ISBN 978-82-326-4248-9 (printed version)
ISBN 978-82-326-4249-6 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Fabio Augusto de Alcantara Andrade

Doctoral theses at NTNU, 2019:326

NTNU

Doctoral Thesis

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

Doctoral theses at NTNU, 2019:326

Fabio Augusto de Alcantara Andrade

Real-time and offline path planning of Unmanned Aerial Vehicles for maritime and coastal applications

NTNU
Norwegian University of
Science and Technology

Fabio Augusto de Alcantara Andrade

Real-time and offline path planning of Unmanned Aerial Vehicles for maritime and coastal applications

Thesis for the degree of Philosophiae Doctor

Trondheim, November 2019

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

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-4248-9 (printed version)
ISBN 978-82-326-4249-6 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2019:326

Printed by Skipnes Kommunikasjon as

Fabio Augusto de Alcantara Andrade

Real-time and offline path planning of Unmanned Aerial Vehicles for maritime and coastal applications

Thesis for the degree of Philosophiae Doctor
Trondheim, November 2019

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

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-4248-9 (printed version)
ISBN 978-82-326-4249-6 (electronic version)
ISSN 1503-8181
ITK Report 2019-9-W

Doctoral theses at NTNU, 2019:326

Printed by Skipnes

To my family,
especially my mom Vilma.

Summary

The use of Unmanned Aerial Vehicles (UAVs), popularly known as drones, is growing considerably. This is due to their versatility, reduced cost and rapid deployment, among other advantages. One of the main areas of interest in this field is the development of autonomous UAVs. Inside this area of interest, the topic of path planning was particularly studied in this thesis.

The goal of path planning is to develop algorithms that enable the UAV to perform a given mission by its own, finding the controls that make it achieve the mission goal in an optimal way. Such algorithms can be executed in real-time or off-line. Real-time path planning algorithms are continuously running and acting on the vehicle's motion. Off-line algorithms may be used before the mission starts to plan an optimal path and the required controls to achieve it. The path planning strategy to be chosen depends on the mission characteristics and objectives.

The real-time path planning algorithms studied in this thesis are suitable for missions where the environment or the mission parameters change dynamically. This can be, for example, a moving target of interest or if local real-time weather information is used in the optimization algorithm.

The first solution being presented in this thesis is the use of an autonomous UAV for the surveillance of a ship's trajectory. In this solution, a Model Predictive Control (MPC) solving an optimal control problem (OCP) is designed, including a kinematic model for the ship's movement in addition to the two-dimensional UAV kinematic model, which has the turn rate as the control input. The chosen strategy for the UAV control was to send calculated waypoints based on the optimized turn rate to the Ardupilot autopilot control unit. The mean error between the measured and desired UAV position was reduced by 18% when the ship's kinematic model was used, compared to when it was not used. In the second solution, multiple UAVs are used as relay nodes to maintain the communication link between an Autonomous Surface Vehicle (ASV) and the ground station. The coordinated turn kinematic model including wind effects is used for the UAV and the MPC also includes a kinematic model for the ASV. In the simulations using two UAVs flying at 100 meters of altitude, it was possible to double the Wi-Fi range between the ASV and the ground station, resulting in an operational area four times bigger.

Finally, a third real-time path planning solution using MPC is described, where multiple UAVs are used in a Search and Rescue (SAR) mission. In this solution, the coordinated turn model is adapted to the UAV control unit commands. The solution is developed using CUDA C programming language to be embedded in the UAVs' on-board computers and tested using a Software-In-The-Loop environment including flight dynamics simulated by JSBSim. Results show that three UAVs are able to reach 50% of Probability of Success 2.35 times faster than when only a single UAV is used.

The off-line path planning algorithms studied in this thesis are applied to long range missions, where the UAV is expected to fly from the origin to a given destination. The technique of Particle Swarm Optimization was used to find the values (solution) of the set of variables of the problem.

In the first off-line solution, a hybrid-electric UAV path is optimized so that it flies close enough to a safe landing spot that it is able to reach with remaining battery capacity in case of an engine failure. The range that the UAV is able to fly is calculated by taking into consideration an aircraft performance model and non-uniform forecasted wind maps. In the main simulated scenario, the straight path had a length of 210.0 kilometers and 72.6% of it being close enough to a safe landing spot. The resulted optimized path had a length of 215.4 kilometer (2.57% increase) and 100% of it being close enough to a safe landing spot. In the second solution, the algorithm was modified so that, instead of calculating the range, the energy consumption is calculated in order to find the leveled path that allows optimal energy savings taking into consideration the wind. The optimized path achieved 4.2% of energy savings for the simulated scenario when compared to the straight path. Finally, the third solution is described, where the path is optimized additionally taking into consideration climbing, descent and electro-thermal icing protection system usage, as well as other meteorological parameters. Twelve operational profiles were compared and the proposed solution achieved 52% of battery savings when compared to the standard straight path.

Contents

Summary **iii**

Contents **v**

Preface **ix**

Nomenclature **xi**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Structure of the Thesis and Main Contribution 4

1.3 Publications 7

I Real-time Path Planning with Model Predictive Control **9**

2 Autonomous UAV surveillance of a ship's path with MPC for Maritime Situational Awareness **11**

2.1 Introduction 11

2.2 System Description 14

2.3 Model Predictive Control 17

2.4 Simulated Data 19

2.5 System Configuration 20

2.6 Results 21

2.7 Discussion 22

2.8 Conclusion 23

3 Path Planning of Multi-UAS Communication Relay by Decentralized MPC **25**

3.1 Introduction 25

3.2 Methodology 27

3.3 System description 33

3.4 Simulations 35
3.5 Discussion 37
3.6 Conclusion 38

4 Autonomous Unmanned Aerial Vehicles in Search and Rescue missions using real-time cooperative Model Predictive Control 39
4.1 Introduction 39
4.2 Materials and Methods 43
4.3 Results and Discussion 53
4.4 Conclusions 62

II Long Range Off-Line Path Planning with Environmental Parameters 65

5 Contingency Path Planning for Hybrid-electric UAS 67
5.1 Introduction 67
5.2 Aircraft performance model 68
5.3 Path Planning 71
5.4 Case study and Experimental setup 73
5.5 Results 75
5.6 Limitations 78
5.7 Discussion 78
5.8 Conclusion 79

6 Inclusion of Horizontal Wind Maps in Path Planning Optimization of UAS 81
6.1 Introduction 81
6.2 Aircraft dynamic model 83
6.3 Effects of wind on in-flight performance 84
6.4 Path Planning 87
6.5 Discussion 93
6.6 Conclusion 93

7 Long range path planning using an aircraft performance model for battery powered sUAS equipped with icing protection system 95
7.1 Introduction 95
7.2 Aircraft performance model 98
7.3 Battery performance model 103
7.4 Meteorological and elevation data 104
7.5 Path Planning 107
7.6 Case study 112

7.7 Results 119
7.8 Conclusion 122

III **123**

8 Concluding Remarks and Recommendation for Future Work **125**

Appendices **129**

A Object Classification in Thermal Images using Convolutional Neural Networks for Search and Rescue Missions with Unmanned Aerial Systems **131**

B Particle Swarm Optimization **147**

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU). The research has been carried out from July 2016 to November 2019 at the Centre for Autonomous Marine Operations and Systems (AMOS), Department of Engineering Cybernetics (ITK), NTNU and at Norut Northern Research Institute AS, that lately became part of NORCE Norwegian Research Centre AS. The work was supervised by Associate Professor Rune Storvold and Professor Tor Arne Johansen. To take my PhD, a leave of absence was granted by the Brazilian Government, first the Ministry of Defense and lately the Federal Center of Technological Education of Rio de Janeiro (Cefet/RJ). During my PhD, I was as an early stage researcher part of the Innovative Training Network on Autonomous Unmanned Aerial Systems for Marine and Coastal Monitoring (MarineUAS) funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 642153.

Acknowledgments

First of all, I would like to thank my family that, through all generations, made it possible for me to get all the support. In my most immediate family, I would like to specially thank my mother Vilma Penna de Alcantara, that lived (and lives) much of her life for my growth, my grandfather Fabio de Alcantara, that was my inspiration of love and dedication to the Academia, and my grandmother Edna Marillia da Silva de Alcantara (Tita) and Creuza Maria da Silva (Dadá) who helped Mom to raise me.

I would also like to thank Dr. Rune Storvold and Professor Tor Arne Johansen for the dedication on the supervision and for supporting all my initiatives.

In this process, two long time friends were fundamental: Felipe Cardoso Moreira, that when I was about to give up looking for PhD positions told me to be persistent and Carlos Alberto Moraes Correia that within all the friendship and support all these years, significantly helped me when all odds were against me. They have all my gratitude for that. With regards to the process of applying for the PhD, I would like to thank the Trade Union of Docents of Higher Education Institutions (ANDES-SN) in the person of the former 1st vice-president of Rio de Janeiro Regional, Professor Luis Eduardo Acosta.

I would also like to thank my colleague Professor Lourival José Passos Moreira from the Brazilian Naval Academy that helped me with the PhD application and with my professional development. From Cefet/RJ, I would like to specially thank Associate Professor Aline da Rocha Gesualdi, Professor Carlos Henrique Figueiredo Alves, General-Director of Cefet/RJ, and all my colleagues from the Department of Electronics Engineering of Cefet/RJ that welcomed me and supported my leave of absence request. Also, thank you Luciano Netto de Lima and Diego Barreto Haddad for the contributions to my research.

During the PhD, among all good friends that I made, two are special: Christopher Dahlin Rodin and Anthony Renier Hovenburg, who I would like to thank for the good time together.

Also, I would like to thank my colleagues from Norut (lately NORCE), specially Tenaw Gedefaw Abate, my Brazilian friends in Trondheim and Tromsø, specially Rodrigo Santiago Juaçaba, Francisco Maciel Ferreira, Marcelo Malachias Gomes and Rafael Barros Costa, that made it possible to get a piece of my home country and culture even being thousands of miles away, and my best friend Luiz André Ventura Gosling that talked to me almost every day during the entire PhD time.

Finally, I would like to thank my girlfriend Mai-Helen Steiro that always motivated and supported me.

I'm sorry if I forgot to mention someone. Thank you all!

- Fabio Andrade

Nomenclature

ASV	Autonomous Surface Vehicle
DTM	Digital Terrain Model
ENU	East-North-Up
FBWB	Fly-By-Wire-B
FSPL	Free-Space-Path-Loss
HITL	Hardware-In-The-Loop
HPBW	Half-Power-Bandwidth
IAMSAR	International Aeronautical and Maritime Search and Rescue
IMC	Inter-Module Communication
IPS	Icing Protection System
LWC	Liquid Water Content
MPC	Model Predictive Control
MVD	Water Droplet Median Volume Diameter
NED	North-East-Down
NMPC	Nonlinear Model Predictive Control
OCP	POptimal Control Problem
POC	Probability of Containing
POD	Probability of Detection
POS	Probability of Success
PSO	Particle Swarm Optimization
RGB	Red Green Blue
SAR	Search and Rescue
SITL	Software-In-The-Loop
sUAS	small Unmanned Aerial Systems
UAV	Unmanned Aerial Vehicles
UAS	Unmanned Aerial System

Chapter 1

Introduction

In this chapter, the background and motivation are presented, as well as the structure of the thesis, where it is explained how the chapters are connected. The problems of real-time path planning and off-line long range path planning are also presented.

1.1 Background and Motivation

While the development of Unmanned Aerial Vehicles (UAVs) was boosted by military applications in the last century, recently, in the last decade, the development of UAVs (drones) for civilian applications has grown considerably.

UAVs are versatile tools that can be used in a broad spectrum of missions. The rapid growth of the use of UAVs is justified by their endurance, reduced cost, rapid deployment and flexibility. This flexibility is mainly due to the many types of sensors that can be mounted on UAVs, enabling them to be used in many different applications. In addition, they offer reduced risk for humans and impact on the environment, when compared to manned aircraft.

There are many examples of civilian applications that arose in the last decade, such as border patrol [47]. A report from the Department of Homeland Security's Inspector General states that the US performed 635 missions with UAVs by the U.S. Customs and Border Protection in the 2017 fiscal year, totaling over 5625 hours of flight [107]. Sea surveillance is another common application [140]. UAVs are such an important facility that Coast Guard Commanders do not want to sail without a UAV on board [130]. UAVs are also being used in farms, such as in precision agriculture [95] [67] as well as in spraying [159], to reduce costs and also access areas where trucks cannot go [12]. Mapping/photogrammetry [116] is also an important field where the employment of UAVs is very noticeable, where UAVs equipped with imaging sensors are able to map three-dimensional models of terrain, vegetation and urban areas [117]. Some other interesting initiatives are arising, such as using UAVs for delivery [150] [63] or even Internet providing, such as Facebook's project to bring Internet to remote places using a UAV [126]. Finally, recreational drones are also gaining a lot of attention in the UAV industry [96]. The list of civilian applications is vast and not limited to the ones that were mentioned.

One of the most desired goals in UAV robotics is user-friendliness and intelligence, in other words, these vehicles should have certain autonomy in the performance of tasks to which they are assigned. The navigation of a UAV is one of the most complex tasks related to the control of robots. This complexity occurs as the navigation should integrate sensors, performance, planning, architecture, hardware and computational efficiency. Thus, the integration of all these elements is intrinsic in order to obtain autonomous navigation.

One of the main fields in UAV navigation is path planning, whose objective is to enable the planning of the UAVs movements without human interference. This is a difficult task, especially due to the vehicle constraints and mission complexity.

In path planning problems solved by optimization techniques, a cost function (also called fitness function or objective function) is built with the aim to achieve the mission objective. The optimization algorithm is responsible for finding the solution of a set of path variables that minimizes the cost function, and, therefore, achieves an optimal path. In non-convex problems, which are the most common, the solution found by the optimization algorithm is usually not the optimal solution. However, if proper optimization techniques are used, the found solution may be very close to the best one.

In this thesis, the path planning techniques studied are divided into two categories: real-time path planning; and off-line long range path planning. Each one is applicable to a different set of mission characteristics, which are described in the following subsections.

1.1.1 The Problem of Real-Time Path Planning of Unmanned Aerial Vehicles

Real-time path planning is the most suitable solution for missions where the environment is changing dynamically. For example, in tracking missions [131], where the tracked objective changes its direction in a way that is not possible to know a priori.

Wind is a phenomenon which magnitude and direction can frequently change along a UAV mission. Since wind has a major impact on small UAV motion [13], real-time path planning techniques can benefit from local in-flight wind estimations [15] if the effects of wind are included in the vehicle's model.

In the real-time path planning solutions proposed by this thesis, the technique of Model Predictive Control (MPC) [24] is used for the solution of the optimal control problem (OCP). In this technique, the cost function is designed to take into consideration the predicted outputs of the model along a finite time horizon. Therefore, the optimization algorithm is responsible for finding the values (solution) of the set of OCP variables (control inputs) for the present and future time steps that minimizes the difference between the desired and predicted sequence of outputs (Figure 1.1).

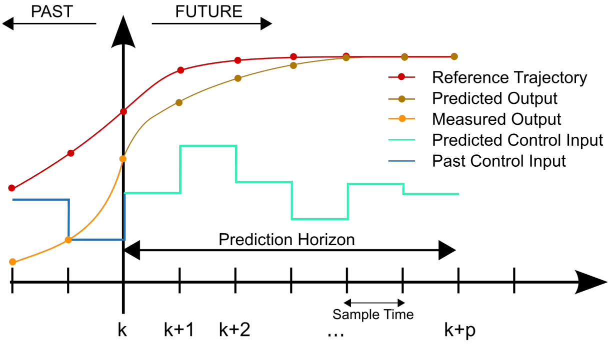


Figure 1.1: A discrete MPC scheme. (Source: Martin Behrendt)

Besides the advantages of dealing with dynamic changes in the environment and finding the control inputs based on a sequence of predicted outputs, MPC has also the capability to directly incorporate vehicle constraints to the model. This is especially interesting for fixed-wing UAVs due to their limited turning radius.

Despite the superior results brought by this approach, it has an increased computational cost. Therefore, a proper optimization technique has to be chosen and the MPC parameters (such as the horizon and number of steps), which are important for the stability of the MPC, as well as the computational cost of the model equations, must be properly addressed. This is fundamental to allow the optimization technique to converge the solution to an optimal value in a time that is compatible with the system requirements. In addition, the development of the algorithms should benefit from the recent advancements in the development of faster processing units, including parallel computing, which is especially favorable for optimization algorithms where the many candidate solutions can be evaluated simultaneously.

1.1.2 The Problem of off-line long range path planning of Unmanned Aerial Vehicles

Certain long range missions, such as missions where the objective is to fly from the origin to a given destination, are usually more suitable for off-line path planning. This is due mainly to the fact that the mission has a defined final state and the optimization algorithm needs to take into consideration the entire geographical domain and not only the region where the UAV could reach in a short time horizon.

The mission also needs to be modelled in large-scale, leaving the low level control within the autopilot control unit. This approach does not compromise the mission because minor deviations will not affect significantly the overall performance, i.e., if the UAV deviates a few meters from the original path, this has minor impact on the cost function if the total path length is in the scale of hundreds of kilometers. In addition, changes in the environment are not expected to be big enough and frequent enough to cause a significant impact on the result.

As mentioned, the effect of wind on small UAV missions is considerable. Therefore, if the wind is used in favor or if the negative impact is mitigated, the mission endurance can be enhanced (i.e. energy consumption reduced). For this to be possible, non-uniform forecasted wind maps must be included in the optimization. Also, in missions with variable altitude, constraints such as the terrain elevation must be addressed, limiting the domain to the region above the ground.

Finally, flying UAVs in icing conditions is challenging and risky if no icing protection system is used. Recent developed systems [134] require a significant amount of electric energy to eliminate the ice on the structure and some solutions employ an alternating usage, turning the system on and off during intervals to reduce the energy consumption. However, this approach degrades the aerodynamic performance of the aircraft, sometimes requiring more propulsive power to compensate for the penalties. Therefore, finding the path that optimizes the use of the icing protection system is also necessary.

In the solutions described in this thesis, the technique of Particle Swarm Optimization (PSO) [35] is used to find the value of the set of variables, such as waypoints, airspeed and climb angles, that minimizes the cost function and, therefore, finds an optimal path. PSO is a meta-heuristic optimization method where the particles (solutions) are updated every iteration based on the best global and local solutions.

1.2 Structure of the Thesis and Main Contribution

This thesis is divided into three parts: the first part presents three applications of real-time path planning; the second part presents three applications of off-line path planning; and the third part presents the concluding remarks and recommendations for future work.

Part I: Real-time Path Planning with Model Predictive Control

The first part has three chapters.

- **Chapter 2:** In this chapter, a mission scenario where the UAV should autonomously overfly a maritime vessel's path is presented. The technique of Model Predictive Control (MPC) is used and the planned path of the vessel is included as a kinematic model in the optimal control problem. This inclusion significantly improves the performance of the system. For the UAV, a basic two dimensional kinematic model with constant speed is used having the UAV turning rate as the control input. The MPC problem is solved by a MATLAB script with the ACADO Toolkit, which uses a quadratic programming solver. After each optimization step, the target waypoints are calculated and sent to the UAV control unit through the IMC Java Protocol library and DUNE framework. The Ardupilot autopilot is responsible for the low level control in order to ensure that the received waypoints are reached by the UAV. The proposed solution was simulated in a Software-In-The-Loop environment including flight dynamics model simulations.

- **Chapter 3:** In this chapter, a multiple UAVs solution for maintaining the communication link between a moving Autonomous Surface Vehicle (ASV) and the ground station is presented. As it is done in the previous chapter, this chapter's system also includes the vessel's kinematic model in the MPC problem. Also, to achieve better performance and reliability, a new UAV kinematic model is developed in order to use the autopilot's control unit mode Fly-By-Wire-B, where the airspeed and roll angle are directly commanded, instead of only waypoints such as the system described in the previous chapter. The UAV kinematic model is, therefore, upgraded to the coordinated turn model, where the control inputs of the optimization problem are the acceleration and roll angle rates. Since in the Fly-By-Wire-B mode the attitude of the UAV is controlled by the commands obtained by the optimization algorithm, the effects of wind are also included in the model. In addition, because of the mission characteristics, the power consumption is also included in the cost function. Finally, the solution is implemented in CUDA C embedded programming language with the objective to benefit from parallel processing and prepare the solution to run in the UAV's on-board computer. Numerical simulations for two different scenarios are evaluated.

- **Chapter 4:** The solution presented in this chapter modifies the previous chapter solution so that the control inputs of the MPC problem are the same as the commands that are accepted by the autopilot control unit. Therefore, the airspeed and roll angle are chosen as the new optimization problem control inputs. The system is fully integrated to the DUNE platform using CUDA C programming language. Software-In-The-Loop simulations are evaluated, including flight dynamics model simulations. The solution is applied to a Search and Rescue (SAR) mission, where multiple UAVs search a given area for survivors. The mission is defined following international SAR directives. Five missions are simulated for each of the three operational profiles: using one; two; or three UAVs. The average of the results are compared and discussed.

Part II: Long Range Off-Line Path Planning with Environmental Parameters

The second part of the thesis presents three path planning problems that are solved using Particle Swarm Optimization.

- **Chapter 5:** In this chapter, a path planning solution is presented to find the path that a hybrid-electric UAV should follow in order to be always (or as much as possible) close enough to a safe lading spot. This is due to the fact that it is desired that the UAV is able to reach a landing spot in case of an engine failure. Therefore, the optimization algorithm evaluates if the distance between the steps of the candidate solution to the landing spots are within the maximum range of the UAV, which is calculated considering variable airspeed commands that give the maximum endurance considering non-uniform wind maps. The technique of Particle Swarm Optimization (PSO) is used to find the optimal path.

- **Chapter 6:** In order to find an optimal cost-efficient two dimensional path between two points, the equations used in the previous chapter are modified so that the power consumption per distance traveled can be calculated for different commanded airspeed and wind, instead of calculating the range based on the commanded airspeed and wind as presented in the previous chapter. Therefore, it is possible to find a path that uses the wind and the aerodynamic performance characteristics in order to reduce the energy consumption and, consequently, increase the endurance.

- **Chapter 7:** In this chapter, the solution presented in the previous chapter is improved by including altitude changes in the aerodynamics performance model. This results in a quasi-three-dimensional (multiple altitudes) path, where the variables being optimized are the airspeed, climb angle and waypoints. Meteorological maps of different parameters and multiple altitudes were obtained from a meteorological service provider. In addition, the use of electro-thermal icing protection systems is introduced in order to allow the UAV to fly under icing conditions and to optimize its energy consumption. The icing conditions, as well as the power required by the icing protection system and its effects on the aerodynamics are calculated based on the meteorological parameters. The meteorological parameters downloaded and considered in the calculations are horizontal winds, air temperature, humidity and liquid water content.

Part III contains the final chapter, where the concluding remarks are presented, as well as the recommendations for future work.

1.3 Publications

This thesis is based on results that were published or submitted for publication in conferences and journals in collaboration with colleagues. The publications that are included in this thesis, and other related publications, are listed in the following.

Publications included in the thesis:

- [7] **F. A. A. Andrade**. Route planning of unmanned aerial vehicles for maritime situational awareness. *Revista de Villegagnon (English Edition)*, 10:146-150, 2015.
    - Parts of this publication related to the automation of UAVs and path planning are included in Chapter 1. Parts that describe the term Maritime Situational Awareness are included in Chapter 2.
- [8] **F. A. A. Andrade**, R. Storvold, and T. A. Johansen. Autonomous UAV surveillance of a ship's path with MPC for maritime situational awareness. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 633–639. IEEE, 2017.
    - Parts of this publication that present a literature review about path planning and Model Predictive Control for UAVs and parts that describe the developed system and its results and conclusions are included in Chapter 2. Parts of the conclusion are also included in Chapter 8.
- [70] A. R. Hovenburg, **F. A. A. Andrade**, C. D. Rodin, T. A. Johansen, and R. Storvold. Contingency path planning for hybrid-electric UAS. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 37–42. IEEE, 2017.
    - Parts of this publication that present a literature review about path planning, as well as parts that describe the developed system and its results are included in Chapter 5. Parts of the conclusion are included in Chapter 8.
- [9] **F. A. A. Andrade**, C. D. Rodin, A. R. Hovenburg, T. A. Johansen, and R. Storvold. Path planning of multi-UAS communication relay by decentralized MPC. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*,pages 1–8. IEEE, 2018.
    - Parts of this publication that present a literature review about Model Predictive Control for problems with multiple UAVs, as well as parts that describe the developed system and its results, are included in Chapter 3. Parts of the conclusion are included in Chapter 8.
- [72] A. R. Hovenburg, **F. A. A. Andrade**, C. D. Rodin, T. A. Johansen, and R. Storvold. Inclusion of horizontal wind maps in path planning optimization

of UAS. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 513–520. IEEE, 2018.

