in all conditions. Time optimisations were 33.74% for **9 - T** and 16.92% for **10 - T**. For reference, **9 - E** experienced an average horizontal wind projected along its course of -1.99 m/s, while **9 - D** experienced -5.22 m/s. This is following calculation of horizontal projected wind as in Equation 7.1.1 for each step and then performing a cumulative moving average similar to Equation 3.5.1. This average improvement of 3.23 m/s experienced over 4h46m47s corresponds to 55.6 km of "free" travelled distance, despite **9 - E** only travelling a path 15 km longer than **9 - D**. This illustrates the potential of optimising paths with respect to surrounding wind.

8 Future work

As the results of the simulations have been presented, it is valuable to suggest which aspects of the system can be expanded upon, and certain modifications to improve the realism and applicability of what's been created. The following section will present some of the author's thoughts regarding the specifics of this potential future work, in the hopes that it can provide inspiration and some guidance to researchers and engineers who wish to expand upon the work.

8.1 Model improvements

As mentioned in the previous section, the IPS configuration almost exclusively prefers permitting icing on the wings and subsequently performing de-icing at the cost of slightly increased drag. Finding some way of quantifying the losses in terms of stability as a result of this icing could allow some more nuanced behaviour from the IPS. Additional cost penalties from degraded stability could for instance be determined by the increased chance of stalling (and thereby crashing) in the conditions at a given step. Generalizing this calculation, and including it into the cost functions, could permit for instance penalizing "risky" behaviour by the path planner. Such behaviour could be flying at sharp angles of attack, made further costly by doing this when ice is present on the airframe.

On the subject of penalizing specifics in the planned paths, calculating details regarding the act of turning could provide great benefit to the system. Especially if higher resolution weather maps will be used in the future, a large number of bank-to-turns will be performed in a route. As of now, the UAV is thought to freely change its heading based on the desired waypoints, and all turns happen instantly. To prevent frequent zigzagging and encourage realistic behaviour, estimates on the energy required to change heading by a desired amount, and potentially costs representing the risk associated with such a turn, could be deployed.

In the current system, icing on the airframe leads to an increase in drag coefficient, and by extension drag force. Despite this, the aircraft is still "permitted" to fly at the maximum airspeed, while it in reality probably should decrease the maximum possible airspeed based on the amount of accumulated ice. Improving the calculations of the balance of forces could make such a relation mathematically provable, and would provide a realistic further penalty to icing by forcing an iced aircraft to stay longer in the air. One issue with the determination of forces on the vehicle is the discretisation. As the algorithm currently entirely disregards acceleration, and considers every step to have a static state, a lot of the potential nuances of force equations are lost.

Currently, the propeller efficiency η_{prop} is assumed to be constant, no matter the weather conditions or state of the vehicle, and employing a more realistic model could benefit the system. Modelling η_{prop} as a function of, amongst others, airspeed, as in [37], would lead to interesting optimisational trade-offs, where high speeds are punished more than they currently are. Employing such a model would provide varying values for η_{prop} from 0 upwards to around 0.8. Since the propulsive force is linearly scaled by η_{prop}, this could have a large and interesting effect on the behaviour of the algorithm.

The HEPS presented in this work is a quite simple model. This is justified by how long the discretisation steps are, and that intricate performance implications and details are lost in these large steps. The structure of the HEPS allows the system to run in its optimal configuration, but it could be interesting to look at more realistic ICE performance maps for added realism, as in [13]. A solid improvement on the system would be performance penalties for starting and stopping the ICE, especially if more advanced control strategies are to be employed. Currently, a simple rule-based method is used for controlling the ICE in the HEPS. The ICE will run from the moment the battery capacity hits the end of the nominal range, and continue to run until the capacity has reached the start of the nominal range, as the battery requires less power to charge in this area. However, in certain situations, this method can lead to infeasible solutions that could have been avoided by using smarter control methods, to ensure the battery has enough capacity when entering power intensive sections. This can be done either by a different rule-based approach, such as setting a cap for how little capacity the UAV can have when entering icing conditions and preemptively controlling the ICE thereafter, or by a more optimisation oriented approach. Preemptively charging the battery based on expected icing conditions make a lot of sense in a system driven by forecasts, so not taking advantage of the available a priori knowledge is a wasted opportunity. Implementing a fuel cell-based HEPS could also be interesting, as HEUAVs are in their relative infancy, and a proper comparison on the performance of the two could be a valuable contribution to the field.