  - Parts of this publication that present a literature review about path planning using atmospheric wind and aircraft performance model, as well as parts that describe the developed system and its results, are included in Chapter 6. Parts of the conclusion are included in Chapter 8. Parts of the publication's Appendix are included in this thesis Appendix.

- [10] **F. A. A. Andrade**, A. R. Hovenburg, L. N. Lima, C. D. Rodin, T. A. Johansen, R. Storvold, C. A. B. Correia, and D. B. Haddad. Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control. *Sensors*, 19(19):4067, 2019.
  - Parts of this publication that present a literature review about autonomous search and rescue with UAVs and Model Predictive Control for multiple UAVs in exploration missions, as well as parts that describe the developed system and its results, are included in Chapter 4. Parts of the conclusion and future work are included in Chapter 8.
- [69] A. R. Hovenburg, **F. A. A. Andrade**, R. Hann, C. D. Rodin, T. A. Johansen, and R. Storvold. Long range path planning using an aircraft performance model for battery powered sUAS equipped with icing protection system. Submitted for publication.
  - Parts of this publication that present a literature review about UAV path planning techniques and parts that describe the developed system and its results are included in Chapter 7. Parts of the conclusion and future work are included in Chapter 8.

Publications included in the Appendix:

- [121] C. D. Rodin, L. N. de Lima, **F. A. A. Andrade**, D. B. Haddad, T. A. Johansen, and R. Storvold. Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems. In *2018 International Joint Conference on Neural Networks(IJCNN)*, pages 1–8. IEEE, 2018.

Other related publication not included in the thesis:

- [120] C. D. Rodin, **F. A. A. Andrade**, A. R. Hovenburg, T. A. Johansen, and R. Storvold. A survey of design considerations of optical imaging stabilization systems for small unmanned aerial systems. Submitted for publication.

Part I

Real-time Path Planning with Model Predictive Control

Chapter 2

Autonomous UAV surveillance of a ship's path with MPC for Maritime Situational Awareness

Maritime Situational Awareness is crucial in maritime operations to identify threats and to deal with them as soon as possible. These threats can be pirates in shipping operations, icebergs when sailing in the northern sea routes, or even unknown vessels or objects that might be on the ship's path. A solution to identify these threats is the use of UAVs to overfly the ship's planned path. This solution is described in this chapter, using an autonomous fixed wing UAV. Based on the provided ship's planned path, the UAV should autonomously fly over the area around to the ship track. To do that, an optimization problem is solved using Model Predictive Control, where the turn rate for the next time period is optimized. Based on the turn rate, the future path of the UAV is calculated and the waypoints are sent to the autopilot. This application is thoroughly tested using a Software-In-The-Loop environment, where flight dynamics model simulations are combined with the autopilot's commands. The results show that the solution is suitable and that the performance is significantly improved if the UAV has information about the ship's velocity in addition to position.

2.1 Introduction

After the tragedy of 11 September 2011, while the United States sought to improve the safety and security of its citizens, arose in that country the concept of Maritime Domain Awareness (MDA). The term had already been used earlier [87], but the concept had not been defined yet. This "maritime awareness" implied in having timely and reliable information on all areas and things of, on, under, relating to, adjacent to, or bordering on a sea, ocean, or other navigable waterway, including all maritime, related activities, infrastructure, people, cargo, and vessels and other conveniences [108]. Thus, the concept of MDA is the effective understanding of any facts, associated to the maritime domain, which can impact the security, defense, the economy and the environment. At the end of 2006, as a result of the Summit meeting of the North Atlantic Treaty Organization (NATO) in Riga, the term Maritime Domain Awareness (a concept which was originally American) evolved into Maritime Situational Awareness [18], reflecting an international concept which was more appropriate to the definitions used by different countries.

Nowadays, many illegal agents' actions are observed in the maritime and coastal environment, for example, piracy, which has returned because of the inability and weaknesses to confront it. Besides, there are also established threats such as terrorism, transnational crimes, illegal immigration and environmental disasters caused by pollution and catastrophes. Maritime and coastal monitoring also includes mapping and monitoring of marine resources, environmental parameters related to the oil spills, climate indicators, icebergs and sea ice in arctic regions, in addition to inspection of fisheries, surveillance for security, oceanography, data acquisition from buoys, surface ships and underwater vehicles, assistance in search and rescue missions, and situation awareness in marine operations, among others.

Thus, the purpose of Maritime Situational Awareness is to develop the ability to identify existing threats as soon and as far as possible, by the integration of intelligence information, surveillance, observation and navigation systems, interacting in the same operational framework. For this capability to be effective, it is necessary a structure that allows data collection, monitoring, air and naval sensors, and the correct analysis of the facts, enabling a quick and accurate response.

To build this capability, UAVs are well suited tools that can be equipped with a wide variety of sensors, such as cameras or radars. The cameras can be infrared, RGB, or multispectral, for example. One application of these sensors is the tracking of floating elements in the sea [91], like drifting objects, enemy vessels, icebergs and others.

This information can be used in collision avoidance systems [78], where the vessel finds a new path based on the obstacles' positions; military operations [7], where the threats or targets can be identified earlier; ice management [64], to map the features of ice or icebergs drifting into the path of the vessel; and search and rescue [115], where the vessel needs to find missing crew.

In seismic data collection, vessels tow arrays of streamers that might be up to one kilometer wide and 5 kilometers long. When towing these arrays, the vessel must keep steady direction and speed, hence to prevent damage to the streamers the area upstream must be inspected for icebergs and growlers and in case of suspected damage to streamers the array and cables themselves need to be inspected while moving [152].

Usually, the vessel is the center of the maritime operation, which, as exemplified before, can be a shipping mission, a search and rescue, or a military surveillance operation. To collect the needed data to support the chosen operation, the UAV must fly over a region according to the vessel's planned path and speed, and in the case of scouting for growlers, combined with knowledge on i.e. potential drift velocities, in a cooperative way ensuring safety margins on the captured data.

Therefore, in the scenario proposed in this chapter, a fixed wing UAV must fly over the ship's planned path at a certain distance ahead of the vessel to allow the ship's captain to timely react to findings. For instance, if it is needed some information 4 minutes in advance, the UAV has to overfly the predicted position of the ship 4 minutes ahead. For the UAV to decide its optimum path to cover that area, an optimization problem has to be solved to minimize the error between the predicted and the desired position of the UAV. To solve this problem, using Model Predictive Control (MPC) is beneficial, because it can consider the predicted output, based in current measurements, to fit the control inputs in a better manner. In this problem, the intended velocity of the vessel can be used to predict its future positions in order to decide the UAV's optimum path.

There are several research describing the use of MPC techniques in the field of UAVs, such as in [81], where a UAV was used to follow a linear path. However, it did not take into account time constraints, so the problem was treated as a path following problem. [65] also shows a solution to track moving objects, but it optimizes the path according to the waypoints sequence, which has much more computational cost than the solution presented in this chapter, where only the turn rate is controlled. In [131], a MPC algorithm running in a ground station is used to control the UAV turning radius and the camera's gimbal to track objects in the sea. The solution proposed in this chapter brings a similar approach but to a different problem.

The system presented in this chapter, with the challenge of integrating different software components and to make it possible to use it in an on-board real time optimization problem, introduces an effective solution for many types of autonomous tracking problems, bringing a beneficial result, especially when the motion of the tracking object is provided.

The UAV's path is optimized by solving a MPC problem where the turn rate is the control input. The UAV is assumed to fly at a constant altitude. The optimization problem is implemented using ACADO for MATLAB, a user-friendly interface of the ACADO Toolkit [68], which is a software environment and algorithm collection for automatic control and dynamic optimization, implemented in C++. MATLAB is used to hold the main software, which does the integration between the simulated ship's data, the MPC and the Ardupilot autopilot.

ArduPilot [141] is an open source autopilot that supports many types of robotic vehicles, including the fixed wing UAV platform used in this application. It is possible to send commands to the ArduPilot or to read the UAV sensors via Micro Air Vehicle Communication Protocol (MAVLink) [142]. The ArduPilot has also a Software-In-The-Loop simulator that can use a flight dynamics model simulator to simulate the UAV's behavior to the actuator commands and the sensors data. The JSBSim [17] was used for the flight dynamics model simulations for the X8, a fixed wing UAV which can fly up to 2 hours. To communicate with the autopilot, the messages travel between MATLAB and the ArduPilot though DUNE [112], which is an open source robot framework developed by the Underwater Systems and Technology Laboratory (LSTS) of the University of Porto. The Inter-Module Communication Protocol (IMC) is used for the messages between MATLAB and DUNE using the IMC Java library for MATLAB.

Details of the system architecture are described in the next section.

2.2 System Description

Figure 2.1 shows the block diagram with the connections between all software components used in this system.

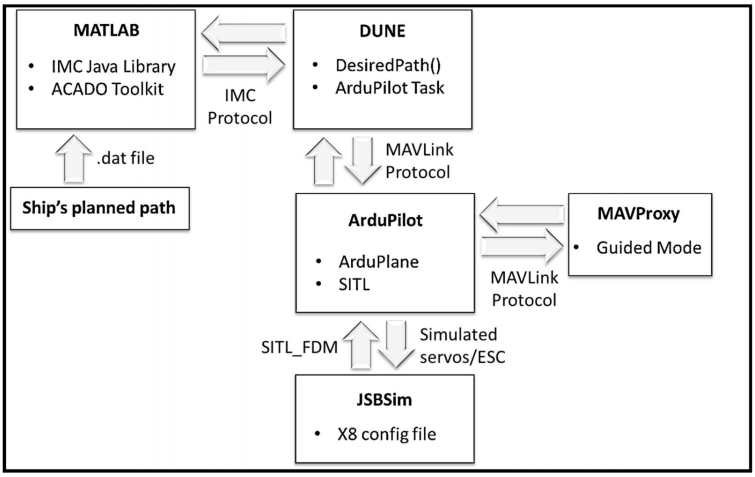


Figure 2.1: System's block diagram, simulation setup.

The main input is the simulated ship's planned path, where the position and velocity in the NED coordinates frame along the time were saved into a MATLAB data file for simulation purposes. In a real mission, the ship's planned path is expected to be sent to the UAV in real-time. MATLAB also receives information from DUNE about the UAV state and uses it, combined to the ship's planned path, to solve the optimization problem and send the waypoints to DUNE. DUNE, in its turn, sends the waypoints to the ArduPilot and gets the information about the UAV sensors. The ArduPilot commands the UAV, which has its flight dynamics simulated by JSBSim. Besides, a MAVProxy ground station is also used to control the UAV in GUIDED mode, if needed.

2.2.1 MATLAB Core Code

The data file is loaded by the core code and the starting position of the ship is taken as the origin of the NED frame. Therefore, the correspondent reference pair of latitude and longitude is defined. In sequence, the update rate of the waypoints is defined and also the time in advance when the ship should receive information from the UAV about its path. This time is used to calculate where the ship might be according to its planned path and then use that information as the desired location where the UAV should overfly.

Position, velocity and attitude of the UAV are also necessary to be used as inputs to solve the two-dimensions optimization problem, which assumes that the UAV will fly at a constant altitude. Therefore, the following information about the UAV is gotten from DUNE:

- Origin (reference) of its local NED frame (latitude, longitude, altitude);
- Position offset (x, y, z) from its NED frame origin;
- Body-Fixed frame 2D linear velocities (u, v);
- Euler angles (Roll (ϕ), Pitch (θ), Yaw (ψ)); and
- Angular velocities over body-fixed frame (p, q, r).

As the starting position of the ship is used in this application as the NED frame reference, it is needed to convert the UAV position in its local NED frame (from DUNE) to the application's NED frame. To do that, first it is made a conversion to geodetic coordinates (latitude and longitude) using the reference of the UAV's Local NED frame from DUNE and then it is made a conversion to the application's NED frame using the ship's starting position as the reference. This is necessary because, in DUNE, the reference of the UAV's Local NED frame is updated every kilometer of distance that it moves from the former defined reference. Besides, for all conversions, the World Geodetic System of 1984 (WGS 84) was used as the reference ellipsoid.

Regarding the yaw rate, as a two-dimensional model is used in this optimization problem and the body-fixed angular velocity r, which is received from DUNE, is not the rate referring to the UAV's yaw angle, the UAV's yaw rate has to be calculated using the relationship between the Euler-angle rates vector $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^\intercal$ and the body-fixed angular velocity $[p, q, r]^\intercal$ as Eq. 2.1 [137].

This is also needed because the UAV used in this application does not have a rudder and uses the banking to turn. Therefore, in the two-dimensional model described in the next section, the yaw rate r used is actually the yaw rate $\dot{\psi}$ obtained from Eq. 2.1.

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \sec\phi\sec\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{2.1}$$

Finally, after solving the optimization problem, the target waypoint is sent to DUNE using the "DesiredPath" function. In that function it is also possible to send a second waypoint as a backup that can be used in case there is a temporary communication problem.

The overall description of the Core Code is shown in Figure 2.2. The program repeats every chosen update time step and it runs until the last planned position of the ship is reached by the UAV, when the UAV starts to loiter and wait for a guided command.

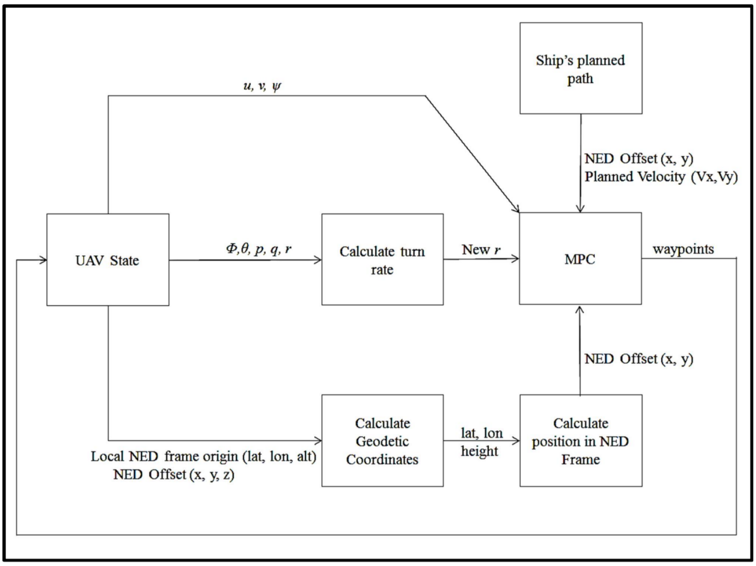


Figure 2.2: Overall Core Code description.

2.2.2 IMC Java library

To send or receive IMC messages, the IMC Java library is required to provide the necessary functions used by the MATLAB script. Besides, in the IMC protocol, the communication is done between nodes. Therefore, the core code has to be started as a IMC node and then connects to the desired node, in this case, the X8 UAV node in DUNE.

2.2.3 DUNE

DUNE is compatible with many different autopilots. In this application, as the ArduPilot is used, when DUNE receives the “DesiredPath()” command, it calls the Task responsible for the communication with the ArduPilot. However, the standard Task available in DUNE’s repository operates the UAV in GUIDED Mode. In that mode, the UAV starts to loiter when it reaches a certain distance from the UAV and never passes through the waypoint. This behavior would make impossible the proposed application.

Therefore, a modification was made in DUNE’s ArduPilot Task in the source code, so that the UAV can be operated in AUTO Mode. In the new Task, a mission containing the waypoint is sent to the autopilot. After the acknowledgement from the autopilot, a command to start the mission is then sent by DUNE.

2.2.4 ArduPilot Software In The Loop

In the Software-In-The-Loop mode, the ArduPilot needs a flight dynamics model simulator. In this application, JSBSim is used to simulate the behavior of the UAV in a real flight. JSBSim simulations are done taking into consideration the mass balance, ground reactions, propulsion, aerodynamics, buoyant forces, gravity, external forces and atmospheric disturbances, such as winds, gusts, turbulence, downbursts etc. In this application, the X8 UAV (Figure 2.3) model [52] was chosen.

Figure 2.3: X8 UAV. (Source: NTNU)

2.3 Model Predictive Control

Model Predictive Control (MPC) is a class of techniques to solve numerical optimization problems, controlling the input variable to minimize the deviations between a desired output and the predicted output in a finite time-horizon [24].

In this application, the first assumption to be made is that the fixed wing UAV is controlled by an Autopilot system, which needs to be fed with the waypoints of the optimal flight trajectory. The MPC module is responsible for calculating this trajectory, based on the UAV’s and ship’s attitude, position and velocity.

2.3.1 UAV kinematic model

Considering that the fixed wing UAV will fly in a constant altitude, maintained by the Autopilot, its kinematic is treated as a two-dimensional motion problem (Figure 2.4).

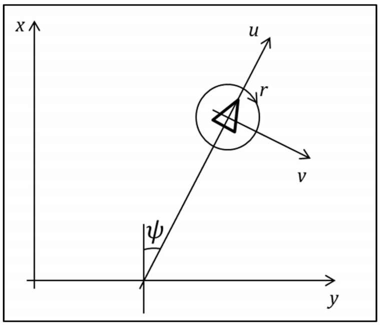


Figure 2.4: UAV model in two dimensions

Therefore, the position and heading of the UAV can be expressed in the NED frame as:

$$\boldsymbol{p}_{UAV} = [x_{UAV}, y_{UAV}, \psi_{UAV}]^{\intercal}, \tag{2.2}$$

where x_{UAV} and y_{UAV} are the horizontal positions and ψ is the yaw angle. The velocities in the BODY frame and the rate of change of the yaw angle forms the following vector.

$$\boldsymbol{v}_{UAV} = [u_{UAV}, v_{UAV}, r_{UAV}]^{\intercal}, \tag{2.3}$$

Besides, the relation between Eq. 2.2 and Eq. 2.3 is given by:

$$\boldsymbol{p}_{UAV} = \mathbf{R}_z(\psi)\boldsymbol{v}_{UAV}, \tag{2.4}$$

where $\mathbf{R}_z(\psi)$ is the rotation matrix between the BODY and NED frames as shown in Eq. 2.5.

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

The UAV's turning rate (r) is the control input of the optimization problem. Therefore, it is necessary to define a maximum value for it as a problem constraint.

2.3.2 Ship's kinematic model

To use the ship kinematic in the MPC, a linear motion model is implemented based on the position and velocity of the ship in its intended path, at the instant the UAV should fly over that location.

$$\boldsymbol{p}_{ship}^{t+\tau} = \boldsymbol{p}_{ship}^{t} + \boldsymbol{v}_{ship}^{t}\tau, \tag{2.6}$$

where $\boldsymbol{p}_{ship} = [x_{ship}, y_{ship}]^\intercal$ is the position of the ship on the path, τ is the time between measurements and $\boldsymbol{v}_{ship} = [v_x, v_y]^\intercal$ the velocity vector.

2.3.3 Cost Function

The Least Squares (LS) function (Eq. 2.7) is used as the cost function to be minimized by the MPC algorithm.

$$J_t = \frac{1}{2}\sum_{i=t}^{t+T} \|h(\boldsymbol{z}_i)\|^2, \tag{2.7}$$

where $\boldsymbol{z}_i = [x_{UAVi}, y_{UAVi}, x_{ship_i}, y_{ship_i}]^\intercal$ is the state vector.

In this application, $(h(\boldsymbol{z}_i))$ is the distance between the current position of the UAV and its desired position, which is equivalent to the ship's future position in its planned path:.

$$h(\boldsymbol{z}_i) = \sqrt{(x_{UAVi} - x_{ship_i})^2 + (y_{UAVi} - y_{ship_i})^2}. \tag{2.8}$$

2.4 Simulated Data

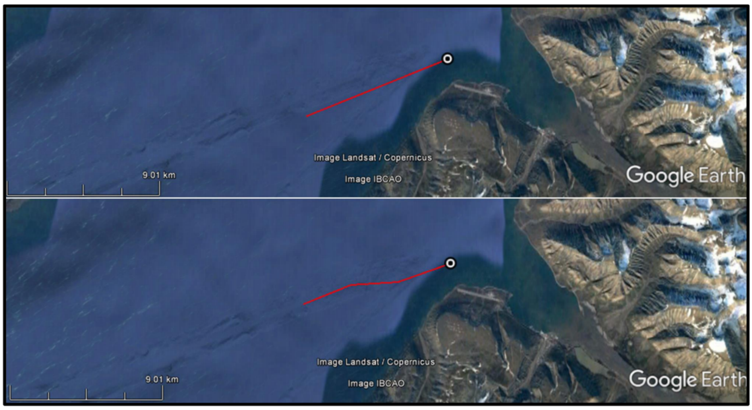


Figure 2.5: Two simulated ship's path

Two simulated datasets with the ship's path were generated. Both have the duration of 15 minutes.

The first dataset was generated for a fixed heading of $-2\pi/3$ rad and fixed velocity of 18 knots.

The second has the same starting heading and velocity of the first one but it changes the direction two times and the velocity one time. After going straight for some time, it makes a slight turn to the right, then it goes straight for a while and makes a final turn to the left, going straight for the rest of the time. The velocity changes from from 18 knots to 21 knots in the middle of the path.

Both datasets were generated for a simulation of a ship navigating close to Longyearbyen, Svalbard (Figure 2.5).

2.5 System Configuration

The following parameters were chosen for the application:

- 2 minutes between the ship's current position and the desired position in its planned path to be overflown;
- 4 seconds for the update period of the UAV waypoints;
- 20 seconds of MPC Horizon;
- 10 MPC input steps;
- 6 seconds between the current position and waypoint;
- Maximum of 30 iterations in the MPC optimization;
- Maximum turn rate of 0.3 rad/s;
- 18 m/s of UAV velocity; and
- 500m of altitude.

The reason to choose 4 seconds between each update is that if the time step is too small, the autopilot may set the waypoint as reached before the next update.

The cruise speed of the X8 (18 m/s) was chosen to be the target constant velocity. For a time of 4 seconds between updates, this results on 72 meters between updates if the UAV is flying straight.

Besides, the target waypoint is set to be 6 seconds from the UAV's current position, what means that it is expected that the waypoint is updated about 36 meters before the UAV reaches the targeted one. This choice was made because if the system wanted to wait the UAV to reach the waypoint or if it allowed the UAV to get very close to the waypoint, the UAV would start to sorely turn trying to reach the waypoint or even it could start to loiter around the waypoint without ever reaching it. This kind of behavior is harmful for the optimization, which should control the UAV in a smooth way.

For the MPC parameters, a maximum number of 30 iterations, 10 input steps and 20 seconds of horizon were chosen to meet the desired update period of 4 seconds with the available hardware. This gives a sampling time of 2 seconds. However, these parameters can be changed and fine-tuned depending on the processing unit being utilized.

The maximum turn rate of the X8 with the standard tuning is around 0.4 rad/s. However, in this optimization problem a lower value of 0.3 rad/s was chosen as the maximum turn rate. This conservative choice was made to avoid sending waypoints that would require too aggressive maneuvers and that could be impossible to reach due to delays between the waypoints message sent by DUNE and the mission acknowledgement by the Ardupilot. The fact that the X8 UAV banks to turn is an additional limiting factor, because if the UAV is turning right and a waypoint in its left is suddenly sent, the UAV has to bank all the way from one side to the other to be able to change the direction of the turn.

2.6 Results

Figure 2.6 shows the path and the distance error of a UAV following the simulated ship's path that has constant velocity and heading. The ship kinematic model was considered in this simulation, using a constant velocity ship trajectory prediction. In the graph on the left, the ship's path is represented as a blue dotted line while the UAV's path is represented in a solid red line. The "x" marker represents the origin of the ship. The right graph shows the error between the current UAV position and the desired one, according to its variation along the path (q from 0, which indicates the start of the mission, to 1, which indicates the end of the mission).

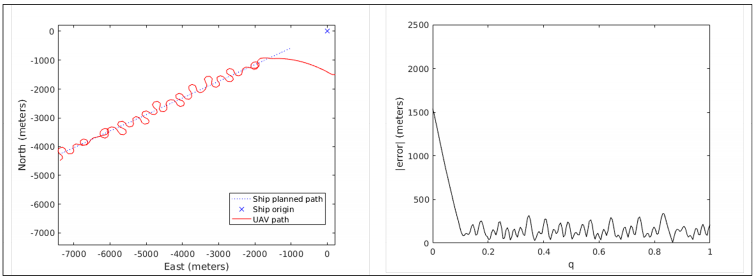


Figure 2.6: UAV path and error for constant ship velocity and heading using the ship model in the MPC.

It is possible to notice that the UAV tries to overfly the path but as its velocity is higher than the ship's, the UAV has to maneuver, crossing the path many times doing turns. The mean error was 217 meters in this case.

When the ship's kinematic model is not used in the MPC (Figure 2.7), the UAV does many 360 deg turns. This can be explained by the absence of the motion of the ship in the model. Therefore, the system does not know that the ship is moving forward and interprets that the desired position is fixed over time. The error in this case was 265 meters, compared to 217 meters of the previous case when the ship's model was considered, i.e., an improvement of 18% was achieved when considering the ship's kinematics in the model.

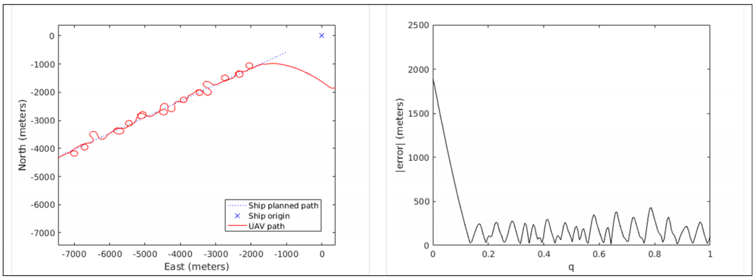


Figure 2.7: UAV path and error for constant ship velocity and heading without using the ship model in the MPC.

Even if the ship's path is not straight and if its velocity changes along the path, the UAV had a good performance keeping the error as small as when the steady ship's path was considered, as shown in Figure 2.8. This shows the ability of the MPC solution to react very well to dynamic changes.

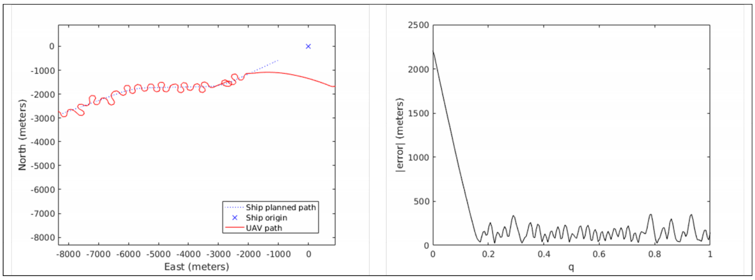


Figure 2.8: UAV path and error for varying ship velocity and heading using the ship model in the MPC.

2.7 Discussion

As this application was developed to run in an on-board computer, the Software-In-The-Loop simulation is a viable way to test the integration of the different systems. E.g., communication delays and protocol challenges are already being faced here. On the other hand, hardware processing performance may only be taken into account with a Hardware In Loop simulation, however, as the presented solution was running in a MATLAB graphic interface installed in a laptop, it will certainly achieve better performance if a dedicated on-board computer running a compiled software written in embedded programming language is used. Therefore, it would be possible to fine-tune the MPC parameters, having more MPC steps and also more iterations to achieve better results.

Besides, it was noticed that just sending waypoints to the autopilot is not the best approach to guide it. It would probably be better if the guidance is done by changing the UAV's attitude and velocity, for example, using the Fly-By-Wire mode, where the controls are sent as the UAV's remote control was used. That new solution would probably demand a more complex model to be used in the MPC, however, the benefits have the potential to overcome the additional complexity.

About the mission scenario, during the development of this solution, it was glimpsed that it would be more useful if the UAV flew over an area around the ship's planned path instead of overflying only the planned path as a line. Thus, it would be possible to detect objects that could be moving into the ship track [78]. To implement this area surveillance feature, a solution similar to the one described in section 4.2.4 may be used.

2.8 Conclusion

In a cooperative mission with a ship and a UAV, a common use for the UAV is to use it to overfly an area of interest. In this scenario, this is the area around the ship's planned path. Therefore, in this application it has been defined that the UAV should overfly the position that the ship is expected to be in 2 minutes from the current time. In the optimal control problem, the UAV should control its turning rate to keep the distance from that position as minimum as possible. Then, according to the optimized turn rate, waypoints are calculated to feed the autopilot.

The path planning system using Model Predictive Control has been shown to be very effective for the proposed problem. The UAV could keep its position very close to the ship's path most of the time, sometimes moving away from it with strategic turns to get back to an optimum position. For the second dataset, where the ship changes its velocity and heading over time, the system also achieved a good result showing that the MPC can adapt to dynamic changes of the ship's behavior.

The difference between the results with and without using the ship's model in the MPC also proved that it is fundamental to use the ship's kinematic model to achieve better performance.

Besides, more simulations can be done for fine tuning, such as changing the number of steps per second. Another suggested improvement is to use hardware in loop simulation aiming flight tests.

Chapter 3

Path Planning of Multi-UAS Communication Relay by Decentralized MPC

When using Autonomous Surface Vehicles (ASV) in marine operations, long distances and/or low power transmissions may severely limit the communication between the ASV and the ground station. One solution to overcome this obstacle is to use a group of small Unmanned Aerial Systems (UAS) to act as relay nodes, in order to provide a user-defined minimum communication capability. To achieve this, a decentralized cooperative multi-agent system using fixed-wing UAS with nonlinear model predictive control is proposed, which aims to guarantee a desired signal strength between the ASV and the ground station. The novelty of the presented research resides in the inclusion of the aircraft performance model and the effects of wind, together with the inclusion of the directivity of the antennas. Experimental results of the proposed method are obtained through simulations.

3.1 Introduction

Using an Autonomous Surface Vehicle (ASV) in maritime missions brings forth the need for a reliable communication link with sufficient signal strength between the ASV and the ground station. Although in some cases a direct link can be used, it is often severely limited in range and affected by the local geography. Satellite communication can manage longer distances, but cannot always be used due to partial satellite coverage, limited bandwidth or the high associated costs. An alternative solution is to use autonomous fixed wing Unmanned Aerial Systems (UAS) to act as communication relay nodes in order to establish a communication link between the ASV and the ground station. By strategically coordinating the

UAS' trajectories, the obtainable communication link can reach the desired signal strength over a larger distance.

In recent scientific literature, there are several approaches to solve similar problems. Grancharova et al. [51] used multiple rotary wing UAS to form a communication network between a base station, a stationary target and a moving target. The author used RF Signal Propagation, Loss, And Terrain (SPLAT!) for calculating the communication path losses. The path losses were approximated to linear functions, and a Model Predictive Control problem was solved by quadratic programming. Johansen et al. [79] describes the use of one fixed wing UAS as a communication relay node between a ground station and an Autonomous Underwater Vehicle (AUV), where the AUV was positioned at the ocean surface. In this reference, horizontally omni-directional antennas were used. However, in their experiment, due to the vertical directionality of the antenna beam, the roll angle of the UAS affects the communication signal power. Kim et al. [84] used multiple autonomous fixed wing UAS as communication relay nodes for a fleet of vessels using a decision making algorithm to choose the waypoints which could satisfy Dubins trajectories and lead to a configuration where the range between the nodes will be less than a specified minimum communication range. The solution was improved in [85], where the paths of the UAS were optimized using Nonlinear Model Predictive Control (NMPC), and the network connectivity was modelled in the context of Mobile Ad hoc NETworks (MANETs) based on global message connectivity. In this reference, the change on the directivity of the antenna pattern due to the effect of the attitude of the UAS was not considered. This made it simpler to model the communication characteristics, but also made it less realistic. In addition, the method does not consider the effects of wind or the power consumption of the UAS. Braga et al. [20] optimized the communication Quality of Service, considering the power consumption and the bandwidth. Here, a simplified power consumption estimation was used, and, again, the wind and the UAS attitude effect on the radiation pattern were not considered. In recent studies, Palma et al. [110] performed field experiments using a UAS as a data mule, i.e. the UAS was used to download and offload data sequentially, rather than to uphold a data link. Different protocols were tested where it is shown that the quality of the communication depends on the protocol being used. The author also states that the combination between the altitude and loiter radius, which determines the angle between the nodes, has a significant effect on the efficiency of data transfer due to the antenna-radiation patterns. Therefore, it is fundamental to define flight trajectories whilst taking this into consideration.

Dixon and Frew [32] used a decentralized algorithm based on the gradient of Signal-to-Noise Ratio (SNR) measurements to obtain a cascaded communication chain between a control station and a moving vehicle. A drawback of their method is the limitation in the vehicle dynamics – a cyclic motion is required in order to obtain an estimate of the SNR – which can result in a non-optimal path. The authors do not consider power consumption of the vehicles, resulting in a functional communication relay, but with a reduced duration compared to an algorithm that considers the power consumption, and has more freedom in vehicle dynamics. Frew and Brown [43] considered a meshed network of UAV relay nodes. Experimental data showed that the meshed network improved the range and throughput of the communication link compared to a static meshed network.

This chapter proposes a solution by using a decentralized nonlinear MPC to optimize the state of multiple UAS to achieve the desired signal strength between the ground station and the ASV. This is done while minimizing the power consumption of the maneuvering aircraft in order to maximize the mission endurance. As the directivity of the antennas is also considered, roll angles of the UAS are taken into account when determining the signal strength. The proposed solution also takes into account the effects of wind on the aircraft performance.

3.2 Methodology

3.2.1 Path Planning

Assuming that the path planning problem starts with all the UAS organized on a network topology (Figure 3.1) providing the required transmitter-receiver signal strength between the ASV and the ground station, the aim of the algorithm is to optimize the states of the UAS to sustain a signal strength which does not fall below a preset requirement. A Nonlinear Model Predictive Control (NMPC) [23] method is used to optimize the airspeed and bank angle of the UAS to achieve the desired signal strength while minimizing power consumption.

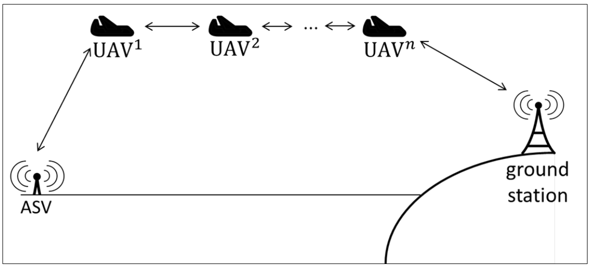


Figure 3.1: Network topology

A centralized control system would need to optimize the control inputs of all the relay nodes, causing significantly increased complexity. With an increased number of relay nodes the method ultimately becomes unfeasible. Therefore, a fully decentralized NMPC is proposed in this chapter, where each UAS plans its own path and attitude, taking into consideration the planned states of the other UAS and the planned position of the ASV in time. This can only be achieved under the assumption that the information can be shared between the UAS. Here, the UAS only needs to be able to communicate with the adjacent nodes.

Assuming that the UAS will fly at a constant altitude maintained by the autopilot, a two-dimensional kinematic model can be used based on the Coordinated Flight Vehicle model [123] as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{v_a} \\ \dot{\phi} \end{pmatrix} = f(\mathbf{x},\mathbf{u}) = \begin{pmatrix} v_a \cos\psi + v_w \cos\psi_w \\ v_a \sin\psi + v_w \sin\psi_w \\ \frac{g \tan\phi}{v_a} \\ u_v \\ u_\phi \end{pmatrix} \tag{3.1}$$

where $\mathbf{x} = (x, y, \psi, v_a, \phi)$ are the North and East positions in the NED frame, heading, air-relative velocity (airspeed) and bank angle of the UAS, respectively. v_w and ψ_w are the velocity and heading of the wind and g is the gravity acceleration of $9.81\ \mathrm{m/s^2}$. $\mathbf{u} = (u_v, u_\phi)$ are the acceleration control input and the roll rate control input, respectively.

The model is discretized by the forward Euler method:

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k + T_s f(\mathbf{x}_k, \mathbf{u}_k), \tag{3.2}$$

where T_s is the sampling period.

The overall control problem is decomposed as a unique local control problem for each UAS node, where each UAS optimizes its own state based on the signal strength with respect to its two adjacent nodes, while taking into consideration the planned states of the adjacent nodes. Collision avoidance and power consumption are also considered.

Considering n UAS ($\mathbf{x}^i, \forall i \in \{1,...,n\}$), a fixed ground station ($\mathbf{x}^0$) and a moving ASV ($\mathbf{x}^{n+1}$), the NMPC algorithm finds a control input sequence $U_k^i = \{\mathbf{u}_0^i, \mathbf{u}_1^i, ..., \mathbf{u}_{N-1}^i\} \in R^{2\times N}$ for the ith UAS, which solves the following optimal control problem:

$$\begin{aligned}
\text{minimize}\quad & \delta^i(\bar{\mathbf{x}}_N^i) + \sum_{k=0}^{N-1} L^i(\bar{\mathbf{x}}_k^i, \mathbf{u}_k^i) && (3.3)\\
\text{subject to}\quad & \mathbf{x}_{k+1}^i = f_d(\mathbf{x}_k^i, \mathbf{u}_k^i) && (3.4)\\
& v_{a\,min} \le v_{a\,k}^i \le v_{a_{max}} && (3.5)\\
& \phi_{min} \le \phi_k^i \le \phi_{max} && (3.6)\\
& \dot{v}_{a_{min}} \le u_{v\,k}^i \le \dot{v}_{a_{max}} && (3.7)\\
& \dot{\phi}_{min} \le u_{\phi_k}^i \le \dot{\phi}_{max} && (3.8)\\
& \|\mathbf{C}_1(\mathbf{x}_k^i - \mathbf{x}_k^j)\| > r_c,\ \forall j \in \{1,...,n\}\backslash\{i\} && (3.9)
\end{aligned}$$

where

$$\delta^i(\bar{\mathbf{x}}^i_N) = aJ^i(\bar{\mathbf{x}}^i_N), \tag{3.10}$$

$$L^i(\bar{\mathbf{x}}^i_k, \mathbf{u}^i_k) = aJ^i(\bar{\mathbf{x}}^i_k) + b{u^i_{vk}}^2 + c{u^i_{\phi k}}^2, \tag{3.11}$$

and

$$J^i(\bar{\mathbf{x}}^i_k) = \alpha E^i(\mathbf{C}_2\bar{\mathbf{x}}^i_k) + (1-\alpha)P^i(\mathbf{C}_3\mathbf{x}^i_k). \tag{3.12}$$

$\bar{\mathbf{x}}^i_k = [\mathbf{x}^{i-1}_k, \mathbf{x}^i_k, \mathbf{x}^{i+1}_k]$ are the states of the adjacent nodes, N is the number of horizon steps and r_c is the minimum safe distance between the UAS to avoid collision. a, b, c are constant weighting factors and $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3 \in R^{2\times 5}$ are used to define which state variables of the vehicles are to be considered in the equations. In case of $\mathbf{C}_1$, only the x and y positions should be used, in case of $\mathbf{C}_2$, x, y, ψ and ϕ are used and in case of $\mathbf{C}_3$, only v_a and ϕ are used.

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{3.13}$$

$$\mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.14}$$

$$\mathbf{C}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.15}$$

As the main goal of the system is to provide the desired signal strength with the lowest power consumption, $E^i(\mathbf{C}_2\bar{\mathbf{x}}^i)$ through Eq. (3.16) is the difference between the desired and actual signal strength between the ith UAS and its adjacent nodes, while $P^i(\mathbf{C}_3\mathbf{x}^i)$ through Eq. (3.27) is the power consumption function. Finally, α is a weighting factor that defines how much power can be used to improve the communication.

3.2.2 Signal strength

The function of the transmitter-receiver signal strength between the ith UAS and its adjacent nodes is the sum of the difference between the calculated signal strength and the desired signal strength, for the ith UAS and each of the two adjacent nodes (Eq. 3.16). If the signal strength is equal or higher than the desired one, the error is considered to be zero (Figure 3.2).

$$E^i(\mathbf{C}_2\bar{\mathbf{x}}^i) = \Delta Err^i(\mathbf{C}_2\bar{\mathbf{x}}^i)\frac{\frac{\pi}{2} - \arctan(-\beta\Delta Err^i(\mathbf{C}_2\bar{\mathbf{x}}^i))}{\pi}, \tag{3.16}$$

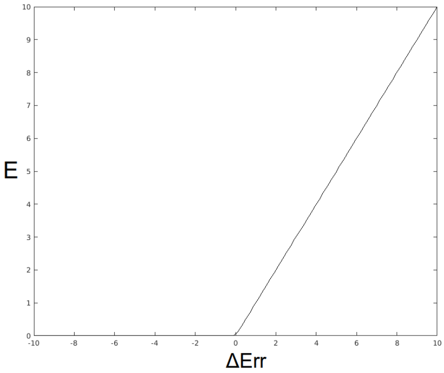


Figure 3.2: $E(\mathbf{C}_2\bar{\mathbf{x}})$ for a $\Delta Err^i(\bar{\mathbf{x}})$ from -10 to 10 and β of 100000.

where β is the constant which defines how close the curve will be to a conditional function and $\Delta Err^i(\bar{\mathbf{x}}^i)$ is the difference between the desired signal strength and the minimum calculated signal strength as the following:

$$\Delta Err^i(\mathbf{C}_2\bar{\mathbf{x}}^i) = P_d - P^i_{min}(\mathbf{C}_2\bar{\mathbf{x}}^i), \tag{3.17}$$

where P_d is the desired signal strength and $P^i_{min}(\mathbf{C}_2\bar{\mathbf{x}}^i)$ is the lowest signal strength between the ith UAS and each one of its adjacent nodes, as the lowest signal strength is the one limiting the link. To calculate the minimum value between two elements, the following equation is used:

$$\begin{aligned} &P^i_{min}(\mathbf{C}_2\bar{\mathbf{x}}^i) = \\ &0.5(P^{i-1}_{dBm}(\mathbf{C}_2\mathbf{x}^{i-1}, \mathbf{C}_2\mathbf{x}^i) + P^i_{dBm}(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_2\mathbf{x}^{i+1}) - \\ &\sqrt{(P^{i-1}_{dBm}(\mathbf{C}_2\mathbf{x}^{i-1}, \mathbf{C}_2\mathbf{x}^i) - P^i_{dBm}(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_2\mathbf{x}^{i+1}))^2)} \end{aligned} \tag{3.18}$$

where $P^i_{dBm}(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_2\mathbf{x}^{i+1}) = 10 \log P^i(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_2\mathbf{x}^{i+1}) + 30$.

To calculate the signal strength between nodes, the Friis equation [88] is used to calculate the received power, based on the distance between the nodes and the directivity of the antennas:

$$\begin{aligned} P^i(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_2\mathbf{x}^{i+1}) = P_t \cdot &D^{i,i+1}(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}) \cdot \\ &D^{i+1,i}(\mathbf{C}_2\mathbf{x}^{i+1}, \mathbf{C}_1\mathbf{x}^i) \cdot \\ &FSPL(\mathbf{C}_1\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}), \end{aligned} \tag{3.19}$$

where $FSPL(\mathbf{C}_1\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1})$ is the Free-Space-Path-Loss:

$$FSPL(\mathbf{C}_1\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}) = (\frac{\lambda}{4\pi d^i(\mathbf{C}_1\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1})})^2, \tag{3.20}$$

and P_t is the transmitted power, λ is the wavelength, $D^{i,i+1}(\mathbf{x}^i, \mathbf{x}^{i+1})$ is the directivity gain with respect to the position and antenna angle of the ith and $(i+1)$th nodes, and $d^i(\mathbf{C}_1\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1})$ is the distance between nodes:

$$d^i(\mathbf{C}_1\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}) = \sqrt{(x^i - x^{i+1})^2 + (y^i - y^{i+1})^2 + (z^i - z^{i+1})^2}, \tag{3.21}$$

where z^i is the constant z offsets of the antennas of the ith node.

The directivities are obtained by the following equation:

$$D^{i,i+1}(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}) = 4\pi \frac{F(\theta(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}))}{F_{avg}}, \tag{3.22}$$

where $F(\theta(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}))$ is the power radiation pattern of the antenna and F_{avg} is the average power density over a sphere, given by:

$$F_{avg} = \int_0^{2\pi} \int_0^{\pi} F(\theta(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1})) sin\theta d\theta d\gamma, \tag{3.23}$$

where $\theta(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1})$ is the angle between the antenna of the ith node and the body of the $(i+1)$th node:

$$\theta(\mathbf{C}_2\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1}) = \arcsin \frac{\mathbf{R}_z(\psi^i)\mathbf{R}_x(\phi^i)\mathbf{v}(\mathbf{C}_1\mathbf{x}^{i+1} - \mathbf{C}_1\mathbf{x}^i)}{d^i(\mathbf{C}_1\mathbf{x}^i, \mathbf{C}_1\mathbf{x}^{i+1})}, \tag{3.24}$$

where $\mathbf{v} = [0\ 0\ 1]$ is the reference vector from where the angle θ will be calculated relative to, $\mathbf{R}_x(\phi)$ and $\mathbf{R}_z(\psi)$ are the rotation matrices in x and y with respect to the angles ϕ and ψ [13].

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}. \tag{3.25}$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.26}$$

3.2.3 Aircraft power consumption

To obtain an accurate model of the overall system performance, the in-flight performance of the aircraft needs to be considered [54][53]. In this study, the path is being optimized with respect to both the aircraft's airspeed and bank angle. Therefore it is necessary for the aircraft performance model to evaluate the in-flight power consumption, and express the penalty for changing airspeed and performing longitudinal maneuvering. For a propeller-driven aircraft in steady flight the consumed power (P) is found through:

$$P = \frac{D\,v_a}{\eta_p}, \tag{3.27}$$

where D is the aerodynamic drag force experienced by the aircraft, v_a is the airspeed, and η_p is the total propulsion efficiency. For level flight the generated lift L equals the aircraft weight W. However, when the bank angle ϕ is no longer zero, the lift force is rotated by an angle ϕ in relation to the gravity component. This results in:

$$L \cos\phi = W. \tag{3.28}$$

In this study the loss of lift caused by an increased bank angle is mitigated by increasing the airspeed. This is done so that a constant altitude can be maintained. It is therefore necessary to account for the effects of bank angle ϕ on the required airspeed v_a. When considering a coordinated flight the adjusted airspeed for a turn with constant-C_L is found through:

$$v_a = \sqrt{\frac{2W}{\rho_\infty S}\frac{n}{C_L}} = v_\infty \sqrt{n}. \tag{3.29}$$

Here v_∞ is the required airspeed for level unaccelerated flight in [m/s]. S is the aircraft's effective wing surface in [m^2], W is the aircraft's total weight in [N], and ρ_∞ is the air density in [kg/m^3]. The initial aerodynamic lift coefficient C_L is determined through:

$$C_L = \frac{2W}{\rho_\infty S v_\infty^2}. \tag{3.30}$$

Furthermore, n is the load-factor, which in accelerated flight is larger than zero. This is defined as:

$$n = \frac{1}{\cos\phi} = \left(\frac{T}{W}\right)\left(\frac{L}{D}\right). \tag{3.31}$$

In level unaccelerated flight and constant speed level turns the condition applies that the generated thrust force T equals the drag force experienced by the aircraft. Assuming a general simplified drag model in a coordinated flight where trim drag is neglected, and the thrust line is parallel to the airspeed, the drag force D is obtained through [5]:

$$D = T = \frac{1}{2}\rho_\infty v_a^2 S \left[C_{D_0} + k\left(\frac{nW}{qS}\right)^2\right]. \tag{3.32}$$

Here C_{D_0} is the aircraft's zero-lift drag coefficient. Finally, k is the lift-induced drag constant, which is defined as:

$$k = \frac{1}{\pi\ AR\ e} \tag{3.33}$$

Here AR is the aircraft's effective aspect ratio, and e is the Oswald efficiency factor.

Figure 3.3: Skywalker X8 sUAS - operated by AMOS UAVLab (Source: NTNU)

In this study the Skywalker X8 aircraft is used in the simulations. The X8 (Figure 3.3) is a small battery-powered unmanned aircraft in flying-wing configuration. It has a wingspan of 2.1 meters with a mission-ready weight of 3.36 kilograms. The effective wing surface is approximately 0.74 square meters. Two elevon control surfaces are located on the outer wings to provide longitudinal, lateral and directional control of the aircraft. The aerodynamic lift model that was used in the simulation was based on wind tunnel experiments, which are described in [52], while the value for k is assumed to be 0.0907. The propulsion characteristics are unknown. Therefore for the remainder of this study the propulsion efficiency η_p is assumed to be invariable at 0.5, and the maximum thrust T_{max} is assumed to be constant at 25 N with ideal-battery discharge characteristics.