In a physical setting, available battery capacity varies greatly based on temperature, and LiPo battery capacity can decrease by up to 35% operating in -20°C compared to room temperature [38]. As of now, the battery model is not temperature dependent, and the available battery capacity at an instant is only dependant on the expended energy up to that moment. Implementing a temperature dependent battery capacity would require simulating heat flux from the battery and ICE into the UAV shell, as well as estimating heat dissipation from the UAV based on the surrounding air temperature. This is a complex task which would require knowledge of material science, chemistry and specifics regarding the inner and outer geometry and layout of the specific aircraft.

8.2 Weather improvements

In all simulations performed above, the optimisations are performed on a static, unchanging weather map. This is a justifiable assumption for very short missions, but in this case, the missions typically take between 3 and 6 hours. Weather can change drastically within such a time period, and the estimated power and time consumption is not likely to be accurate in a practical application. One method of creating dynamic weather maps would be through the linear interpolation of consecutive data sets. Say data sets can be acquired at a frequency of 3 hours. Acquiring and storing both the data set from take-off, as well as the data set from 3 hours later, the evaluated value for some weather parameter W_n at some step n can be expressed by the associated values from the chronological data sets W_1 and W_2 as:

$$W_n = \frac{(3 \cdot 3600 - T_n)W_{1,n} + T_n \cdot W_{2,n}}{3 \cdot 3600} \tag{8.2.1}$$

where T_n is the cumulative time spent up to step n in seconds. Such a linear interpolation between data sets was implemented in this work, but later decided to not be used in simulations. The reasoning for this relates to the nature of the liquid water content parameter, which is essential to determining the icing effect. For continuously changing parameters such as wind or temperature, interpolation provides decently realistic moving weather images. However, LWC varies from typically 0 g/m^3 in a clear area to at most 0.4 g/m^3 in dense clouds, but typically quite a bit lower. Performing interpolation between two sets representing such values causes the resulting time varying data set to represent a wide-spanning fog, with LWC values never quite high enough, and never quite low enough. So until a method is created which can estimate the continuous movement of a cloud in its entirety, linear interpolation of data sets is put on ice, so to speak.

Simulations in this work are performed on historical weather data, which is useful for determining the potential for optimisation, but not for any practical purposes. To employ the system in a physical setting one would need to use weather forecasts of sufficient accuracy. Whether this data is readily available sufficiently far in advance to be usable is uncertain at this point. Potentially, to make a scalable service utilising this system, one would need cooperation with certain meteorological data centres to produce the required subset of data at regular intervals.

Another limitation of the algorithm currently is the resolution of the data sets. As the cells of weather data are 2.5 km by 2.5 km, performing detailed optimal path planning is difficult. One could feasibly perform linear interpolations within a specific data set to increase the resolution of the map, but this might to some degree encounter the issue regarding interpolation described above.

Quantifying the contributions of considering vertical wind in the system has proven difficult. The author could not find significant differences to energy expenditure when performing optimisation with and without the data set containing upwards wind. To properly utilise the expected upwards wind, the UAV must be allowed to soar for extended periods to harvest the energy of vertical wind, as in [39]. This would increase time expenditure, but could be a dynamic asset in long-range missions with limited energy, that are not constrained by desired flight times.

8.3 Missions and simulation environment

The current mission profiles are relatively simple in that they consider a starting and stopping waypoint, and simply finds a path to traverse the world between these two points. Complications to the mission profiles could be made, such as adding lawnmower patterns as fixed waypoints, which is a common surveying method. Another could be to find the optimal way of covering the maximal amount of a given area, which would be useful in surveillance applications. Further, adding no-fly zones and altitudes, or discouraging ICE-active flight over populated areas could be a step towards a physically applicable system.

The system as is would be well suited to being used as a high-level guidance system. That is, a system which produces desired waypoints and velocities, which the onboard computer interprets into actuation signals. The actual performance of the vehicle through an operation is likely to be higher than what is estimated here, as variable wind and suboptimal flight control leads to higher energy expenditure. A way of increasing the fidelity of the simulations could be to use the guidance system and static weather maps as is, and on top of that add random noise to emulate a continuous, varying weather map. On this map, one could employ for instance model predictive control that in continuous time attempts to follow the waypoints and velocities produced by the system presented in this thesis. Gust and turbulence could be simulated by for instance running white noise through the Dryden model, as found in [21].

8.4 Efficiency and running time

Running time is a matter of interest if one desires to run the algorithm on an onboard computer and adapt routes in real time. All simulations were run on a 3.40 GHZ Intel Core i7-6700 CPU. During simulations, roughly 50% of this CPU's resources were utilised simultaneously alongside approximately 70 MB of RAM. One advantage to the efficiency of this algorithm is that each particle's calculations can be parallelised as long as a central module performs comparisons between each iteration. One iteration, updating 256 particles and candidate solutions, took generally a bit under 0.3 seconds. For a total of 256 iterations, this equals a total running time of 75 seconds.

However, only rudimentary code optimisation was performed, to the degree where efficiency was no longer a bottleneck. The author is quite certain one can achieve significantly better performance than the system experiences in its current state, especially if for instance GPUs are introduced. Also of note is that the system does not necessarily need to run onboard on every UAV that requires it. Instead, a central hub or ground station can run the optimisations, and then broadcast the ideal waypoints and airspeeds to the individual UAVs, leaving the UAV computers only with the task of following these waypoints.

9 Conclusion

In conclusion, a discrete-time, high-level path planner for a hybrid electric UAV with an icing protection system was implemented as presented in this work. Included was a deterministic model for the aircraft's state at a given moment based on four input variables and parameters representing the surrounding weather conditions. The hybrid electric power system was created by combining a simple combustion engine model with a 10-cell rechargable LiPo battery in the form of a Tremblay model. The components were combined in a modified series parallel architecture considered to run in a constant, optimal configuration following a rule-based control algorithm.

A UAV model representing a mid-sized 20 kg vehicle, based on a real-world counterpart, operated discretely through spatial weather maps considering power consumption and flight times. This was made possible by the availability of a wide range of historical weather data of the relevant areas which could be downloaded and simulated upon. The main configuration of the UAV was equipped with a model of an icing protection system that could perform de-icing or anti-icing operations at the cost of power. Another configuration not equipped with an icing protection system was also deployed to illustrate the effects of unmitigated icing of a UAV. Achieving this was done by adapting a realistic model of the expected degradation of aerodynamic coefficients based on specific weather parameters.

Through the particle swarm optimisation method, the ideal set of waypoints, desired airspeeds and climbing angles was generated to represent the best way of traversing said maps. Paths were optimised based on a cost function determining total energy consumption, or total flight time.

Results were generated from data sets of winter weather from northern Norway, flying between a total of four different towns in the area, representing routes with lengths varying from around 180 km to 430 km. Most of the data sets were representing icing conditions, with non-negligible wind. Results were generated flying multiple directions, both in tailwind and headwind. Energy optimisations generally showed improvements to the magnitudes of between 30% and 50%, while time optimisations were between approximately 20% and 50% over a default solution going straight between the two waypoints at nominal cruise velocity.

This work has demonstrated the potential versatility of hybrid-electric UAVs in long-distance missions, and the potential for increased performance when employing path planning optimisation if knowledge is had regarding the surrounding weather conditions.

References

[1] 'Global unmanned aerial vehicle (uav) market - analysis and forecast 2019-2029', *BIS Research*, 2019. [Online]. Available: `https://www.giiresearch.com/report/bis853981-global-unmanned-aerial-vehicle-uav-market-focus-on.html`.

[2] J. Sliwinski, A. Gardi, M. Marino and R. Sabatini, 'Hybrid-electric propulsion integration in unmanned aircraft', 2017.

[3] E. F. L. Narum, 'Mission planning for fixed-wing uavs in wind and icing conditions (unpublished project assignment)', 2019.

[4] E. F. L. Narum, R. Hann and T. A. Johansen, 'Optimal mission planning for fixed-wing uavs with electro-thermal icing protection and hybrid-electric power systems', *ICUAS*, 2020.

[5] A. R. Hovenburg, 'Flight performance optimization for small unmanned aerial vehicles using path planning methods', PhD thesis, Norwegian University of Science and Technology, 2019.

[6] B. H. Wang, D. B. Wang, Z. A. Ali, B. T. Ting and H. Wang, 'An overview of various kinds of wind effects on unmanned aerial vehicle', *Measurement and Control*, 2019.

[7] J. Osborne and R. Rysdyk, 'Waypoint guidance for small uavs in wind', 2005.

[8] A. Winter, R. Hann, A. Wenz, K. Gryte and T. A. Johansen, 'Stability of a flying wing uav in icing conditions', 2019.

[9] R. Hann, A. W. Wenz, K. Gryte and T. A. Johansen, 'Impact of atmospheric icing on uav aerodynamic performance', in *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Linkoping*, 2017.

[10] S. Armanini, M. Polak and J. e. a. Gautrey, 'Decision-making for unmanned aerial vehicle operation in icing conditions', Dec. 2016. DOI: `10.1007/s13272-016-0215-2`.