From Eq. (3.31) and (3.32) it can be observed that as the bank angle increases, the power consumption increases. The in-flight performance, and consequently the power consumption, are affected through the required increase in airspeed. It is important to note that to avoid the aircraft from entering a stall, the maximum load factor n_{max} is limited by the available thrust. The thrust-limited maximum load factor can be found by substituting the maximum available thrust in Eq. (3.31). Solving for ϕ yields the thrust-limited maximum bank angle. The minimum value for the airspeed v_{stall} as a function of load factor n and $C_{L_{max}}$ can be found by inserting the maximum lift coefficient into Eq. 3.29. This results in:

$$v_{stall} = \sqrt{\frac{2}{\rho_\infty}\frac{W}{S}\frac{n}{C_{L_{max}}}}. \tag{3.34}$$

Finally, a *structural* limitation exists. As the structural limit load factor for the Skywalker X8 platform is not known it is chosen to implement a stricter limitation for the simulations as further referenced in section 3.4.

3.3 System description

The simplified block diagram of the control system is shown in Figure 3.4. The Model Predictive Control (MPC) runs inside DUNE [112], which is an open source

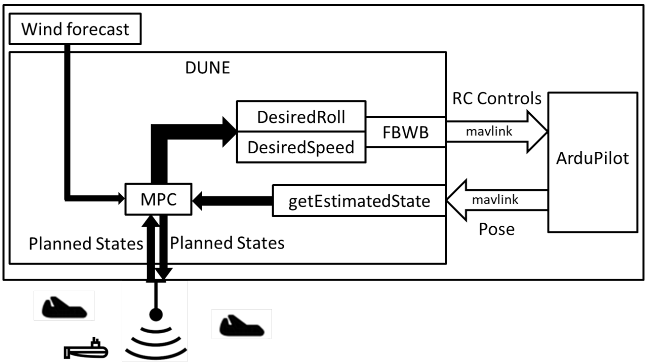


Figure 3.4: Individual system's block diagram.

robot framework developed by the Underwater Systems and Technology Laboratory (LSTS) of the University of Porto. DUNE is installed on the onboard computer and communicates with the ArduPilot board via Micro Air Vehicle Communication Protocol (MAVLink). When executing the maneuver, the ArduPilot opperates in Fly-By-Wire-B (FBWB) mode and gets the desired roll and airspeed as Radio Control (RC) inputs from DUNE.

The MPC is included in a DUNE task. Every time the UAS gets the updated planned states of the adjacent nodes, it calls the MPC function and, based on its current state and on the adjacent nodes states, it calculates its own planned states. In this step, the wind forecast present in a data file is also considered. After the optimization, the UAS broadcasts its planned states to the other nodes and sends the commands to the ArduPilot.

The coordination of the mission is done by the dispatch and consumption of Inter-Module Communication Protocol (IMC) messages. In DUNE, if a IMC message is dispatched by a node, all nodes that are monitoring that type of message will receive it and run the routine binded to the reception of that message.

3.3.1 Mission coordination

Each UAS awaits the start of the mission in loiter mode. When the ASV starts the plan, it dispatches a "PlanControl" message that is consumed by the UAS. All UAS then start the maneuver and dispatch their planned states considering the loiter maneuver, except the UAS closest to the ASV and the one closest to the Ground Station. These two UAS wait for the planned states of their adjacent nodes and are the first ones to run the MPC. Figure 3.5 shows the flow chart of the IMC messages between the nodes.

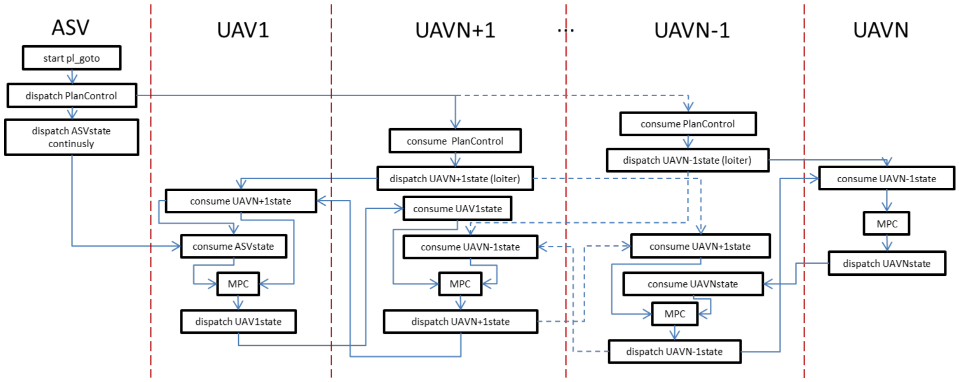


Figure 3.5: Communication between nodes using IMC messages and DUNE Tasks

3.4 Simulations

The first simulation (Figure 3.6) was done for a scenario considering two UAS to close the Wi-Fi link between the ASV and a ground station. The simulation starts when the ASV is at position [300,300] - 423 meters away from the ground station. The first UAS starts the mission at position [100,100] and the second one at position [200,200]. The ASV is moving north with constant speed over ground of 1.6 m/s.

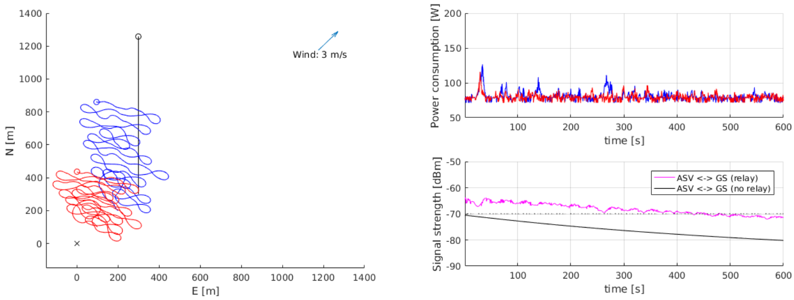


Figure 3.6: First simulation scenario. Ground station as a black x, ASV's path in black, UAS 1's path in blue and UAS 2 path's in red.

It is possible to notice that when the mission starts, the straight link between the ASV and the ground station has less than -70 dBm, which was chosen as the desired signal strength as it is the minimum power to establish a Wi-Fi connection. Using the two UAS, the ASV can progress with the mission for more 8 minutes (or 768 meters north), when two UAS are not capable to provide the desired signal strength anymore. This means an increase from around 400 to 800 meters of operational radius around the ground station, resulting in an operational area 4 times bigger.

This increase can be expanded if more UAS are used as relay nodes.

Considering the different limitations of the load factor, the maximum bank angle ϕ_{max} is chosen to be restricted to 20^o, which offers an associated stall speed of 8.9 meters per second. Therefore the minimum airspeed $v_{a_{min}}$ is defined to be 12.0 meters per second. The other parameters used in the simulation are shown in tables 3.1, 3.2 and 3.3.

To solve the NMPC problem, Particle Swarm Optimization (PSO) [35] was used. The algorithm, which is described in the Appendix B, was developed using CUDA C programming language, benefiting from the parallelization to reduce the computational time. Each optimization takes around 250 ms, proving the algorithm suitable for real-time applications. The computational time can be reduced if necessary, by adjusting the horizon and number of steps. This would result in a decrease of the optimality, but may be beneficial in real environments, as the system will calculate the control inputs for a state closer to the state that it was when the calculation began.

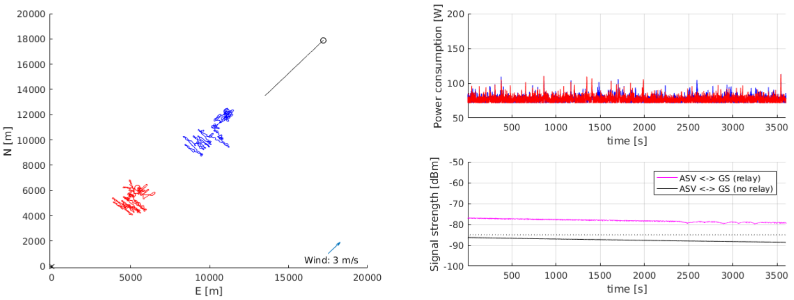


Figure 3.7: Second simulation scenario. Ground station as a black x, ASV's path in black, UAS 1's path in blue and UAS 2 path's in red.

A second simulation was done for a scenario where the nodes were equipped with a communication system inspired by the characteristics of the Maritime Broadband Radio (MBR) 144 radio from the company Kongsberg Maritime [86], together with a 7 dBi antenna with around 25 degrees of $HPBW$ in the elevation plane. This radio can provide up to 20 km of range [164] for a frequency between 4.90 and 5.85 GHz. The communication parameters of this simulation were chosen as it is shown in table 3.4 and the results are shown in Figure 6. The ASV starts the simulation at position [13500,13500] (around 19 km from the ground station) and the UAS 1 and UAS 2 at [4500,4500] and [8000,8000], respectively. Here, the duration of the mission was set to be one hour. It was found that besides the significant increase in obtainable communication range, it is also noticeable that the link between the ASV and the ground station when using the two UAS as relay nodes has a higher signal strength throughout the duration of the mission.

Table 3.1: MPC Parameters

Param	Name	Value	Unit
v_w	velocity of wind	3.0	m/s
ψ_w	heading of wind	0.7	rad
g	gravity acceleration	9.81	m/s^2
t	time horizon	15.0	s
N	horizon steps	30	m/s
h	altitude of UAS	100.0	m
hN	altitude of the ground station	20	m
$v_{a_{min}}$	minimum airspeed	12.0	m/s
$v_{a_{max}}$	maximum airspeed	20.0	m/s
ϕ_{min}	minimum bank angle	-0.349	rad
ϕ_{max}	maximum bank angle	0.349	rad
$\dot{v}_{a_{min}}$	minimum acceleration	-0.2	m/s^2
$\dot{v}_{a_{max}}$	maximum acceleration	0.2	m/s^2
$\dot{\phi}_{min}$	minimum bank angle rate	-1.4	rad/s
$\dot{\phi}_{max}$	maximum bank angle rate	1.4	rad/s
r_c	safe distance between UAVs	50.0	m
a	weight of cost function	1.0	
b	weight of acceleration control	1.0	
c	weight of bank angle rate control	1.0	
α	weight of signal strength / power saving	0.99	

Table 3.2: Communication Parameters of Scenario 1

Param	Name	Value	Unit
β	conditional function fitting constant	99999	
P_d	desired signal strength	-70	dBm
P_t	transmitter power	100.0	mW
λ	wavelength	12.5	cm
$HPBW$	Half-Power-Bandwidth	2.09	rad

3.5 Discussion

In the current implementation of the algorithm, the wind is assumed to be constant. For small aircraft, however, the wind can have a major effect on the power consumption. It would therefore be beneficial to use a forecast wind map or local wind estimations in the optimization algorithm. A proposed method for the forecast map is to fit an analytic function to a discrete wind map in order to improve the convergence of the optimization algorithm.

Field experiments are also necessary in order to define constraints for the signal strength variations, that could affect the overall network capability.

Regarding the number of UAS and the ASV behavior, it is necessary to simulate

Table 3.3: Power consumption Parameters

Param	**Name**	**Value**	**Unit**
η_p	propulsion efficiency	0.5	
W	aircraft weight	32.96	N
ρ_∞	air density	1.225	kg/m^3
C_{D_0}	zero-lift drag coefficient	0.125	
k	lift-induced drag constant	0.0224	
S	wing surface	0.74	m

Table 3.4: Communication Parameters of Scenario 2

Param	**Name**	**Value**	**Unit**
β	conditional function fitting constant	99999	
P_d	desired signal strength	-85	dBm
P_t	transmitter power	2.0	W
λ	wavelength	6.0	cm
$HPBW$	Half-Power-Bandwidth	0.437	rad

scenarios where more UAS are used and that the ASV moves in different patterns to evaluate the system performance. Simulations using different altitudes for the UAS should also be considered.

3.6 Conclusion

In this chapter, a communication relay solution was presented. The goal of the system is to provide a minimum signal strength between the Autonomous Surface Vehicle and a ground station by using Unmanned Aerial Systems as communication relay nodes. The system was build to be used with DUNE robotic framework and was modeled as a Nonlinear Model Predictive Control problem. Simulations show that the system is capable to be tested in field experiments and may be a suitable tool in maritime missions.

Chapter 4

Autonomous Unmanned Aerial Vehicles in Search and Rescue missions using real-time cooperative Model Predictive Control

Unmanned Aerial Vehicles (UAVs) have recently been used on a wide variety of applications due to their versatility, reduced cost, rapid deployment, among other advantages. Search and Rescue (SAR) is one of the most prominent areas for the employment of UAVs in place of a manned mission, especially because of its limitations on the costs, human resources, and mental and perception of the human operators. In this work, a real-time path planning solution using multiple cooperative UAVs for SAR missions is proposed. The technique of Particle Swarm Optimization is used to solve a Model Predictive Control (MPC) problem that aims to perform search in a given area of interest, following the directive of international standards of SAR. A coordinated turn kinematic model for level flight in the presence of wind is included in the MPC. The solution is fully implemented to be embedded in the UAVs on-board computer with DUNE, an on-board navigation software. The performance is evaluated using Ardupilot's Software-In-The-Loop with JSBSim flight dynamics model simulations. Results show that when employing three UAVs, the group reaches 50% of Probability of Success 2.35 times faster than when a single UAV is employed.

4.1 Introduction

Search and Rescue (SAR) is one of the fields where the employment of UAVs brings many advantages over manned missions, such as its reduced costs, lower use of human resources, and mental and perception limitations of human operators. [127] was one of the first works to perform experimental tests of a complete autonomous single UAV SAR solution. A probability density function (PDF) that expressed the likelihood of the target's location was one of the main inputs of the system. Video data from the UAV was transmitted to the ground station, that processed it in real time using computer vision techniques to detect the presence of the target and update the PDF. Paths were generated by the ground station to maximize the probability of finding the targeted object. The experimental flights showed satisfactory results in searching and detecting the target. The main necessary improvements identified by the authors were to implement on-board computing and to use multiple UAVs in the future.

SAR missions with autonomous UAVs are usually defined as an exploration problem. Exploration approaches can be used in a wide range of applications. For example, ice management, such as proposed by [58], where a Centralized Model Predictive Search Software was used for surveillance and tracking of ice using multiple UAVs. In the solution proposed by this reference, the algorithm finds a set of optimal waypoints that are sent to the autopilot control unit. The solution was tested in a Software-In-The-Loop environment and the results were evaluated for a different number of UAVs. [3] compared five exploration algorithms for SAR missions using a team of multiple UAVs. A centralized mission planner was used for the path planning, whose objective was to find a sequence of waypoints for each UAV to follow. In [124], an off-line algorithm to find a sequence of waypoints to be followed by a team of UAVs in a SAR mission is proposed. The cost function is built to minimize the travelled distance and the risk that people are exposed to. The authors created a non-uniform "risk map" and "probability of people map" for the simulated scenario.

A broad literature review about the persistent surveillance problem was done by [105] focusing on the use of multiple UAVs. Persistent surveillance is a type of exploration problem where the areas must be revisited over time. Among the many topics that the literature review covers, grid decomposition and path planning techniques are the ones of the most interest for this work. The author reviews the most common types of grid decomposition classifying the rectangular one, that is also used by the enhanced solution proposed by this work, as the most popular. Regarding the path planning techniques, the author states that the most common methods are classical search methods such as A* [75], decision theoretic methods such as Mixed Integer Linear Programming (MILP) [40], and Spanning Tree Coverage (STC) methods [45]. Model Predictive Control (MPC) [24] is mentioned as a topic less studied compared to the other planning techniques, but with significant advantages because it directly incorporate dynamic constraints, it is less heuristic and can react to changes in the environment.

MPC is a receding horizon control technique where the motion constraints are integrated in the control problem, which is particularly interesting for problems with fixed-wing UAVs. Also, as the optimization is done for a finite time horizon, the technique is proper for real-time problems where the environment can dynamically change during the mission execution. In [131], MPC was used for sea Search and Track missions using an autonomous UAV. Hardware-In-The-Loop tests were performed. Waypoints were optimized and sent to the autopilot. Gimbal attitude was also optimized and sent to the servo system. The MPC optimization was not run on-board but on a dedicated computer in the ground control station. In [158], a MPC solution was proposed for target tracking using a solar-powered UAV. Energy harvesting and consumption were also included in the cost function. A three-dimensional dynamic model was used with the thrust, roll angle and lift coefficient as control inputs. The model also included horizontal winds. [157] describes a tracking application using multiple solar-powered UAVs in urban areas. A two-dimensional kinematic model is used without considering the wind. The cost function includes the energy harvesting and considers the shadows made by the buildings. Energy consumption is also considered. In addition, a distance constraint was included to guarantee the communication quality. The cooperative solution used is similar to the one described in this study, where the UAVs share their planned states. The meta-heuristic technique Grasshopper Optimization Algorithm is used for the optimization. A performance comparison between optimization techniques is presented, including the performance of Particle Swarm Optimization algorithm, which is used in this study. In this reference, the cost function is composed by the feasibility cost, the mission cost, the energy cost, the safety cost, the collision cost and the communication cost. In [9], a cooperative multiple UAVs solution using MPC was used to close the communication link between a moving Autonomous Surface Vehicle and the ground station. Each UAV had to minimize a local cost function that took into consideration the planned states of the adjacent UAVs. In [31], a multiple UAVs receding horizon strategy was proposed for a cooperative surveillance problem. A potential field method was used for collision avoidance and network topology control management. The cooperative searching model was established based on the detection probability of the UAVs on targets in cells. In addition, a forgetting factor was included to indicate how fast the detection efforts are forgotten, so the UAVs can revisit the areas that were searched before. Simulations for different parameters were compared. Also, the performance of the proposed method was compared to the performance of a parallel sequence search. In [146], a multi-vehicle cooperative search solution was proposed using MPC. Decoupled, centralized, cooperative and greedy approaches were compared.

In this study, a multiple UAVs cooperative Nonlinear Model Predictive Control solution to search a given area is proposed. The technique of Particle Swarm Optimization is used to find the control inputs that solve the optimal control problem.

A coordinated turn kinematic model is implemented considering the effects of wind. The search area is divided into cells and each cell has an associated reward, that in this work is defined according to the international Search and Rescue directives. A Software-In-The-Loop (SITL) environment with flight dynamics simulations is used to test the solution. SITL simulations are more realistic numerical simulations that run in real-time and closed loop with the control system software, in which aircraft sensor data is simulated by a flight dynamics model.

In addition, it is assumed that there is always connection between UAVs in the search area so that they can always share information between them. It is beyond the scope of this work to discuss network design. Research about communication and networks for UAVs in maritime missions is available at [163] and [165], where communication technologies, standards and protocols are discussed and system architectures are proposed, including hardware description.

The main contributions of this research to the field are:

1. Research about the use of receding horizon techniques in exploration problems is limited. This research contributes to the field by bringing a novel multiple UAVs solution to an exploration problem using Model Predictive Control where a finite time horizon grid search cost function with cells rewards and terminal cost is proposed.

2. The solutions found in the literature are developed with simplified vehicle models, in which the effects of wind are not considered when UAV platforms are employed. In small UAV missions, wind can easily exceed half of the UAV's airspeed, significantly affecting the UAV performance. In this work, a coordinated turn kinematic model that takes the wind into consideration is designed. The model is developed so that the control inputs are the same as the autopilot control unit controls.

3. Usually, the solutions in the literature are only simulated in environments without embedded programming restrictions and where vehicle dynamics are not simulated. This makes results not close enough to what is expected in real-life applications. Implementing the solution in software for real-time applications brings additional challenges such as communication delays, processing time, actuator limitations, among others. In this research, the algorithm is fully implemented in an embedded software and tested in a real-time SITL environment that also simulates the flight dynamics, bringing results that are very close to reality.

4. In the literature, the SAR scenarios created for testing the proposed solutions are not based on standard directives. Therefore, making a fair comparison between different solutions is difficult. In this work, international SAR directives

are considered to define the mission scenario and the performance indicators. This also allows testing the proposed solution in a relevant case.

4.2 Materials and Methods

4.2.1 Embedded System

The path planning algorithm was implemented as a task in DUNE: Unified Navigation Environment [112]. It allows the operation of a wide variety of robots using the same environment. This facilitates the development because the communication between DUNE and the different control units is transparent to the user.

The embedded system is outlined in Figure 4.1. The communication between DUNE and the Ardupilot [141] autopilot control unit is done via MAVLink Micro Air Vehicle Protocol [142]. To command the Ardupilot, DUNE's MPC task must dispatch a message with the desired command, which will be interpreted by DUNE's Ardupilot control task and then sent to the Ardupilot via MAVLink.

The communication between DUNE tasks is done via the IMC: Intermodule Communication API protocol [112]. This protocol basically operates by dispatching and consuming messages. So, if a message is dispatched by a task, another task that is waiting for that message will consume it.

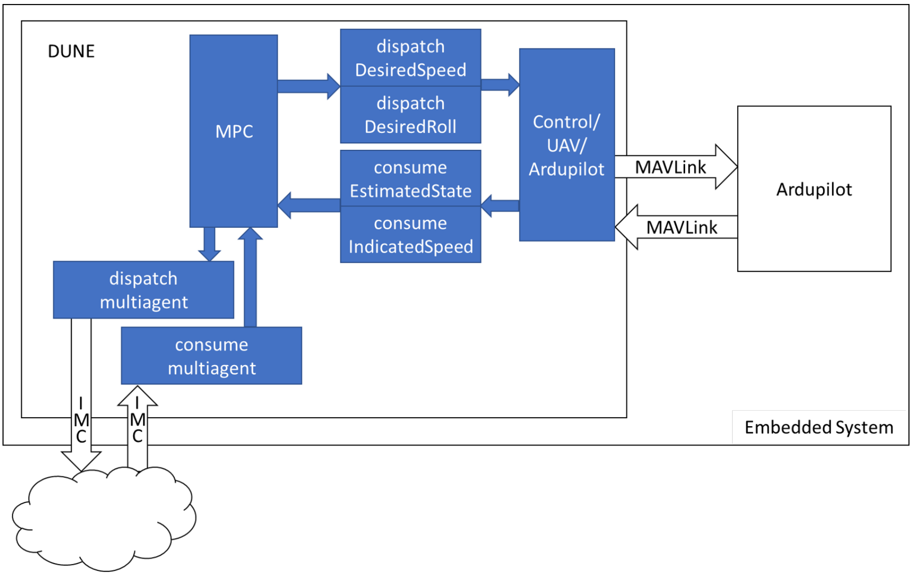


Figure 4.1: Simplified DUNE block diagram.

The communication with other UAVs in DUNE's cloud and with the ground station is done via IMC protocol also. The vehicles must be in the same network. Therefore, IP addresses and TCP and UDP ports must be properly configured and included in each vehicle's DUNE configuration files. To allow the communication between vehicles, the InterVehicleCommon UDP transport module is used.

Inside DUNE resides the MPC task. The commands to control the UAV are given by the DesiredSpeed and DesiredRoll IMC messages, which carry, respectively, the airspeed and roll control inputs given by the optimization. These messages are interpreted by DUNE's Ardupilot control task, that sends the correspondent MAVLink message to the Ardupilot unit. In addition, DUNE's Ardupilot control task is responsible for changing the Ardupilot flight mode to the Fly-By-Wire-B (FBWB) mode when the mission is started. In this flight mode, the Ardupilot control unit is responsible for holding the aircraft's altitude. Airspeed, roll angle and altitude external controls are accepted by the Ardupilot in the FBWB mode. The Ardupilot control unit is responsible for the low level control loops to maintain the commanded altitude, airspeed and roll angle. In this work, only roll and airspeed are controlled by the optimization algorithm, therefore, the altitude control is fixed at the desired altitude during the whole mission.

DUNE's Ardupilot control task is also responsible for receiving the UAV's pose and attitude information and to dispatch it in the IMC messages EstimatedState and IndicatedSpeed. These messages are consumed by the MPC task to be used as the current state of the UAV.

The communication between the UAVs is done by the multiagent message, which was created and included in the IMC messages list specifically for this application. This message carries the information that the UAVs need to share between them, such as planned control inputs and current state.

Each UAV waits for the multiagent messages from all other UAVs before running the MPC optimization, e.g. if a team of three UAVs is employed, UAV 0 only runs the optimization algorithm after receiving the multiagent messages with the needed information from UAVs 1 and 2. This flow is described in Figure 4.2. Once the messages are received and the optimization loop is over, the UAV dispatches its multiagent message containing all information that need to be shared with the other UAVs.

In order to prevent control issues caused by communication and processing delays, the UAVs states used by the MPC optimization are predicted according to the previously commanded control inputs and the measured delays by using Eq. 4.7. Therefore, in the end of the optimization loop, the real states of the UAVs are expected to be close to the ones considered by the optimization algorithm when finding the optimal control inputs. This results in a superior behavior since the control inputs are obtained for UAVs attitudes and positions that are closer to reality.

Note that, in an extension of the method proposed by this work, measures can

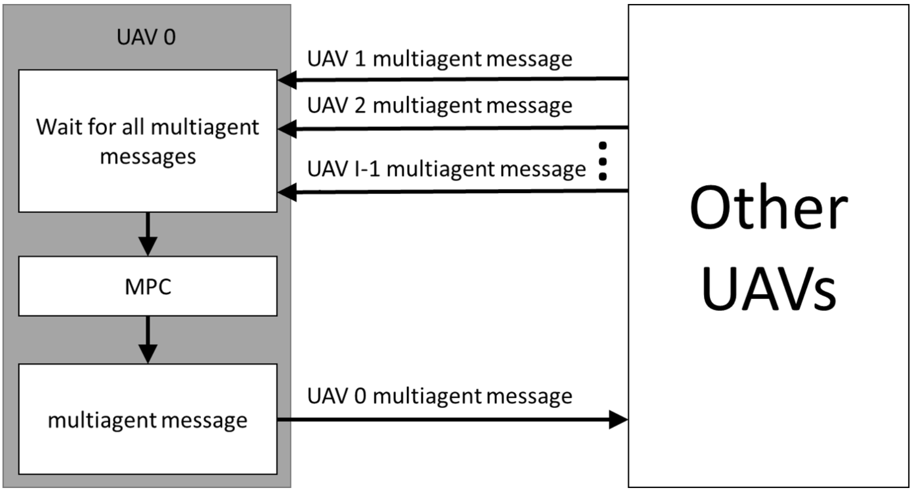


Figure 4.2: UAV agents flowchart.

be employed to protect the system from communication failures, so that the UAVs do not wait for delayed messages for too long but uses predictions instead. Also, distance constraints can be included in the MPC in order to guarantee that the UAVs do not separate further than the the maximum distance that enables the communication.

4.2.2 Optimal Control Problem

Coordinated Turn Model

A two-dimensional kinematic model was developed based on the Coordinated Turn model for level flight [13]. In this model, the UAV turns by changing its roll angle so that there is no net side force acting on the UAV. Therefore, it is possible to relate the course rate and the roll angle by making the centrifugal force acting on the UAV equal and opposite to the horizontal component of the lift force acting in the radial direction (Figure 4.3):

$$F_{lift} \sin u_{\phi} \cos(\chi - \psi) = m v_g \dot{\chi}, \tag{4.1}$$

where F_{lift} is the lift force in [N], u_{ϕ} is the roll control input in [rad], χ is the course angle in [rad], ψ is the aircraft heading in [rad], m is the aircraft mass in [kg], v_g is the ground speed in [m/s] and $\dot{\chi}$ is the course rate in [rad/s].

Also, the vertical component of the lift force should be equal and opposite to the projection of the gravitational force (Figure 4.3):

$$F_{lift} \cos u_{\phi} = mg, \tag{4.2}$$

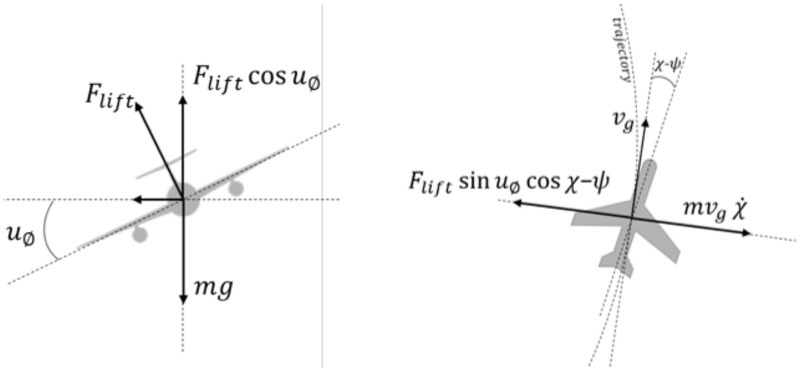


Figure 4.3: Vertical (left) and horizontal (right) forces in the coordinated turn maneuver.

where g is the gravitational acceleration of 9.81 $\mathrm{m/s^2}$.

By combining Eq. 4.1 and 4.2 it is possible to find the relation between the course rate and the roll control input:

$$\dot{\chi} = \frac{g}{v_g} \tan u_\phi cos(\chi - \psi). \tag{4.3}$$

As wind is a major issue on UAV missions as it can likely reach more than half of the UAV's maximum airspeed, the Coordinated Turn model used in this project was developed to consider the influence of wind on the UAV kinematics.

Therefore, the coordinated turn kinematic model for level flight in the presence of wind is given by:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\chi} \end{pmatrix} = f(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} v_g \cos\chi \\ v_g \sin\chi \\ \frac{g}{v_g} \tan u_\phi cos(\chi - \psi) \end{pmatrix}, \tag{4.4}$$

where $\mathbf{x} = (x, y, \chi)$ are the north and east positions in the NED frame in [m] and the course angle in [rad], respectively. $\mathbf{u} = (u_v, u_\phi)$ are the airspeed control input in [m/s] and roll control input in [rad], respectively, and with the ground speed (v_g in [m/s]):

$$v_g = \sqrt{(u_v \cos\psi + v_w \cos\psi_w)^2 + (u_v \sin\psi + v_w \sin\psi_w)^2}, \tag{4.5}$$

where v_w is the wind speed in [m/s], ψ_w is the wind heading in [rad] and with the aircraft heading (ψ in [rad]) calculated using the law of sines:

$$\psi = \chi - \arcsin\left(\frac{v_w}{u_v} \sin\left(\psi_w - \chi\right)\right). \tag{4.6}$$

The model is discretized by the forward Euler method:

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k + T_s f(\mathbf{x}_k, \mathbf{u}_k), \tag{4.7}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ are the state and control inputs vector, respectively, in the discretization step k and T_s is the sampling period.

Model Predictive Control Problem

To reach the mission goal, a centralized optimization approach might not be feasible because the problem would be too complex with too many control inputs. In a non-convex problem with a very long vector of variables to optimize, falling very early in a local minimum is a common issue. In addition, the necessary processing power to optimize so many control inputs would be difficult to achieve by the on-board processing unit of the UAV. In contrast, optimizing the controls of all UAVs in a ground station would not be an ideal solution, due to communication range limitations and because that in case of a communication failure, the UAVs would not receive its controls, which could compromise the mission.

Therefore, in this research, the problem is addressed as a cooperative control problem, where each UAV optimizes its own control inputs to update its state so that a local cost function is minimized. The cost function also takes into consideration the planned states of the other UAVs. As each UAV follows the same process, it is expected that the global mission goal is achieved cooperatively. Collision avoidance between UAVs is also considered.

Considering I UAVs ($\mathbf{x}^i, \forall i \in \{0, ..., I-1\}$), the algorithm finds a control input sequence $U_k^i = \{\mathbf{u}_0^i, \mathbf{u}_1^i, ..., \mathbf{u}_{K-1}^i\} \in R^{2\times K}$ for the ith UAV, which solves the following optimal control problem:

$$\begin{aligned}
\text{minimize} \quad & \delta(\mathbf{C}\bar{\mathbf{x}}_K) + \sum_{k=0}^{K-1} L^i(\mathbf{C}\bar{\mathbf{x}}_k, \mathbf{u}_k^i) && (4.8)\\
\text{subject to} \quad & \mathbf{x}_{k+1}^i = f_d(\mathbf{x}_k^i, \mathbf{u}_k^i), && (4.9)\\
& v_{a_{min}} \leq u_{v_k}^i \leq v_{a_{max}}, && (4.10)\\
& \phi_{min} \leq u_{\phi_k}^i \leq \phi_{max}, && (4.11)\\
& |\mathbf{C}(\mathbf{x}_k^i - \mathbf{x}_k^j)| > r_c, \forall j \in \{0, ..., I-1\} \backslash \{i\}, && (4.12)
\end{aligned}$$

where

$$\delta(\bar{\mathbf{x}}_K) = F(\mathbf{C}\mathbf{x}_K) - aJ(\mathbf{C}\bar{\mathbf{x}}_K), \tag{4.13}$$

and

$$L^i(\mathbf{C}\bar{\mathbf{x}}_k, \mathbf{u}_k^i) = aJ(\mathbf{C}\bar{\mathbf{x}}_k) + b(u_{v_k}^i - u_{v_{k-1}}^i)^2 + c(u_{\phi_k}^i - u_{\phi_{k-1}}^i)^2. \tag{4.14}$$

Consider $u^i_{v_{-1}}$ and $u^i_{\phi_{-1}}$ as the commanded airspeed and roll angle for the ith UAV, respectively, in the previous optimization loop, $\bar{\mathbf{x}}_k = [\mathbf{x}^0_k, ..., \mathbf{x}^{I-1}_k]$ as the states of all UAVs, K as the number of horizon steps and r_c as the minimum safe distance between the UAVs to avoid collision. a, b, c are constant weighting factors and $\mathbf{C} \in R^{2\times 3}$ is used to define that only the x (north) and y (east) positions are used from the state vector:

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4.15}$$

In Eq. 4.13, the function J represents the grid search function, which is the sum of the rewards of visited cells, and F is the terminal cost (cost-to-go) function, which is the distance from the terminal position to the unvisited cell with highest reward. Both functions are described in detail in Section 4.2.4.

Optimization Technique

In this application, the Particle Swarm Optimization (PSO) [35] technique is used to find an optimal set of airspeed (u_v) and roll angle (u_ϕ) that minimizes the cost function. PSO is a meta-heuristic optimization method where the particles (solutions) are updated every iteration based on the best global and local solutions. In this application, a standard PSO algorithm was implemented using CUDA C programming language in order to benefit from the parallelism of the Nvidia Graphics Processing Unit that is assumed to be used in the UAV on-board computer.

The algorithm was set to run a fixed number of iterations on every loop. In addition, the number of particles must be defined. These two parameters affect the processing time and need to be fine-tuned according to the requirements.

The initial solutions are initiated with random values following the uniform distribution, where the minimum and maximum values are the defined boundaries of the airspeed and roll angle control inputs (Eq. 4.10 and 4.11).

4.2.3 SAR directives applied to UAVs equipped with remote sensing

The Search and Rescue (SAR) consists, according to the Department of Defense (DoD) of the United States of America, in "the use of aircraft, surface craft, submarines, and specialized rescue teams and equipment to search for and rescue distressed persons on land or at sea in a permissive environment" [33]. This work focuses on the sea cases, therefore, the following description emphasizes sea SAR missions. Also, as only Unmanned Aerial Vehicles (UAVs) are used in this work, only the directives for aircraft facilities are studied.

Search Area

According to the International Aeronautical and Maritime Search and Rescue (IAMSAR) Manual [73], the Total Adjusted Search Area (A_t), which is the mission's actual search area, is calculated based on the Total Available Search Effort (Z_{ta}), the Optimal Search Area (A_o) and the targeted Probability of Detection (POD). The first is a measure of the total area that a set of search facilities can effectively search within limits of search speed, endurance, and sweep width. The second is the search area which will produce the highest probability of success when searched uniformly with the available search effort and is essentially calculated based on the leeway and the Datum probable position error. Leeway is the the movement of a search object through water caused by winds blowing against exposed surfaces and Datum is a geographic point, line, or area used as a reference in search planning, such as the "Last Known Position" or the "Estimated Incident Position".

If the Total Available Search Effort (Z_{ta}) is smaller than the Optimal Search Area (A_o), a strategy must be chosen to balance the Probability of Detection (POD) and the Total Adjusted Search Area (A_t). Usually, the chosen strategy is to fly on higher altitudes, increasing the sensor's footprint or the crew's field of view while decreasing the POD. However, in this work, as UAVs equipped with automated remote sensing are assumed to be used, resolution requirements usually cannot be relaxed. Therefore, no trade-off between the POD and the search area is made and the POD is set to its maximum value of one, which makes the Total Adjusted Search Area (A_t) equal to the Total Available Search Effort (Z_{ta}).

In order to calculate the Total Available Search Effort (Z_{ta}), the sweep width (W) must be defined. When employing UAVs equipped with automated remote sensing in such missions, the sensor being used has a direct influence on this parameter. Altitude, view angle and image quality may affect the capability of identifying a survivor or an object on the sea. This is specially important to be taken into account because if the image does not contain the object of interest properly recorded, the computer vision algorithm will not identify it, independently of the ability that the algorithm has on identifying an important occurrence on an image. This can occur due to low image quality or too long sensing distances, making the object of interest imperceptible.

In the IAMSAR manual, the sweep width is calculated based on the altitude of the aircraft, the visibility and the sensor system specifications. In sea SAR missions with aircraft facilities and visual search, the Corrected Sweep Width (W) is adjusted regarding the weather, velocity and crew fatigue correction factors. However, these factors can be excluded from the equation when automated remote sensing systems are used and the system's velocity constraints are respected. Therefore, in this work, the Corrected Sweep Width (W) is considered equal to the original

Uncorrected Sweep Width (W_u).

Finally, the Search Effort (Z_a), which represents the area which can be covered by a specific facility, is calculated by:

$$Z_a = V \times T \times W, \tag{4.16}$$

where V is the Search Facility Speed (average speed) in [m/s], T is the Search Endurance in [s] and W is the Sweep Width in [m].

Note that the Search Endurance is the time available for the facility to fly looking for the survivors. The IAMSAR manual considers this time as 85% of the lower value between the Daylight Hours Remaining and the On-Scene Endurance. This is due to the fact that human crew is often only able to search with visible light. In contrast, UAVs are often capable to equip sensors that are not affected by that, such as infrared cameras, which allows the task to be done even along the night. This is a considerable advantage of using UAVs equipped with remote sensing systems.

By summing the Search Effort of all facilities, the Total Available Effort (Z_{ta} in [m^2]) can be found:

$$Z_{ta} = \sum_{f=1}^{F} Z_{af}, \tag{4.17}$$

where F is the number of facilities.

As described above, in this work the Total Adjusted Search Area (A_t) is equal to the Total Available Effort (Z_{ta}). Therefore, for Single Point Datum, the Length and the Width of the search area are given by the square root of the Total Available Effort (Z_{ta}) as defined by the IAMSAR manual.

Probability Map

The Probability of containment (POC) distribution in the search area is very important to guarantee efficient employment of the SAR facilities. When the initial indications do not provide enough information about the area, a standard distribution is assumed. The two most used types of standard distributions are the standard normal distribution and the uniform distribution, according the nature of the datum. For datum point and lines, the standard normal distribution is used. For datum areas, the uniform distribution is the most used. In this work, only the single point datum case is studied. Single point datum occurs, for example, when there is no significant leeway (e.g. when the target is a person in water [22]).

The probability map is a set of grid cells where each cell is labelled with the probability of containing (POC) the search object in that cell. As the the probability map follows a probability distribution function, the total sum of all cells should be equal to 100%. An example of probability table for single point datum with 12×12 cells is shown in Figure 4.4:

0.00%	0.00%	0.00%	0.01%	0.02%	0.03%	0.03%	0.02%	0.01%	0.00%	0.00%	0.00%
0.00%	0.01%	0.02%	0.06%	0.12%	0.17%	0.17%	0.12%	0.06%	0.02%	0.01%	0.00%
0.00%	0.02%	0.09%	0.24%	0.47%	0.65%	0.65%	0.47%	0.24%	0.09%	0.02%	0.00%
0.01%	0.06%	0.24%	0.65%	1.28%	1.79%	1.79%	1.28%	0.65%	0.24%	0.06%	0.01%
0.02%	0.12%	0.47%	1.28%	2.51%	3.52%	3.52%	2.51%	1.28%	0.47%	0.12%	0.02%
0.03%	0.17%	0.65%	1.79%	3.52%	4.93%	4.93%	3.52%	1.79%	0.65%	0.17%	0.03%
0.03%	0.17%	0.65%	1.79%	3.52%	4.93%	4.93%	3.52%	1.79%	0.65%	0.17%	0.03%
0.02%	0.12%	0.47%	1.28%	2.51%	3.52%	3.52%	2.51%	1.28%	0.47%	0.12%	0.02%
0.01%	0.06%	0.24%	0.65%	1.28%	1.79%	1.79%	1.28%	0.65%	0.24%	0.06%	0.01%
0.00%	0.02%	0.09%	0.24%	0.47%	0.65%	0.65%	0.47%	0.24%	0.09%	0.02%	0.00%
0.00%	0.01%	0.02%	0.06%	0.12%	0.17%	0.17%	0.12%	0.06%	0.02%	0.01%	0.00%
0.00%	0.00%	0.00%	0.01%	0.02%	0.03%	0.03%	0.02%	0.01%	0.00%	0.00%	0.00%

Figure 4.4: Initial probability table. Source: IAMSAR Manual [73]

4.2.4 Cost Function

An exploration cost function was developed based on [146] to search a given area.

The region of interest is divided into $M \times N$ square cells of a width (r_e in [m]), which value must be chosen to be smaller than the optical imaging sensor's footprint radius (R_e in [m]) times the square root of 2. The sensor radius is equal to the radius of the circle inscribed in the sensor's footprint. Figure 4.5 shows an example of a 4×4 grid with 100 m of cell width (r_e) and a UAV at position $\mathbf{Cx}$ equipped with a sensor with 100 m of radius (R_e).

The matrix $\mathbf{B}^i \in R^{M \times N}$ is used to identify if a cell was visited by the ith UAV. The matrix $\mathbf{b}^i \in R^{M \times N}$ is used to identify if a cell is planned to be visited by the ith UAV in the MPC horizon. In every $M \times N$ matrix used to identify if the cells are visited, each element has an associated value of 1 if the referring cell is visited or 0 if it is unvisited. Each cell has also an associated reward, given by $\phi \in R^{M \times N}$.

The function $J(\bar{\mathbf{x}}_k)$ is the sum of all cells associated value (1 or 0) in the step k times the correspondent reward:

$$J(\bar{\mathbf{x}}_k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \phi_{mn} y_{mn_k}(\bar{\mathbf{x}}_k), \tag{4.18}$$

with

$$y_{mn_k}(\bar{\mathbf{x}}_k) = \tag{4.19}$$

$$(\|\mathbf{C}\mathbf{x}_k - r^1_{mn}\| < R_e \wedge \|\mathbf{C}\mathbf{x}_k - r^2_{mn}\| < R_e \wedge \|\mathbf{C}\mathbf{x}_k - r^3_{mn}\| < R_e \wedge \|\mathbf{C}\mathbf{x}_k - r^4_{mn}\| < R_e) \tag{4.20}$$

$$\vee\, y_{mn_{k-1}}, \tag{4.21}$$

where r^1_{mn}, r^2_{mn}, r^3_{mn} and r^4_{mn} are the four vertices of the cell (Figure 4.5) and $y_{mn_{k-1}}$ is the associated value of the cell in the previous horizon step.

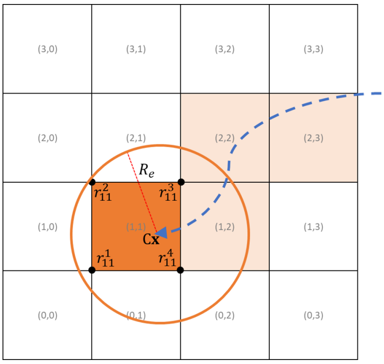


Figure 4.5: Cells grid example.

The starting value of y_{mn_0} is given by the logical sum of the matrices of already visited cells of all UAVs and the matrices of cells planned to be visited by other UAVs:

$$y_{mn_0} = \mathbf{B}^i_{mn} \vee \mathbf{B}^j_{mn} \vee \mathbf{b}^j_{mn}, \forall j \in \{0, ..., I-1\} \backslash \{i\}. \tag{4.22}$$

Finally, $F(\mathbf{x}_K)$ is the terminal cost. This function is necessary for the algorithm to consider the search beyond the prediction horizon by having a *cost-to-go* term. It is given as the minimum euclidean distance from the latest state of the UAV in the horizon, to the center of the closest unvisited cell, weighted by the correspondent reward, in the end of the horizon:

$$F(\mathbf{C}\mathbf{x}_K) = \min_{\forall m \in O, \forall n \in P} \frac{\|\mathbf{C}\mathbf{x}_K - \mathbf{r}_{mn_K}\|}{\phi_{mn}}, \tag{4.23}$$

where $O \subseteq M$ and $P \subseteq N$ are subsets of all unvisited cells and $\mathbf{r} = [x, y]$ are the north and east positions of the cell's center.

4.3 Results and Discussion

4.3.1 Software-In-The-Loop simulations environment

To evaluate the proposed solution, a Software-In-The-Loop (SITL) environment was set up using Ardupilot SITL simulator. This simulator allows to test the behavior of the Ardupilot code by running the Ardupilot in any computer without the Ardupilot hardware. The aircraft sensor data is simulated by JSBSim [17], an open source Flight Dynamics Model. Therefore, it is able to compute the UAV dynamics according to the actuator controls given by the Ardupilot code.

An aircraft platform model must be chosen for JSBSim flight dynamics calculations. In this work, the Skywalker X8 UAV platform (Figure 4.6) is used. The X8 is a UAV with around 4.0 to 4.5 kg of weight including the payload, 2.1 m of wingspan and 35.7 cm of mean aerodynamic chord. The aerodynamic model used to feed the JSBSim configuration file was developed based on wind tunnel tests [52]. In addition, the JSBSim source code was modified to use in its calculations the same wind map that is used by the MPC optimization.

Figure 4.6: X8 UAV. Source: NTNU

Figure 4.7 shows the interconnection between modules. For each UAV, an Ardupilot SITL instance must be started linked to a JSBSim module. Each Ardupilot instance uses a different TCP port. Therefore, one DUNE module must be started for each UAV, configured with the correspondent TCP port. Finally, Neptus [112], a command and control software, is used to visualize the UAVs telemetry and location and to give commands to the UAVs, such as take off, loiter and to start/stop the Search and Rescue mission.

4.3.2 Mission Simulation Scenario and Parameters

In this section, the parameters that define the mission scenario are described.

Aircraft and remote sensing platform

The X8 is a battery powered small UAV which can fly for around 80 min with the automated remote sensing payload and proper battery. The radius of the remote sensor's range is equal to 200 m, which is half of the width of the sensor's footprint. This footprint was chosen assuming that a computer vision algorithm, such as the

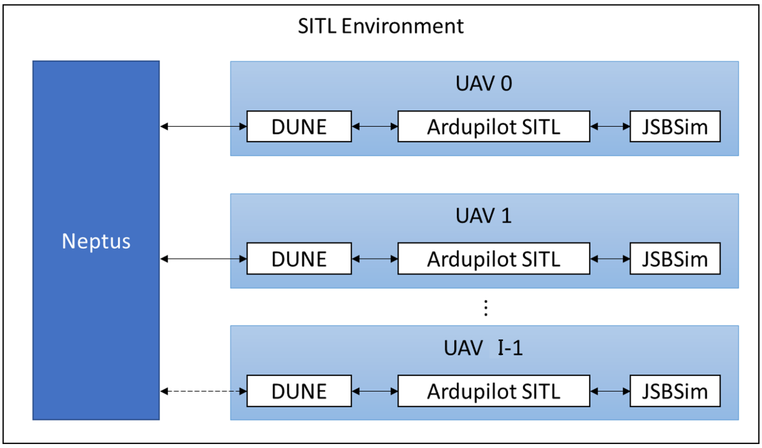


Figure 4.7: Software-In-The-Loop setup.

one described in the Appendix A, can detect the target in images captured at 400 m of altitude by an infrared camera with 7.5 mm of lens focal length, 640 × 480 pixels of resolution and 17 μm of pixel size.

Search Domain

The reference search area used in this work is equivalent to the Search Effort (Z_a) of one X8 UAV, calculated by Equation 4.16. Considering the total endurance of 80 min, the On-Scene Endurance (T) is equal to 60 min (85% of the total endurance). The Search Facility Speed (V) is equal to the average airspeed of the aircraft, in this case 16 m/s. The Sweep Width (W) is equal to 400 m, which is the lateral length of the required sensor footprint. Therefore, the search area is equal to 23.04 km^2, which gives a length and width equal to 4.8 km as the area has a squared shape because the datum is a single point.