[11] J. Schömann, 'Hybrid-electric propulsion systems for small unmanned aircraft', PhD thesis, Technische Universität München, 2014.

[12] O. Tremblay, L.-A. Dessaint and A.-I. Dekkiche, 'A generic battery model for the dynamic simulation of hybrid electric vehicles', *IEEE*, 2007.

[13] J. Y.-C. Hung, 'Investigation of methods for increasing the energy efficiency on unmanned aerial vehicles (uavs)', Master's thesis, Queensland University of Technology, 2011.

[14] F. G. Harmon, A. A. Frank and S. S. Joshi, 'The control of a parallel hybrid-electric propulsion system for a small unmanned aerial vehicle using a cmac neural network', 2005.

[15] U. D. of Transportation, 'Aviation weather advisory circular 00-6b', 2016.

[16] B. E. K. Nygaard, J. E. Kristjánsson and L. Makkonen, 'Prediction of in-cloud icing conditions at ground level using the wrf model', *American Meteorological Society*, 2011.

[17] G. Thompson, P. R. Field, R. M. Rasmussen and W. D. Hall, 'Explicit forecasts of winter precipitation using an improved bulk microphysics scheme. part ii: Implementation of a new snow parameterization', *Monthly Weather Review*, vol. 136, no. 12, pp. 5095–5115, 2008. DOI: `10.1175/2008MWR2387.1`. [Online]. Available: `https://doi.org/10.1175/2008MWR2387.1`.

[18] N. Fajt, R. Hann and T. Lutz, 'The influence of meteorological conditions on the icing performance penalties on a uav airfoil', 2019. DOI: `10.13009/EUCASS2019-240`.

[19] *Qgis*, 2019. [Online]. Available: `https://www.qgis.org/en/site/`.

[20] *Local tangent plane coordinates.* [Online]. Available: `https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates`.

[21] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice.* Princeton University, 2012.

[22] K. L. Sørensen and T. A. Johansen, 'Flight test results for autonomous icing protection solution for small unmanned aircraft', in *Int. Conf. Unmanned Aircraft Systems, Miami*, 2017.

[23] R. Hann, K. T. Borup, A. Zolich, K. Sørensen, H. Vestad, M. Steinert and T. A. Johansen, 'Experimental investigations of an icing protection system for uavs', in *International Conference on Icing of Aircraft, Engines, and Structures, Minneapolis*, 2019.

[24] R. Hann, A. Enache, M. C. Nielsen, B. N. Stovner, J. van Beeck, T. A. Johansen and K. T. Borup, 'Uav icing: Experimental heat loads for electrothermal anti-icing and de-icing', in *Atmospheric and Space Environments Conference*, 2020.

[25] *Hybrid powered aircraft in paris*, 2011. [Online]. Available: `https://www.avweb.com/news/hybrid-powered-aircraft-in-paris/`.

[26] *Energy density of gasoline*, 2003. [Online]. Available: `https://hypertextbook.com/facts/2003/ArthurGolnik.shtml`.

[27] M. Dudek, P. Tomczyk, P. Wygonik, M. Korkosz and P. Bogusz, 'Hybrid fuel cell – battery system as a main power unit for small unmanned aerial vehicles (uav)', *International Journal of Electrochemical Science*, 2013.

[28] *Energy density of hydrogen*, 2019. [Online]. Available: `https://skai.co/hydrogen-details`.

[29] *Fuel cells*, 2015. [Online]. Available: `https://www.energy.gov/sites/prod/files/2015/11/f27/fcto_fuel_cells_fact_sheet.pdf`.

[30] G. Xu, L. Liu and X. Zhang, 'Modelling and performance analysis for low altitude electric uavs', 2016.

[31] S. Gudmundsson, 'A biomimetic, energy-harvesting, obstacle-avoiding, path-planning algorithm for uavs', PhD thesis, Embry–Riddle Aeronautical University, 2016.

[32] R. Eberhart and J. Kennedy, 'A new optimizer using particle swarm theory', 1995.

[33] G. Beni and J. Wang, 'Swarm intelligence in cellular robotic systems', 1993.

[34] *Maritime robotics px-31*, 2019. [Online]. Available: `https://www.maritimerobotics.com/px-31`.

[35] *Oslo universitetssykehus vil frakte blodprøver og livsviktig biologisk materiale med droner*, 2018. [Online]. Available: `https://www.tu.no/artikler/oslo-universitetssykehus-vil-frakte-blodprover-og-livsviktig-biologisk-materiale-med-droner/441111`.