Cells Grid

The grid was built with a cell width of 100 m. Therefore, the 23.04 km^2 of search area were divided into 48 × 48 cells. A two dimensional normal distribution curve was fitted to the single point datum reference table provided by the IAMSAR manual (Figure 4.4). The fitted curve of the probability (Figure 4.8), that gives the reward of each cell is given by:

$$\phi_{mn} = 0.002946 \exp\Big(- \Big(\frac{(m-23.5)^2}{108.28} + \frac{(n-23.5)^2}{108.28} \Big) \Big) \tag{4.24}$$

where m and n are the horizontal and vertical indexes of the cell, respectively.

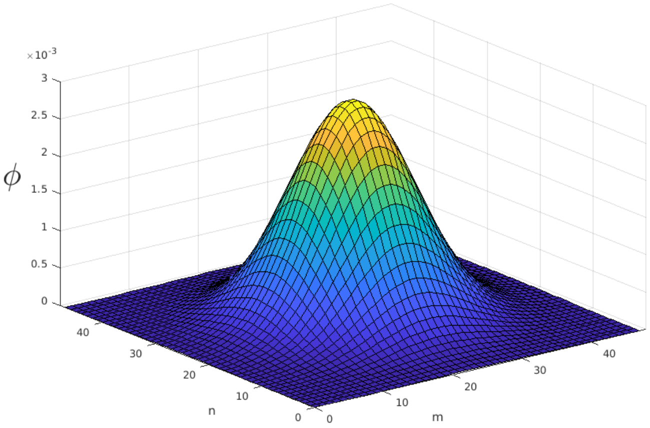


Figure 4.8: Reward of cells.

MPC Parameters

The airspeed range was chosen to be between 12 to 22 m/s. The reason for this choice was to keep the airspeed around the cruise speed, so that the battery consumption does not reach too high values. The roll angle range was chosen to be between -45 and 45 degrees so that the aircraft performs smoother maneuvers but still with freedom. The safe distance between UAVs was chosen to be 100 m and a wind of 9.9 m/s pointing to 45 deg was considered in the flight dynamics simulation.

A time horizon of 20 s and 20 horizon steps were the parameters chosen for the MPC problem. With, for example, a ground speed of 17 m/s, this means 340 m of straight distance, or a 180 deg turn. The weighting factor a was chosen to be 10000, because the rewards are of a very low value (the sum of all cells rewards is equal to one). The weighting factors b and c were chosen to be 1, in order to avoid unnecessary aggressive maneuvers.

Regarding the PSO parameters for the optimization, a total of 384 particles was used and the algorithm runs 35 iterations with local and global coefficients of 1.

Simulation platform

The optimization algorithm was written in CUDA C programming language in order to benefit from the parallelism, with the goal to embedded it on a NVIDIA Jetson board in the future for Harware-In-The-Loop simulations and field tests.

The simulations were run in a laptop with the NVIDIA 940MX graphics card, which has 384 CUDA cores and 8 GB 128-bit memory. With the parameters presented in this study, each UAV is able to run one optimization in around 400 ms when 3 UAVs perform the optimization at the same time. When only one UAV is used, each optimization loop takes around 250 ms. Therefore, it is expected that the NVIDIA Jetson board, which has 256 CUDA cores and 4 GB 64-bit memory will be able to run one optimization in less than 400 ms. If this performance is not achieved, the parameters can be fine-tuned to achieve shorter processing time. Another possibility is to implement an optimization stopping feature that will run as many iterations as possible within a given time, instead of a fixed number of iterations.

It is also relevant to mention that the optimization time of each step was adjusted to 400 ms also when using only one or two UAVs. This was done by inserting a delay, so each UAV's optimization time is the same for simulations when one, two or three UAVs are used. Therefore, this gives a fair comparison of the results.

4.3.3 Simulations

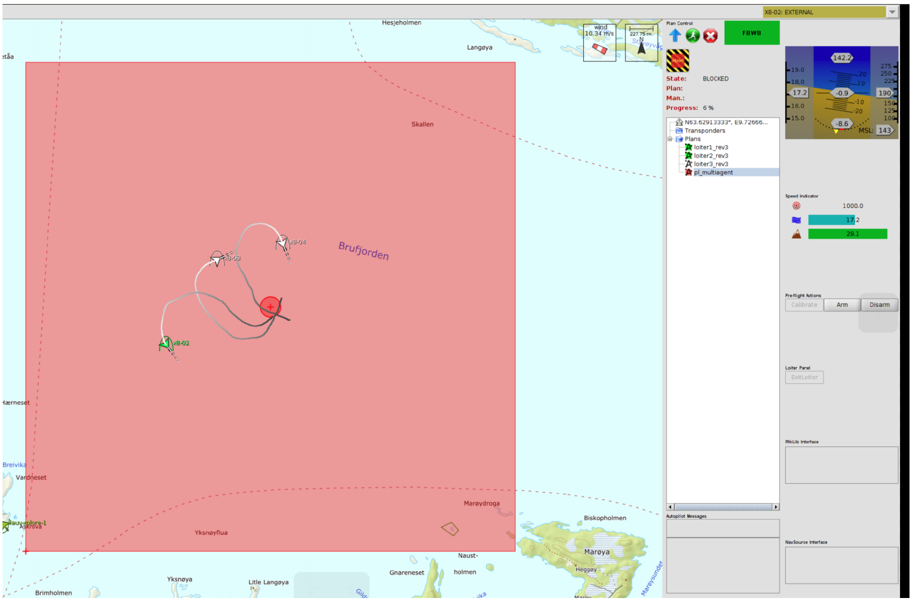
Three operational profiles were evaluated for the mission scenario: employing only one UAV; employing two UAVs; or employing three UAVs. Five missions were executed for each one of the profiles in order to obtain the average performance.

The reference search area was the Total Adjusted Search Area (A_t) for one UAV facility and Probability of Detection (POD) equal to 1, as described in the previous section.

The area was kept the same when employing two or three UAVs in order to allow a fair performance comparison between the profiles. Figure 4.9 illustrates one mission with three UAVs being monitored by the Command and Control software Neptus. The light red area is the search area and the dark red cross in the middle is the single point datum.

In all missions the UAVs departed from the same region (southeast of the search area as shown in Figure 4.10) where they were loitering and waiting for the mission start command. After receiving the command, the UAVs departed to the search area and the mission time started to count from when the first UAV collected the first reward.

The IAMSAR manual describes the Probability of success (POS) as the probability of finding the search object with a particular search. For each sub-area searched, $POS = POD \times POC$. It is therefore the way to measure search effectiveness. As the Probability of detection (POD) is kept at 1, the POS is equal to the POC of the area, which in this work is the sum of all rewards collected by the group of UAVs in the area.



State: BLOCKED
Plan:
Man.:
Progress: 6 %

N63.62913333°, E9.72666...
Transponders
Plans
loiter1_rev3
loiter2_rev3
loiter3_rev3
pl_multiagent

Figure 4.9: Snapshot of a mission with 3 UAVs being monitored with Neptus.

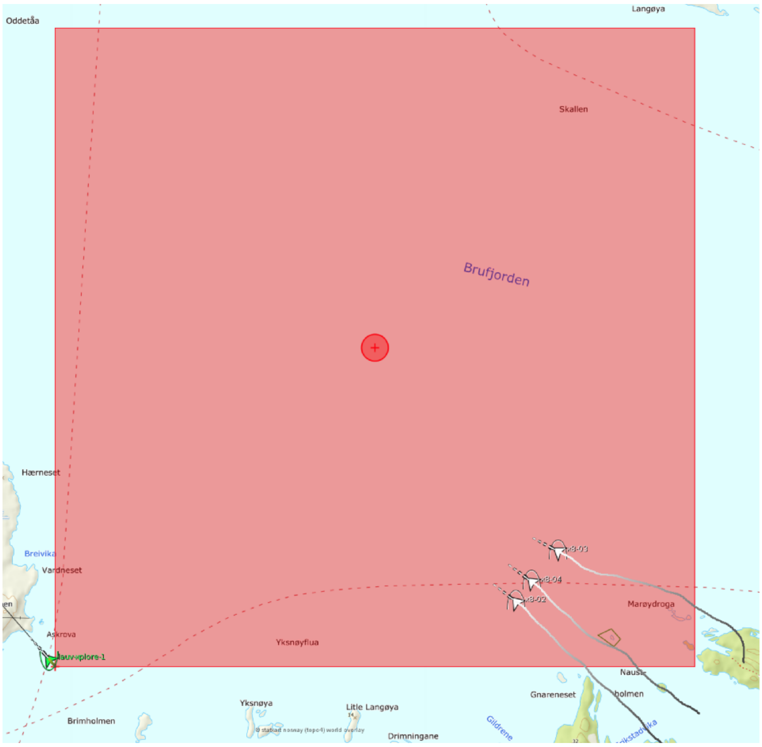


Figure 4.10: Snapshot of the beginning of a mission.

4.3.4 Results

The boxplot of the time to reach 50% of POS is shown in Figure 4.11 for the three operational profiles: employing one; two; or three UAVs. It is possible to notice that the gain when a pair of UAVs is used is very significant when compared to the single UAV profile, reaching 50% of POS 84% faster. When adding a third UAV, the gain was less significant: on average, the group reached 50% of POS 28% faster than when employing a pair of UAVs. The decrease on the gain is probably due to the fact that the UAVs are often flying over areas that have already been flown. A possible solution to avoid this situation is to reduce the width of the cells, increasing the resolution of the grid. Therefore, the UAVs would better tune their maneuvers and still have the cells inside the UAVs' sensor radius. However, this will increase the required computational power. This issue could be mitigated by optimizing the algorithm, for example.

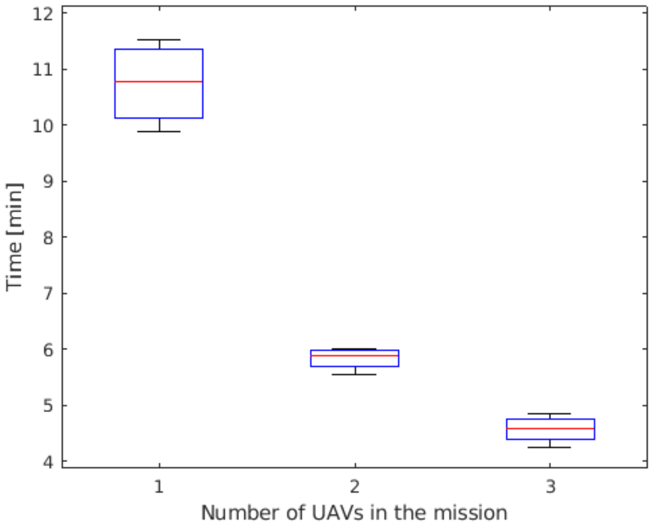


Figure 4.11: Time to reach 50% of Probability of success (POS).

Figure 4.12 shows the average POS during 20 min of mission. It is possible to notice that the results match the observed behavior when the missions were monitored. From the mission start to around 4 min, the UAVs fly to the area close to the datum, where the reward (Probability of containment) is higher. When two UAVs are employed, they fly parallel so that they cover more cells than when employing a single UAV. However, when three UAVs are employed, even if they form a parallel path, they fly close to each other and, therefore, do not visit many more cells than the pair of UAVs. This happens because in case the three UAVs get far enough from each other to not visit the same cells, they would take a longer path to arrive at the central area (highest rewards), not being a cost beneficial solution.

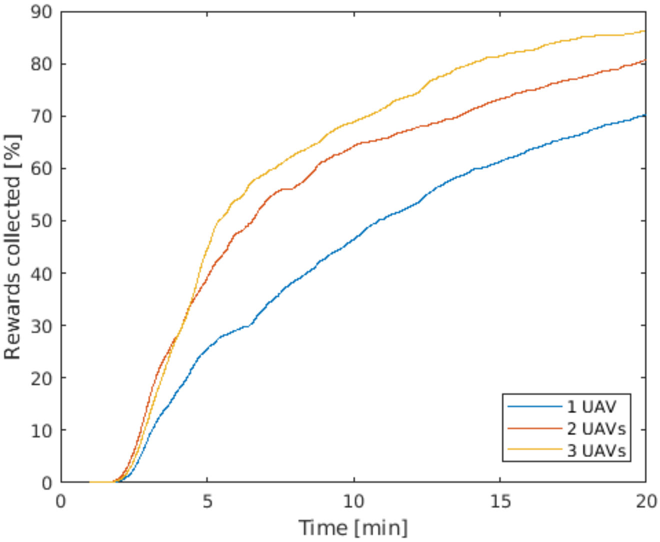


Figure 4.12: Average Probability of Success in time.

After reaching the area close to the datum at around 4 min, the curve of rewards collection grows steadily and the difference between the three operational profiles is clear. When three UAVs were employed, the group reached 50% of POS 2.35 times faster than the single UAV. The gain, however, reduces over time. For example, to reach 65% of POS, the group of three UAVs did it 2.07 times faster than the single UAV. The reason for the decrease on the gain is that the more cells are already visited and rewards collected, further the UAVs have to fly to visit new cells and collect new rewards (that are also lower in value). Therefore, the closer it is to the end of the missions, smaller is the difference between the performance of the different operational profiles.

Figure 4.13 shows the boxplot of the POS after 20 min of mission for the three operational profiles. According to the IAMSAR Manual calculations, the single UAV is expected to reach 100% of POS in 60 min. It is possible to notice that the group of three UAVs is able to reach close to 90% of POS in 20 min, showing that the improvement of adding extra UAVs is approximately linear.

In order to evaluate the benefits of using the wind information in the kinematic model and test the system's robustness, extra simulations were performed for: a scenario where the wind was not considered in the MPC; and a scenario where the wind considered in the MPC was underestimated by 20%. Results are shown in Figure 4.14. It is possible to notice that the performance is superior when the wind

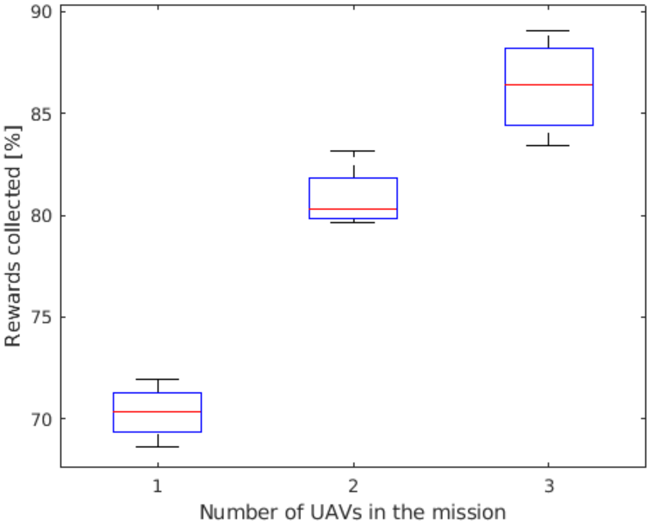


Figure 4.13: Probability of success (POS) in 20 minutes of mission.

information is used, collecting on average 4.4% more rewards in 20 min of mission time than when the wind is not considered in the model. The performance of the underestimated wind case was close to the ideal case, proving the system's robustness. It is important to notice that despite the higher median of the underestimated wind case, the ideal case was able to reach higher values of rewards collected having both the upper quartile and maximum value higher than the underestimated wind case.

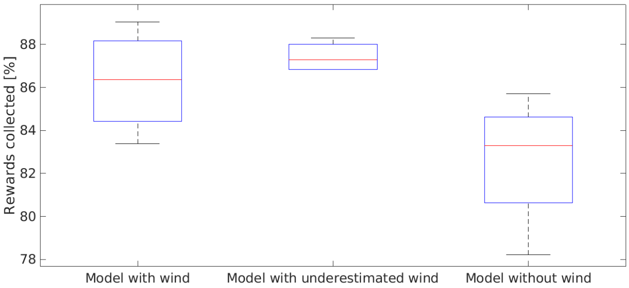


Figure 4.14: Probability of success (POS) in 20 minutes of mission.

Finally, a pre-made path was created (Figure 4.15) in order to compare to the performance of the single UAV with the real-time MPC optimization. In this path, the UAV flies from the mission origin to the grid's midpoint and then flies a spiral path. This spiral path is close to the standard path suggested by the IAMSAR Manual.

In the spiral path, the lanes are equally spaced allowing the best coverage by the sensor's footprint. This would be the best possible simple standard path for the mission scenario being investigated. Also, before starting the spiral path, the UAV is assumed to first fly to the center of the area.

Figure 4.16 shows that for the same average airspeed of 15.5 m/s, the performance of the MPC path planning was superior in the first 20 minutes of mission. Also, in the spiral path, 50% of POS was reached in around 13 min, while it took less than 11 min when the MPC optimization was used.

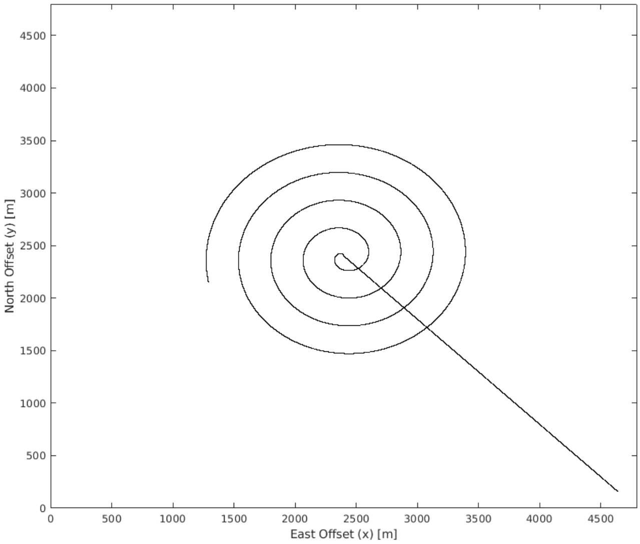


Figure 4.15: Spiral path.

In the spiral path, wind was not considered and the UAV keeps the ground speed constant, while in the MPC path planning the UAV optimizes its speed to reach higher coverage, for example reducing the airspeed to achieve a steeper turn when needed. Another advantage, that is perhaps the most important, is that the MPC solution has the capability to deal with dynamic changes on the environment and mission parameters during the mission, as it is a real-time optimization. These changes can be wind variations, updated search and rescue reports or even the lost of one UAV in the middle of the mission due to technical problems or the addition of extra UAVs that arrived later when the mission had already started.

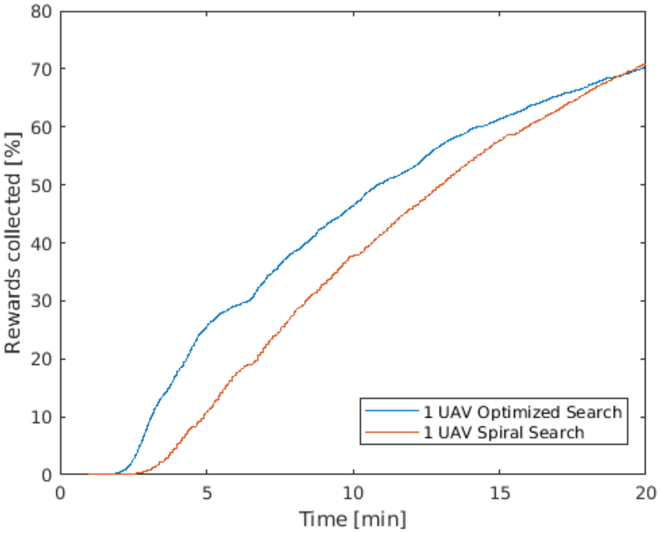


Figure 4.16: Probability of Success.

4.4 Conclusions

In this chapter, a real-time path planning for search and rescue with Model Predictive Control solved by Particle Swarm Optimization was proposed. The solution was implemented to be embedded in the UAVs' on-board computer and tested in a Software-In-The-Loop environment with flight dynamics simulations. In future work, Hardware-In-The-Loop simulations will be conducted in order to prepare the system for flight tests. The search area was defined using the International Aeronautical and Maritime Search and Rescue (IAMSAR) directives. Also, the area was divided into a grid of cells, where each cell had a correspondent reward, referred to the IAMSAR's Probability of containment. Results were analyzed for missions where one, two or three Unmanned Aerial Vehicles (UAVs) were employed. To reach 50% of Probability of success, the performance of the group of three UAVs was on average 2.35 times faster than the single UAV search. The effects of the inclusion of wind information were also evaluated. When wind information was used, even if the wind speed was underestimated by 20%, the team of three UAVs achieved on average 4.4% higher Probability of success in 20 minutes of mission, when compared to the scenario where no wind information was used. The performance of the single UAV was also compared to a standard search pattern based on the IAMSAR's suggested pattern. The search using the proposed solution outperformed the standard search pattern in the first 20 min, with the additional advantage of being a real-time method that can deal with environmental dynamic changes and new mission directives.

Part II

Long Range Off-Line Path Planning with Environmental Parameters

Chapter 5

Contingency Path Planning for Hybrid-electric UAS

This chapter presents a path planning optimization method which aims to mitigate the risks in the event of a critical engine or generator failure in hybrid-electric UAS. This is achieved through continuous determination of the optimum flight path, based on the remaining battery range and expected local wind conditions. The result is a dynamically adjusting flight path which ensures the aircraft to remain within range of pre-specified safe landing spots. The developed algorithm uses the Particle Swarm Optimization technique to optimize the flight path, and incorporates regional wind information in order to increase the accuracy of the expected in-flight performance of the aircraft.

5.1 Introduction

Hybrid-electric power trains used by long-range fixed-wing unmanned aerial vehicles often employ an internal combustion engine as the main source of power. It is witnessed that the internal combustion engine often is a critical point of failure. In such an event a functioning electric motor may still be able to propel the aircraft for a short period by utilizing the remaining battery capacity.

In an effort to contribute to the current scientific search towards path safety within autonomous decision-making, this chapter proposes a new method for contingent path planning optimization. The main goal of this study is to derive a method for autonomous path planning which ensures that the aircraft is able to reach a safe landing spot in the event of a critical engine or generator failure. This is done while taking into consideration the presence of expected local winds and their effect on the obtainable battery range. The resulting optimum path is found by applying the technique of Particle Swarm Optimization (PSO) [35].

In recent scientific literature path planning algorithm methods are described which use local wind information. In [2], the author integrates the uncertainty of the wind field into the wind model, and uses a Markov Decision Process for path planning. The goal was to minimize the power consumption of the aircraft and minimize time-to-goal. A similar approach was chosen in [77], where the technique of Ant Colony Optimization (ACO) is used to find the path which minimizes the travel time considering the wind. However, as with most studies the wind is considered constant during the flight. The ACO is a bio-inspired metaheuristic optimization algorithm suchlike the Particle Swarm Optimization used in this study. PSO is widely used for path planning, such as described in [92], where the author uses the method to minimize the UAS path's length and danger based on the proximity of threats.

The study presented in this chapter builds further upon the before mentioned methods by incorporating a dynamic wind model and translating this into the real-time in-flight performance of hybrid-electric UAS.

5.2 Aircraft performance model

The resulting achievable flight range is described in the following subsections. This is illustrated by analyzing the different segments of a flight that suffers a critical engine or generator failure. The flight segments are divided into normal operations, battery-powered flight, and unpowered glide.

5.2.1 Aircraft Power Model

For a conventional propeller-driven aircraft in level and unaccelerated flight, the power that is required for obtaining the maximum flight range is expressed, in watts, by:

$$P_{rmr} = W \left(\frac{C_D}{C_L} \right)_{min} \cdot v_{TAS}, \tag{5.1}$$

where W is the aircraft total weight in Newton, $(C_D/C_L)_{min}$ is the minimum obtainable ratio between the aerodynamic drag and lift coefficients, and v_{TAS} is the true airspeed occurring at the $(C_D/C_L)_{min}$ condition.

One advantage of utilizing a hybrid-electric power train is that there is the possibility of co-powering the main drive shaft (continuous or intermittent). Depending on the sizing of the hybrid system and mission specifications, this may prove to be beneficial for the resulting range. A method for determining the achievable range in the case of co-powering of a hybrid system is described by Marwa in [99]. In the remainder of this analytical study it is further assumed that a functioning main generator set is sized so that it is capable of supplying the necessary power required for maintaining the maximum range cruise speed (v_{mr}). Therefore during cruise it is given that the necessary amount of co-powering by the electric motor is zero.

5.2.2 Aircraft range - Normal operations

In the case of a hybrid powered aircraft where the electric co-powering equals zero, the obtainable maximum range may be modelled similarly to conventional fuel-powered aircraft. The Breguet range equation is a commonly used first-order approximation to determine the achievable maximum range of a conventional propeller aircraft [38]. When assuming no wind and a parabolic drag polar, the resulting maximum range in normal operations (R_{no} in meters) is expressed by [5]:

$$R_{no} = \frac{\eta_{pg}}{c} \cdot \left(\frac{C_L}{C_D}\right)_{max} \cdot ln\left(\frac{W_0}{W_1}\right), \tag{5.2}$$

where η_{pg} is the complete propulsion efficiency of the hybrid system, c is the specific fuel consumption of the generator in Newtons per second per watt, W_0 is the aircraft's total weight at the beginning of the cruise flight, W_1 is the aircraft's total weight at the end of the cruise flight, and $(C_L/C_D)_{max}$ is the maximum achievable ratio between the aerodynamic lift and drag coefficients in level and unaccelerated flight.

5.2.3 Aircraft range - Battery-powered flight

In a situation where the main engine or generator fails, the hybrid system remains able to supply the power necessary to propel the aircraft by utilizing the remaining battery capacity. To obtain the performance model the aircraft is considered purely battery-powered. The adopted method for determining the maximum achievable range of battery-powered sUAS has previously been described in [71]. The adopted method in that study is an extension on the classical determination of battery-powered aircraft range, by including the Peukert effect on the battery capacity. This allows for a more accurate determination of the aircraft's battery range [34][144]. When assuming no wind and a parabolic drag polar, the maximum range (R_{bp}, in kilometers) for battery-powered sUAS in level and unaccelerated flight, without the influence of wind, is expressed by:

$$R_{bp} = \left(\frac{V \times C\,\eta_{pe}}{W\left(\frac{C_D}{C_L}\right)_{min}}\right)^{n} \left(\sqrt{\frac{2W}{\rho_\infty\, S\, C_L}}\right)^{1-n} \cdot {R_t}^{1-n} \cdot 3.6. \tag{5.3}$$

Here V is the battery bus voltage, C is the battery capacity, n is the battery-specific Peukert constant, R_t is the battery hour rating (i.e. the discharge period at which the rated capacity C was determined), and η_{pe} is the propulsion efficiency of the battery-powered system. The amount of experienced resistance depends on the specific set-up, and the availability of mitigation systems, such as mechanical decoupling.

5.2.4 Aircraft range - Unpowered glide

After the main batteries have been drained, the aircraft may be able to fly further by exchanging its altitude for range. This flight phase is modelled as pure unpowered glide, depending solely on the altitude (h), and maximum glide ratio $(L/D)_{max}$. When the aircraft's glide angle is moderate the maximum range for unpowered glide in no-wind condition and for flat terrain is expressed by [6]:

$$R_{ug} = h \left(\frac{L}{D} \right)_{max} . \tag{5.4}$$

5.2.5 Effects of Wind

Small unmanned aircraft are often operating in relatively high wind speeds - commonly exceeding half of the true air speed. Depending on the speed and direction, en-route winds may have a significant influence on the obtainable range of the aircraft. Therefore, in an effort to accurately estimate the maximum obtainable range of an aircraft, one has to include the wind effects. This study aims to describes a method for determining the maximum obtainable range, while including the effects of horizontal winds that are encountered en-route. Note that this wind model shall be applied to each segment of the flight.

In [57] it is described how the optimum airspeed may be determined when accounting for head- and tailwinds. The author states that for propeller-powered aircraft the optimum airspeed is obtained through:

$$m_{br} = \frac{v_{TAS}}{v_{br}} = \left[\frac{2\, m_{br} \, \pm \, \left(\frac{v_w}{v_{md}} \right)}{2\, m_{br} \, \pm \, 3 \, \left(\frac{v_w}{v_{md}} \right)} \right]^{\frac{1}{4}} . \tag{5.5}$$

Here m_{br} is the relative airspeed parameter between the true airspeed and the best-range airspeed (v_{br}). v_w is the wind speed, with ± indicating a head- or tailwind, v_{md} is the minimum-drag airspeed. Solving for m_{br} yields the optimum ratio to achieve the best range in the event of head- or tailwind. As Eq. (5.5) only takes into account head- or tailwinds along the flight path, it is necessary to include and isolate the crosswind component that may be encountered en-route. When assuming a flat and non-rotating Earth and flying in level and unaccelerated flight, the equations of motion through decomposed wind vectors are modelled as:

$$v_{GS} = v_{TAS} \begin{bmatrix} cos\, \phi \\ sin\, \phi \end{bmatrix} + v_w \begin{bmatrix} cos\, \theta_w \\ sin\, \theta_w \end{bmatrix} . \tag{5.6}$$

Here v_{GS} is the aircraft's ground velocity vector, and θ_w is the direction of the wind. The aircraft's commanded heading (ϕ) is the sum of the course angle (θ) and the crab angle (β). Here β is defined as the angle between the TAS vector and the ground course angle. The wind components perpendicular and parallel to the resulting ground track, in relation to the reference horizontal path, can be found by rotating the wind's x and y components through angle θ, resulting in [153]:

$$\begin{aligned} v_{\|} &= v_{wN}\, \cos\,\theta + v_{wE}\, \sin\theta \\ v_{\perp} &= -v_{wN}\, \sin\,\theta + v_{wE}\, \cos\theta. \end{aligned} \tag{5.7}$$

Considering the horizontal wind field to be described in the NED (North, East, Down) frame, then v_{wE} is the decomposed East wind component, and v_{wN} is the decomposed North wind component. $v_{\|}$ is the wind component parallel to the aircraft's ground course and $v_{\perp}$ is the perpendicular wind component to the ground course. In relation to Eq. (5.5) the parallel wind component constitutes the value for v_w.

5.3 Path Planning

In this section, the optimization problem formulation and the cost function that has to be minimized by the Particle Swarm Optimization (PSO) are presented. The Apeendix contains a detailed description of the PSO algorithm.

5.3.1 Optimization Problem Formulation

In this study the goal is to find a safe path with the shortest length. Therefore, the cost function ought to take into consideration both the path's length and the safety. Here a safe path is defined as a path in which the aircraft, in the event of a critical engine or generator failure, is within flight range of a pre-specified safe landing spot.

A two-dimensions geographical approach is used in this work, where the optimization variables represent a set of waypoints of the path, with x (North) and y (East) positions in the NED frame. As the input positions of the origin, destination and landings spots are given in latitude and longitude coordinates, a conversion to the NED frame is needed. Besides, to use the result as an input for an Autopilot system it may be required to convert the waypoints to positions expressed in latitude and longitude.

The domain has to be defined taking into consideration that the UAV may not deviate too far from the straight line path between the departure and destination.

To initialize the optimization algorithm, first a straight path from the origin to the destination is generated - with waypoints distributed equally along the path. This strategy is crucial, as usually the optimal solution will be a deviation from this straight path. If only particles initialized with random positions are used, they might have uncommon waypoints displacement, causing the algorithm to take longer time to find an optimal solution or to fall into a local minimum.

The stop criteria used is straight forward. The algorithm runs until it reaches a pre-defined number of iterations.

5.3.2 Cost Function

Always flying within range of a safe landing spot may not always be desired (or realistic), since this may cause the path to be too long. For that reason a cost function (f) is proposed that employs a method for weighing the importance of the safety against the importance of the path length. As shown in the following equation, α represents the weight of the path's length over the safety. This results in:

$$f = (\alpha)\, length + (1-\alpha)\frac{1}{safety}, \tag{5.8}$$

where *length* and *safety* are relative values and in this study presented as percentages. The function is calculated based on the minimum length, which is the straight line between the departure and destination, as per Eq. (5.9), and the maximum safety, if all points along the path are safe, as per Eq. (5.10).

$$length = \frac{\text{current length } (L_{cur})}{\text{minimum length } (L_{min})}. \tag{5.9}$$

$$safety = \frac{\text{number of safe points } (n_{safe})}{\text{total number of points } (N_{path})}. \tag{5.10}$$

To ensure that the safety is evaluated for the entire path, and not only for the position of the waypoint, an interpolation is performed to discretize the path into N_{path} number of points. The effective flight range using battery power is calculated to all landing spots that are not further than the maximum range which the UAV could fly by taking into consideration a vector of wind that would provide the maximum range, thus, landing spots very far away are not considered in that step of the algorithm, saving computational power. As the wind might change along the path between the point and the landing spot, the range is recalculated every R_{step} considering the closest wind vector.

For each point (n), it is evaluated if it is close enough to at least one landing spot (j). Then one unity is added to the safety result variable, demonstrated by:

$$n_{safe} = \sum_{n=1}^{N_{path}} \begin{cases} 1 & \text{if } (r_1^n < d_1^n)||...||(r_j^n < d_j^n) \\ 0 & \text{otherwise} \end{cases}, \tag{5.11}$$

where r_j^n is the maximum obtainable range in case of an engine failure (section 5.2) from point n to the landing spot j and d_j^n is the distance of the aircraft from point n to the landing spot j.

The total length is the sum of the distance between the points of the interpolation:

$$L_{cur} = \sum_{n=1}^{N_{path}} \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}, \tag{5.12}$$

where x_n and y_n are the North and East positions in the NED frame at point n, respectively.

5.4 Case study and Experimental setup

5.4.1 Case Study

The scenario chosen for this simulation comprises the area located to the north of the Norwegian city of Trondheim (Figure 5.1). For this study the Norwegian Defence Research Establishment (FFI) made regional wind models available for calculating wind speeds and directions for different altitudes with the resolution of 2.5 km. The date of the data retrieval was July 5th 2017, with wind information valid between 03:00PM and 09:00PM of that day. The orange polygon in the figure illustrates for which region the information was available. The locations of the origin (A) and destination (B) were chosen so that the maximum length across the wind model's specified area was utilized. The straight-line path distance between origin and destination is 210 kilometers, illustrated by the red dotted line, and has a path safety of 72.6%. The safe landing spot locations were manually selected through studying satellite imagery. The initial cruise altitude is chosen to be 1500 meters, meaning it can pass any mountain the aircraft may encounter within the specified region.

5.4.2 Experimental Setup

This case study utilizes the P31015 concept UAV, which is a fixed-wing aircraft in conventional pusher configuration. In this theoretical study the model shall employ a hybrid propulsion system with sufficient capacity to allow continuous in-flight recharging of the batteries - effectively enabling fully charged batteries during cruise flight. The total propulsion efficiency through the generator (η_{pg}) is assumed to be invariable at 28.5% while the specific fuel consumption c equals 383 grams per horsepower per hour. The direct electric propulsion efficiency (η_{pe}) is assumed to be invariable at 50%. These values lie within range of experimental data found in [34], [101] and [74]. At the initiation of the cruise phase the aircraft is modelled to carry 5.5 liters of on board fuel, and to have a fully charged battery with a

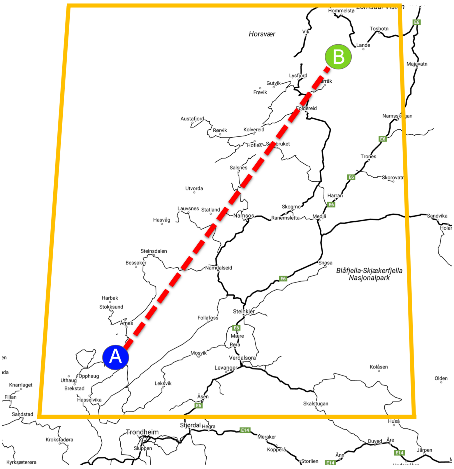


Figure 5.1: Simulation scenario

usable nominal capacity of 10 Ampere-hours at 44.4 Volts. The total weight of the aircraft is assumed to be 17.5 kilograms. More detailed aircraft specifications may be found in [71].

The theoretical maximum obtainable range of the aircraft at any point in time during the cruise phase is calculated as the sum of R_{no}, R_{bp} and R_{ug}. As R_{ug} is defined as the maximum obtainable range related to flat terrain, this requires to be re-evaluated when considering operating in mountainous terrain, such as the selected region for this experiment. The terrain in the selected region is mountainous, and contains elevated fields nearly as high as the selected cruise altitude. Therefore the actual obtainable glide range is considered too variant. Thus, in this specific simulation it is decided to leave out the potential range of unpowered glide, leaving the sum of R_{bp} as the maximum obtainable range in case of an engine failure.

5.4.3 MATLAB Code description

A MATLAB script was written to perform the simulation. The main user inputs are the WGS-84 coordinates of the origin, destination and safe landing spots. The wind information obtained was previously saved in a .dat file, which is used for the evaluation of the cost function. The airframe characteristics and battery efficiency, as detailed in previous sections, need to be configured. The optimization parameters are shown in the next subsection. The script is set up so that it runs the optimization algorithm until a pre-determined number of iterations is reached. Figure 5.2 shows a simplified block diagram of the script.

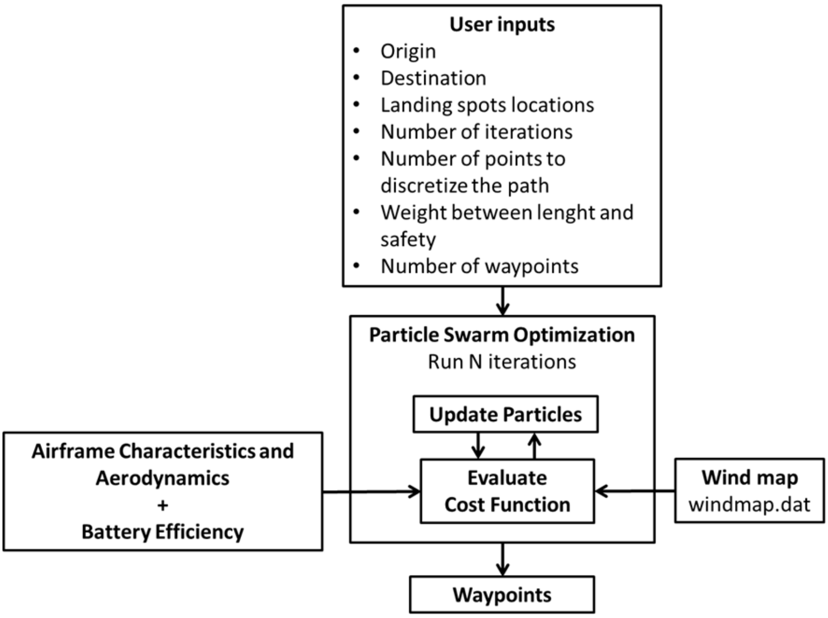


Figure 5.2: Overall block diagram

5.4.4 Parameters and Optimization Algorithm

Some parameters need to be defined in the optimization algorithm. Several of these parameters may strongly affect the convergence speed of the algorithm, and can prevent it from falling into local minima. Among these are the PSO velocity constraint, which was chosen to be attached to the domain in order to be automatically changed according to the problem; and the initial and final inertia weight, which were chosen to range from 1.0 to 0.1, to allow a more global search at the beginning, and a more local search in the course of iterations. However, other parameters can hardly be modified as they are specified by the user's requirements, which affects the processing time. For instance, this is the case with the number of points which discretize the path (N_{steps}). The parameters used in this simulation are presented in the table below, where the domain is defined by $[x_{min}, x_{max}, y_{min}, y_{max}]$:

5.5 Results

Figure 5.3 shows the simulation result in the NED frame for a simulation using $\alpha = 0.3$. The yellow square and the green star represent the origin and destination respectively. The red circles are the generated waypoints, while the red crosses represent the safe landing spots. The path's discretized points are presented as

Table 5.1: List of parameters

Name	Value
Number of iterations	100
Particle velocity constraint	0.1 x Domain
w_{ini}	1.0
w_{fin}	0.1
N_{steps}	84
Safe landing spots	5
x_{min}	x_s-$L_{min}/2$
x_{max}	x_t+$L_{min}/2$
y_{min}	y_s-$L_{min}/2$
y_{max}	y_t+$L_{min}/2$
R_{step}	5 km

black dots. The arrows represent the distance from the point of the path to a landing spot which the UAS can reach during a critical engine failure. This also takes into account the effects of wind. What may be observed is that the projected path is always close enough to at least one of three safe landing spots, resulting in two landing spots to not be utilized within this scenario. It was found that in this specific scenario the calculated path has a safety factor of 100%, with a total length of 215.4 kilometers. This route is 5.4 kilometers longer than the straight path distance between origin and destination. The convergence of the algorithm is shown in Figure 5.4, where the vertical axis refers to the Cost Function, Eq. (5.8).

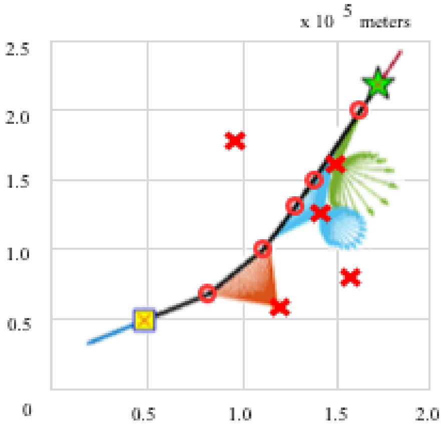


Figure 5.3: Result for $\alpha = 0.3$

In an attempt to study the effects of the positioning and amount of safe landing

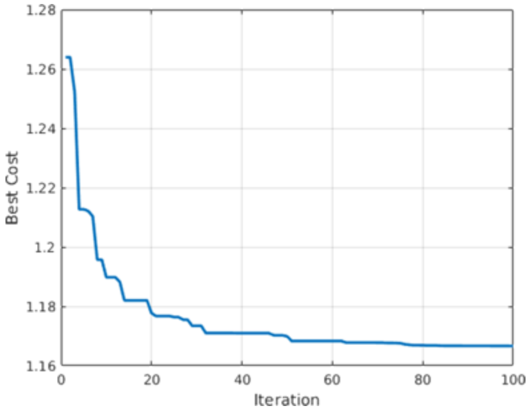


Figure 5.4: Algorithm convergence

spots, a second simulation was conducted where the first landing spot was removed, as illustrated in Figure 5.5. Here it is impossible to reach a safety score of 100% due to the distance between the landing spots. The grey dots forming a line indicates the part of the path where the UAS can not reach a landing spot in case of engine failure. In this specific case it was found that for $\alpha = 0.3$ the rated path safety was 83.3%, while the path distance had increased to 231.4 kilometers.

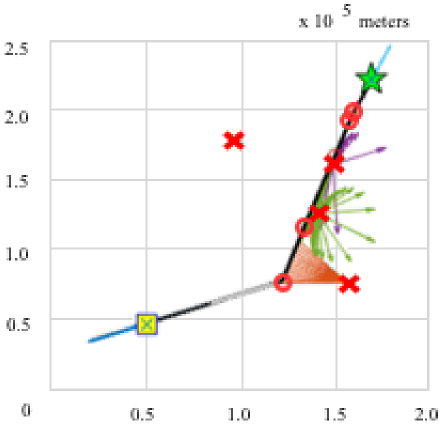


Figure 5.5: Result without LS1 for $\alpha = 0.3$

In a third scenario where five safe landing spots were placed at different locations, the simulation for $\alpha = 0.3$ resulted in a path with length 229.4 and safety of 100%. When $\alpha = 0.7$ was chosen, the length was shorter (219.9 km), while the safety was 90.48%.

Therefore in this scenario it was found that compared to the straight-line trajectory, the trade-off for $\alpha = 0.3$ is 27.4% increased safety against an increase of

19.4 kilometers in path length. When using an α of 0.7, the trade-off was of 17.9% increased safety against 9.9 kilometers increase in path length, when compared to the straight-line trajectory.

Figure 5.6 illustrates a magnified part of the path, including the plotting of the wind vectors (orange arrows). The blue arrows represent by size the range that the UAS can obtain using the electrical battery from the discretized point of the path (black dots) to the landing spot (red cross). It may be observed that the wind is pointing south, resulting in a larger obtainable range when flying North to South, compared to flying South to North.

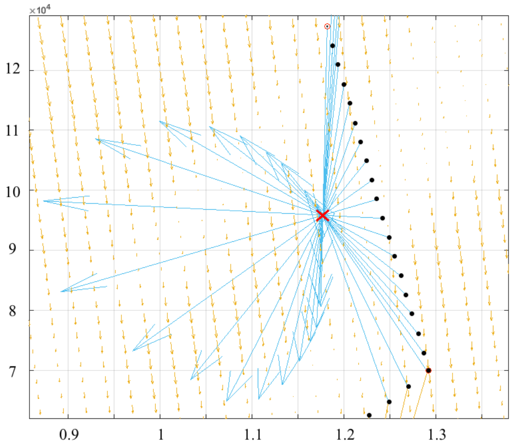


Figure 5.6: Extract of path including the wind vectors

5.6 Limitations

The suggested methods are evaluated through performing simulations. The physical input parameters, such as the wind, are predicted values based on weather model simulations. The suggested method does not include vertical winds, nor was this information available. The aerodynamic and propulsion efficiency parameters that were used are drawn from a concept aircraft and hybrid system, with assumed performance characteristics. Although the geographic model is realistic, the safe landing spots were chosen arbitrarily. All before mentioned limitations may influence the accuracy of the proposed model, and therefore a verification of the model may be warranted.

5.7 Discussion

To eliminate the limitations described in the previous section, and to further verify the proposed model, more simulations are needed. Besides the verification, a validation through flight test is to be conducted through utilizing an aircraft with known performance characteristics. One desirable improvement is to optimize the algorithm to reduce the processing time, so that in-flight real-time recalculation becomes a possibility.

Though it is beyond the scope of this study, one might consider using a more sophisticated glide performance model, such as described in [28] which also includes the effects of turn performance on the achievable range, and consider [41] and [148] which describe an optimization method for the in-flight performance in variable (and altitude dependent) winds.

5.8 Conclusion

In this chapter a method was proposed that aims to increase the operational safety of hybrid-electric powered UAVs. This was done by taking into account the possibility of direct-electric propulsion in case of main generator failure. It was demonstrated that the aircraft's projected flight path can be adjusted so that it remains within range of pre-specified safe landing spots. This is done while taking into account the pre-calculated effects of winds that are encountered en-route by including an altitude-dependent wind model. As it is not always possible (or desirable) to have a completely safe path, the proposed method includes a cost function in which the user may specify the importance of path safety over the path length. A first simulation ($\alpha = 0.3$) with arbitrarily picked safe landing locations shows a path safety of 100%, while having a total path length of 215.4 kilometers. This is opposed to the straight path distance of 210.0 kilometers which offers a path safety of 72.6%. In this instance it is concluded that the trade-off may be an increased path safety of 27.4% at the cost of a longer path length of 5.4 kilometers. In a second simulation (also $\alpha = 0.3$) one central safe landing spot was removed. This simulation resulted in a path safety of 83.3%, while having a total path length of 231.4 kilometers. The trade-off may be determined through a similar approach. A third scenario was tested where the five safe landing spots were placed at a different location, while α was chosen to be 0.3 and 0.7. In this scenario it was shown that compared to the straight-line trajectory the trade-off made for $\alpha = 0.3$ is 27.4% increased safety against an increase of 19.4 kilometers in path length, while for $\alpha = 0.7$ this is 17.9% increased safety against 9.9 kilometers increase in path length. The study presented here was a theoretical study based on simulations utilizing hypothetical aircraft systems. To further validate and verify the proposed methods, a future study with more diverse simulated scenarios is suggested, and to perform test flights conducted with an aircraft that has known performance characteristics.

Chapter 6

Inclusion of Horizontal Wind Maps in Path Planning Optimization of UAS

Earlier studies demonstrate that en-route atmospheric winds significantly affect the in-flight performance of unmanned aircraft. Nevertheless today the inclusion of wind is not common practise in determining the optimal flight path. This chapter aims to contribute with an accessible method that includes forecast horizontal wind maps which are commonly available, and discuss the methods on how these maps can be integrated in order to obtain the most energy efficient horizontal path of fixed-wing aircraft. The benefits of including horizontal wind maps into the path planning optimization are demonstrated through a simulation, which utilizes Particle Swarm Optimization to find the optimal cost-beneficial path.

6.1 Introduction

Atmospheric winds pose constraints on the operations of unmanned aircraft. This holds especially true for smaller aircraft, as here it is common for wind speeds to constitute 20-50% of the airspeed [13]. This has a substantial effect on the mission safety and the aircraft's in-flight performance. It is therefore considered to be warranted to account for atmospheric winds in the planning of the aircraft's flight path. As the unmanned aircraft industry is maturing, a growing scientific search towards in-flight performance optimization is noticed. Accurate estimations of the aircraft's in-flight performance allow for optimal utilization of the system within its specified mission objectives.