[36] *Kart over norges akuttsykehus*, 2017. [Online]. Available: `https://www.regjeringen.no/no/tema/helse-og-omsorg/sykehus/nasjonal-helse--og-sykehusplan2/nhsp-2015/kart-over-norges-akuttsykehus/id2551362/`.

[37] U. Solies, 'Numerical method for estimation of propeller efficiencies', 1993. DOI: `10.2514/3.46597`.

[38] E. Ranquist, M. Steiner and B. Argrow, 'Exploring the range of weather impacts on uas operations', Jan. 2017.

[39] V. Bonnin, E. Benard, J.-M. Moschetta and C. A. Toomer, 'Energy-harvesting mechanisms for uav flight by dynamic soaring', *International Journal of Micro Air Vehicles*, vol. 7, no. 3, pp. 213–229, 2015. DOI: `10.1260/1756-8293.7.3.213`.

Appendices

A Appendix A

A.1 ECEF to Geodetic coordinate transform

The following is a code implementation in C++ of Ferrari's solution on an ECEF object with members *x*, *y* and *z* to be converted into the geodetic object *geo* with members *lat*, *lon* and *h*:

```
Geodetic ECEF::toGeo() {
double zp, w2, w, r2, r, s2, c2, s, c, ss;
double g, rg, rf, u, v, m, f, p;

Geodetic geo;
zp = abs(z);
w2 = x * x + y * y;
w = sqrt(w2);
r2 = w2 + z * z;
r = sqrt(r2);
geo.lon = (180.0 / PI) * atan2(y, x);
s2 = z * z / r2;
c2 = w2 / r2;
u = a2 / r;
v = a3 - a4 / r;
if (c2 > 0.3) {
s = (zp / r) * (1.0 + c2 * (a1 + u + s2 * v) / r);
geo.lat = (180.0 / PI) * asin(s);
ss = s * s;
c = sqrt(1.0 - ss);
} else {
c = (w / r) * (1.0 - s2 * (a5 - u - c2 * v) / r);
geo.lat = (180.0 / PI) * acos(c);
ss = 1.0 - c * c;
s = sqrt(ss);
}
g = 1.0 - e2 * ss;
rg = a / sqrt(g);
rf = a6 * rg;
u = w - rg * c;
v = zp - rf * s;
f = c * u + s * v;
m = c * v - s * u;
p = m / (rf / g + f);
geo.lat = geo.lat + (180.0 / PI) * p;
geo.h = f + m * p / 2.0;
if (z < 0.0) { geo.lat *= -1.0; }
return(geo);
}
```

A.2 Trilinear interpolation

Trilinear interpolation is used to make three-dimensional discrete data sets continuous. The method is an extension of linear and bilinear interpolation. In this work trilinear interpolation was used to realise a function describing the aerodynamic coefficients C_D and C_L as a function of AOA, MVD and temperature. Data with discrete values for these three parameters was structured as a cube, intended to be made continuous.

The first step is to find the coordinates of the points surrounding the point of interest. Say evaluating the function in point (x, y, z) is desired. The discrete values (x_0, y_0, z_0) and (x_1, y_1, z_1) are then found such that $x_0 < x < x_1$, $y_0 < y < y_1$ and $z_0 < z < z_1$. Let c_{000} indicate the function value of the coordinate (x_0, y_0, z_0), c_{100} (x_1, y_0, z_0) and so on. Then define:

$$\begin{aligned} x_d &= \frac{x - x_0}{x_1 - x_0} \\ y_d &= \frac{y - y_0}{y_1 - y_0} \\ z_d &= \frac{z - z_0}{z_1 - z_0} \end{aligned}$$

Interpolate along the x-axis:

$$\begin{aligned} c_{00} =& c_{000}(1 - x_d) + c_{100}x_d \\ c_{01} =& c_{001}(1 - x_d) + c_{101}x_d \\ c_{10} =& c_{010}(1 - x_d) + c_{110}x_d \\ c_{11} =& c_{011}(1 - x_d) + c_{111}x_d \end{aligned}$$

Then interpolate along the y-axis:

$$\begin{aligned} c_0 =& c_{00}(1 - y_d) + c_{10}y_d \\ c_1 =& c_{01}(1 - y_d) + c_{11}y_d \end{aligned}$$

Finally interpolate along the z-axis:

$$c = c_0(1 - z_d) + c_1 z_d$$

which is the desired value in point (x, y, z).