Early studies demonstrate the advantages of utilizing atmospheric winds in the aircraft's route optimization [25]. More recently efforts have been made to include the complete wind field in the optimization, such as found in [48], and more recently in [42], which utilize the Ordered Upwind Method and the stochastic Dijkstra algorithms, respectively, for determining the optimal flight path.

The study of path planning optimization in the context of *unmanned* aircraft is relatively new but abundant. Most notably, in [104] a method is presented that successfully incorporates wind fields in path following methods utilizing straight-line and circular arc paths. In [123] a sophisticated method was described where Model Predictive Control (MPC) methods were employed for path planning optimization, while including the effects of wind. However, neither studies included the effects of wind on the aircraft performance within the optimization, such as was the case in [42], which describes a method for the incorporation of weather uncertainty for manned aircraft in long-distance flights. In [29] the aircraft performance was successfully included, with the assumption of a constant wind field.

More recent sophisticated wind-energy harvesting methods have received increased scientific interest. Most notably in [53] two refined methods are described which utilize updraft winds from locally observed wind-fields in order to extend the aircraft's range and endurance. Considering such complex wind fields offers the potential of effective path optimization. However, the limitation of such methods in context of the study presented in this chapter is that it relies on the availability of detailed local wind measurements and terrain observations or maps. In practise the extraction of lift due to *vertical* winds over terrain, known as orographic lift, is relatively complex to obtain [100]. This is in contrast to forecast horizontal wind gradients which are relatively well described, and are commonly obtainable through meteorological institutions.

The study presented in this chapter positions itself in the current literature by describing an accessible method that includes forecast horizontal wind maps, and discuss the methods on how these maps can be integrated in order to obtain the most energy efficient horizontal flight path of fixed-wing unmanned aircraft. To achieve this, it specifies and includes the effects of horizontal winds on the in-flight performance of the aircraft. In this chapter the Particle Swarm Optimization (PSO) technique is applied, such as described in the Appendix B, in order to simulate how the inclusion wind affects the flight performance. PSO was chosen because it is easy to implement, there are few parameters to adjust and it uses global and local performance, which is advantageous in this type of problem where it is expected that the optimal solution is likely to be a relatively small deviation from the known solution candidate - the straight path.

The goal of the developed algorithm is to find the path which minimizes the total energy consumption from origin to destination by using a forecast wind map and by optimizing the path and airspeed of the aircraft. Minimizing the energy consumption results in a lighter aircraft as battery-powered aircraft are required to carry fewer on-board batteries, while fuel-powered aircraft require to carry less block fuel. Alternatively, one could consider the reduction in required fuel/batteries to increase the cargo capacity of the aircraft, or to offer a larger safety margin through energy reserves.

6.2 Aircraft dynamic model

The aircraft's kinematic model is described through the North-East-Down (NED) inertial reference frame. As the aim is to optimize the energy consumption per distance travelled, it is necessary to describe the wind field in a similar way. Because in this study horizontal wind maps are used, the wind field is being described in a two-dimensional plane.

When assuming a flat, non-rotating earth then x aligns north, y aligns east, and z is pointing down to earth as positive direction. Relating to the wind navigation triangle, as shown in figure 6.1, the aircraft's inertial velocity in a coordinated flight can be described as a function of the aircraft's ground course χ and ground speed v_g. Similarly, this can be described as a function of the true airspeed v_a, heading ψ, wind speed v_w and wind speed direction ψ_w. These relations are found through:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = v_g \begin{bmatrix} \cos\chi \\ \sin\chi \end{bmatrix} = v_a \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix} + v_w \begin{bmatrix} \cos\psi_w \\ \sin\psi_w \end{bmatrix} \tag{6.1}$$

The relation between heading and course angle is conveniently described using the law of sines, resulting in [123]:

$$\psi = \chi - \arcsin\frac{v_w}{v_a}\ \sin\left(\psi_w - \chi\right) \tag{6.2}$$

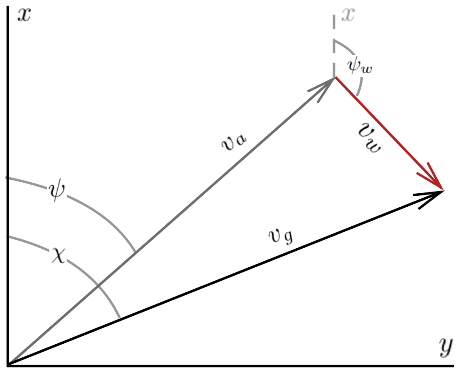


Figure 6.1: Wind Navigation Triangle in Coordinated Flight

In aviation the wind maps and directional indications are often expressed in the navigation representation, rather than the mathematical representation. Therefore it is considered convenient to apply the same standards here. Here the directional indications are related to x (true north), where the clockwise rotation is positive. Note that wind maps commonly indicate the direction where the wind is coming from, rather than where it is going towards.

In an attempt to more accurately determine the aircraft's in-flight performance, one may consider including the parallel wind speed along the aircraft's heading $v_{w\psi\|}$. When the wind components are decomposed as demonstrated in equation (6.1), then the wind speed $v_{w\|}$ can be found through:

$$v_{w\|} = v_{wx}\cos\psi + v_{wy}\sin\psi. \tag{6.3}$$

6.3 Effects of wind on in-flight performance

Depending on the magnitude and direction of the wind in relation to the aircraft's desired ground path, the presence of wind has an effect on the in-flight performance. In a typical mission considered in this study most in-flight time will be spent during the cruise phase. In the context of path planning optimization it is therefore the cruise phase that is considered most relevant. The remainder of the study shall consider path planning optimization methods and considerations for the cruise phase of fixed-wing unmanned aircraft.

The basis of the optimization methods presented in this chapter relies on the trade-off between energy consumption and distance covered. To illustrate; when flying an A-to-B mission with a fixed distance, then the optimization goal considered in this study is to minimize the energy consumption during the execution of this mission. It is therefore required to express the aircraft's energy consumption as a function of distance covered. The required power (P_r in Watts) of propeller-driven aircraft is found through [5]:

$$P_r = D\,v_a = \sqrt{\frac{2\,W^3}{\rho_\infty\,S}\frac{C_D^2}{C_L^3}} \tag{6.4}$$

Where W is the aircraft weight in [N], C_L and C_D are the aircraft's aerodynamic lift and drag coefficients respectively, ρ_∞ is the air density in [kg/m^3], and S is the aircraft's effective wing surface in [m^2].

The aircraft's in-flight performance can be optimized for different mission scenarios. The best range airspeed is found by flying at the airspeed where the energy consumption per travelled distance is minimized. Considering that:

$$v_a = \sqrt{\frac{2}{\rho_\infty}\left(\frac{W}{S}\right)\frac{1}{C_L}} \tag{6.5}$$

Then, when substituting equation (6.5) in (6.4) we find:

$$\left(\frac{P_r}{v_a}\right) = W\left(\frac{C_D}{C_L}\right) \tag{6.6}$$

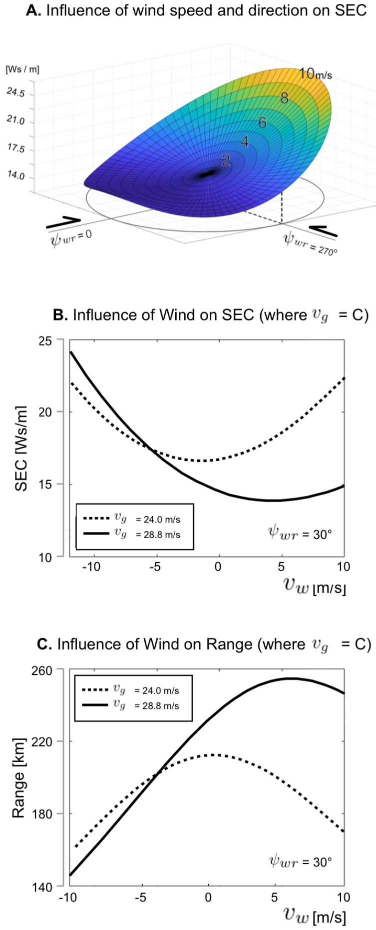
This expression shows that the condition for maximum range occurs at the airspeed where C_L/C_D is maximized. However, as this expression relates travelled distance solely to airspeed rather than ground track speed, this does not necessarily hold true in the presence of en-route winds.

The specific energy consumption *SEC* is defined as the consumed energy per distance travelled. This can be expressed in unit Newtons (N), Joule per meter (J/m), or alternatively Watt-second per meter (Ws/m). Here the latter is chosen since manufacturers of batteries often express the energy capacity in Watt-hour. By plotting the specific energy consumption (obtained from equation 6.6) as a function of v_w and ψ_{wr} (obtained from equation 6.1 and 6.4), the effects of wind on the in-flight performance can be visualized. Here ψ_{wr} is the wind direction relative to the aircraft's course.

In Figure 6.2A such a plot is illustrated which holds valid for the aerodynamic model of the P31016 unmanned aircraft, flying at a ground velocity of 28.8 meters per second and an altitude of 1500 meters under ISA conditions. The P31016 is the unmanned platform used in the path planning scenario, which is specified further in section 6.4.2.

Figure 6.2B shows the performance penalty of the presence of wind at the commanded ground speed of 24.0 and 28.8 meters per second, for the arbitrarily chosen ψ_{wr} of 30 degrees. This figure illustrates that for one given wind speed and direction the maximum range may be obtained by changing the commanded ground speed accordingly. Note that without the presence of wind flying at an airspeed of 24.0 meters per second requires less power per unit *time* compared to flying at 28.8 meters per second. However, as this figure illustrates the energy consumption per unit *length* is found to be lower when flying at 28.8 meters per second. This balance changes depending on the present wind. This is further demonstrated in 6.2C where the resulting obtainable in-flight range is illustrated for both ground speeds.

Path planning optimization algorithms that are set up so that the cost-function optimization considers the energy consumption as a function of covered ground distance in the presence of wind will inherently optimize the commanded airspeed to give the best range. In other cases where the cost function algorithm is set up to command the desired airspeed independently of ground speed, methods such as described by [57] can be applied. In [57] it is suggested that the best-range airspeed can be approximated through:


A. Influence of wind speed and direction on SEC

B. Influence of Wind on SEC (where v_g = C)

C. Influence of Wind on Range (where v_g = C)

Figure 6.2: A: Energy consumption of the P31016 per distance travelled as a function of wind speed, and the wind direction relative to the aircraft's course ψ_{wr}. B: SEC for different wind speed components, valid for ψ_{wr} of 30 degrees with fixed ground speeds. C: Flight range for different wind speed components, valid for ψ_{wr} of 30 degrees with fixed ground speeds. ($v_w > 0$ is a tail wind, while $v_w < 0$ is a head wind)

$$m_{br} = \left[\frac{2\, m_{br} \, \pm \left(\frac{v_{wp}}{v_{md}} \right)}{2\, m_{br} \, \pm \, 3 \left(\frac{v_{wp}}{v_{md}} \right)} \right]^{\frac{1}{4}} \tag{6.7}$$

Here v_{md} is the minimum drag airspeed, v_{wp} is the wind speed along the commanded heading of the aircraft. By solving for m_{br} the ratio between the best-range airspeed in the presence of wind, and the airspeed that gives the best-range without the presence of wind can be found. The symbol $\pm$ indicates a head- or tailwind, where positive values are considered a tailwind.

Missions that require the longest flight endurance, such as observation missions, ought to optimize the airspeed so that the energy consumption per unit time is minimized. Observing equation (6.4) it becomes clear that when the air density, aircraft weight, and wing surface are constant, the total energy consumption becomes a sole function of C_L and C_D. Since C_D and C_L are functions of v_a, the minimum power consumption, and thus the maximum endurance, is found at the v_a where C_L^2/C_D^3 is minimized. Note that the presence of wind does not change the optimum value for v_a to achieve the maximum endurance.

Similarly, path planning optimization algorithms where the cost-function considers the energy consumption as a function of time will inherently optimize the airspeed to obtain the best flight time.

6.4 Path Planning

In this section the path planning solution with the inclusion of the horizontal wind maps is presented. The results are shown after describing the optimization problem formulation, the parameters of the aircraft used for this simulation and how the wind map was obtained to perform the wind interpolation.

6.4.1 Optimization Problem Formulation

An area north of Trondheim, Norway, was chosen for this study. The objective of the optimization is to fly from A to B while using as little energy as possible, while taking the wind into consideration. To achieve this the mission waypoints and the airspeed along the path are optimized using the Particle Swarm Optimization technique, through methods as described in the Appendix B.

A two-dimensional geometric approach is used in this work, where the optimization variables represent a set of airspeed inputs V and waypoints of the path W, with x (North) and y (East) positions in the NED reference frame. The altitude was chosen to be 1500 meters.

As the positions of the origin $[x_s, y_s]$, destination $[x_t, y_t]$ and wind vectors $[v_w, \psi_w]$ are given in latitude and longitude coordinates, a conversion to the NED frame is needed. Besides, to use the result as an input for an autopilot system, it may be required to convert the waypoints to positions expressed in latitude and longitude. To reduce the error coming from the conversion between frames, the coordinates of the origin of the NED frame are defined as the midpoint between the origin and the destination.

As the path is divided into V velocity steps and W waypoints, the algorithm needs to do an interpolation to discretize the path obtained from the W waypoints into a path with V velocity steps. Therefore, the new path will have $V+1$ new interpolated waypoints, where $[x_1, y_1] = [x_s, y_s]$ will be the origin of the mission, $[x_{V+1}, y_{V+1}] = [x_t, y_t]$ will be the destination and the other $V-1$ points are resulted from the interpolation.

The cost function f is set in order to evaluate the energy consumption along the path. Therefore, it adds the energy consumption used to travel each V step through:

$$f = L_{step} \sum_{n=1}^{V} \frac{P_{r_n}}{v_{gs_n}}, \tag{6.8}$$

where P_{r_n} is the required power (equation (6.4)) and v_{gs_n} is the ground speed in meters per second for the nth velocity step. L_{step} is the length of each step of the path, given by:

$$L_{step} = \frac{L}{V} \tag{6.9}$$

where L is the total length of the path:

$$L = \sum_{n=1}^{V} \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \tag{6.10}$$

The domain (x_{min}, x_{max}, y_{min}, y_{max}) has to be defined taking into consideration that the UAS may not deviate too far from the straight line path between the origin and destination. In addition, the airspeed must be optimized within the limits of the aircraft constrains.

To initialize the optimization algorithm, first a straight path from the origin to the destination is generated - with waypoints distributed equally along the path, while the airspeed along the path is set as the airspeed that would give the best range without the presence of wind. This strategy is crucial, as usually the optimal solution will be a deviation from this straight path. If only particles initialized with random positions are used, they might have uncommon waypoints displacement, causing the algorithm to take a long time to find an optimal solution or to get stuck in a local minimum.

The other paths generated for the initialization of the optimization algorithm have the waypoints randomly chosen following the rule that the next waypoint must be closer to the destination than the previous one. The airspeed variables are randomly chosen between the minimum ($\boldsymbol{v}_{a_{min}}$) and maximum ($\boldsymbol{v}_{a_{max}}$) airspeed. Figure 6.3 shows an example of initial guesses for the paths.

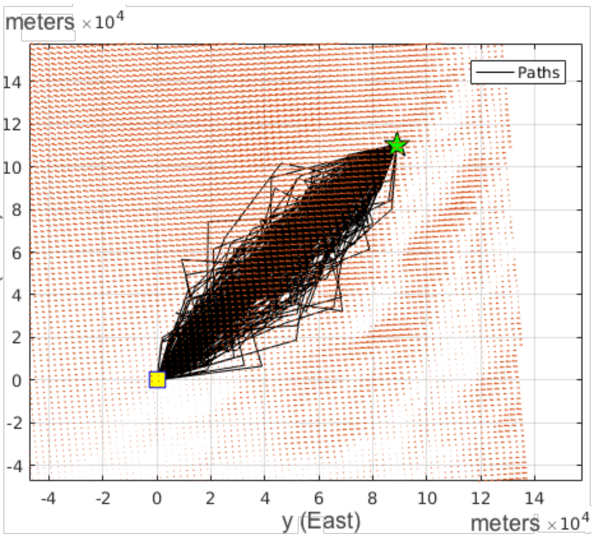


Figure 6.3: 200 paths generated in the initial guess. The yellow square is the origin and the green star is the destination. The red arrows are the wind vectors.

6.4.2 Aircraft Platform

The P31016 (figure 6.4) is a small battery-powered aircraft that is powered by a 6.0 kilowatt brushless motor, and has a battery capacity of 977 Watt-hour. The propulsion efficiency is assumed constant at 50% with an ideal electrical discharge pattern. The aircraft has a wing surface of 0.81 square meters and has a typical mission-ready mass of 17.5 kilograms. Its aerodynamic characteristics were determined through a simplified model of the aircraft in the software tool XFLR5. Here it was found that at an altitude of 1500 meters under ISA conditions the airspeed for maximum range occurs at 28.8 meters per second, while the airspeed for maximum endurance is found at 24.0 meters per second. The aircraft's stall speed with extended flaps is 14 meters per second, while the maximum speed is limited to 38 meters per second.

6.4.3 Wind vector maps

The horizontal wind map used was originally obtained from the Norwegian Meteorological Institute (MET), and provided by the Norwegian Defence Research Establishment (FFI). The wind map contains the amplitude and direction of the

Figure 6.4: P31016 concept battery-powered fixed-wing unmanned aircraft

wind for each point in the grid at a given altitude. The grid has a resolution of approximately 2.5 kilometers, and the position of the points are given as the latitude and the longitude. Figure 6.5 illustrates a section of the wind map used.

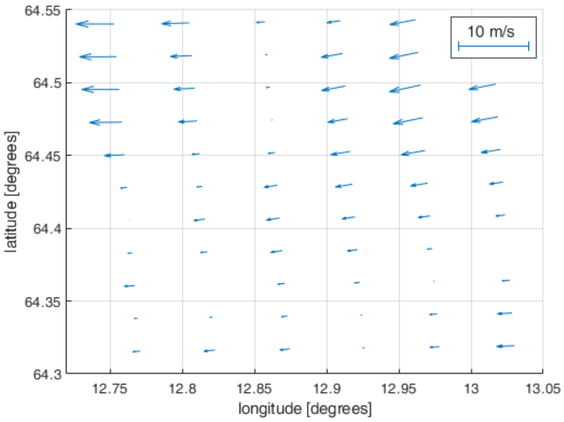


Figure 6.5: Part of the wind map used. The arrows show the amplitude and direction of the wind for each point in the grid. (Source: Maritime Robotics AS)

In order to obtain wind data in between the grid points in the wind map, the wind data needs to be interpolated. Nearest neighbor interpolation could be used for fast interpolation, and could provide sufficient accuracy for a smooth wind field. However, the discontinuity of nearest neighbor interpolation or abrupt wind changes could cause significant errors. In order to improve this, biharmonic spline interpolation [125] is used. Biharmonic spline interpolation has the benefits of creating a smooth surface (has minimum curvature) and passes through each data point. To obtain a sufficiently low computation time, the 16 surrounding grid points (the smallest and second smallest squares, each containing unique grid points, and enclosing the point to be interpolated) are selected as the data points for calculating the interpolation function.

6.4.4 Results

The parameters chosen to set the optimization algorithm are shown in Table 6.1. Figure 6.6 shows the optimized path (black dots) for the mission where the objective is to fly from the yellow square (origin) to the green star (destination). The algorithm has optimized the position of the five waypoints (blue dots) and the airspeed at each V-step. The resulted optimized airspeed is shown in figure 6.7. In this mission the total energy consumption calculated for the straight line path, when flying at the no-wind best-range airspeed of 28.8 meters per second, was 691 Watt-hour. The total energy consumption of the optimized path was 662 Watt-hour. This is a saving of 4.2% of consumed energy. This is despite the fact that the optimized path is 3.6 kilometers longer than the straight path. An overview of the results of the flight time, path length and energy consumption as a comparison between the straight path and the optimized path are shown in Table 6.2.

Table 6.1: List of parameters

Name	Value
Iterations	200
Particles	200
Waypoints (W)	5
V	50
Particle Size	55
Particle velocity constraint	0.1 x Domain
w_{ini}	1.0
w_{fin}	0.1
x_{min}	x_s-$L_{min}/3$
x_{max}	x_t+$L_{min}/3$
y_{min}	y_s-$L_{min}/3$
y_{max}	y_t+$L_{min}/3$
$v_{a_{min}}$	18 m/s
$v_{a_{max}}$	38 m/s

Table 6.2: Simulation results

	Straight Path	Optimized Path
Length	141.8 km	145.4 km
Time	1h 37min	1h 29min
Consumed energy	691 Wh	662 Wh

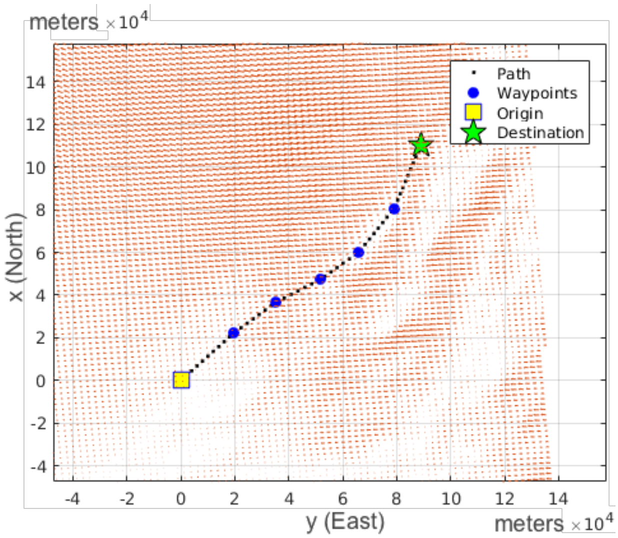


Figure 6.6: Final path - Accounting for en-route winds

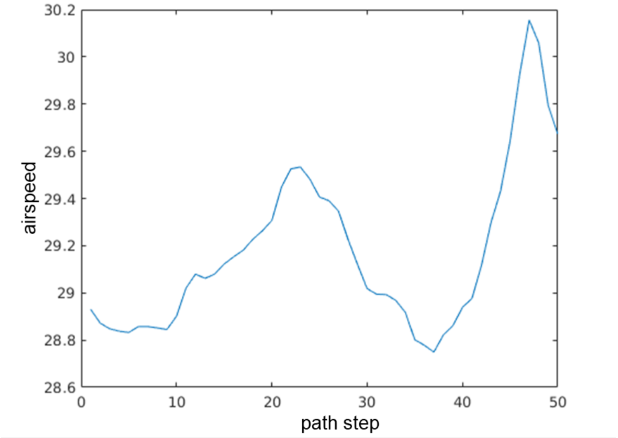


Figure 6.7: Optimized commanded airspeed along the route

6.5 Discussion

The wind maps used in this study represents the wind information obtained through meteorological wind models. As the current wind map is only valid for that moment in time, for longer flights it may prove useful to include forecast wind maps valid for future time windows. Moreover, in-situ path planning may be complemented with real-time wind field estimations through methods such as described in [53] and [89]. In the work described in [15] a real-time field estimation method is described utilizing a moving horizon estimator, which may be used to identify both steady and turbulent wind velocities.

The simulation results presented in this chapter are valid for one chosen scenario. Depending on the local wind field and aerodynamic characteristics of the aircraft the obtainable savings may be higher or lower for other scenarios. It is important to extend this research in the future with more varied scenarios, while having a validated aerodynamic model and propulsion efficiency model of the used aircraft. Finally, the accuracy of the simulation results are as always limited by the accuracy of the input parameters, which to a large extend include the predicted wind field model. As horizontal wind maps do not specify vertical wind components, these effects are not included. It is therefore warranted that in a future research the proposed model is verified through field tests. This is done preferably for a variety of mission scenarios with a different wind field, altitude and terrain.

It is important to complement the proposed method with the ability to include horizontal wind maps of different altitudes, and thereby effectively creating a quasi-three-dimensional (multiple altitudes) wind field. This allows for en-route adjustment of the cruise altitude which has the potential to further increase the obtained flight efficiency.

6.6 Conclusion

In this chapter a method was presented for the inclusion of horizontal wind maps into a path planning optimization algorithm. An aircraft performance model is presented that incorporates the effects of wind on the in-flight energy consumption, in relation to the airspeed and the resulting ground speed. It is demonstrated that in the presence of wind the best-range airspeed is no longer found at the airspeed associated with $(C_L/C_D)_{max}$, thus en-route airspeed optimization is warranted. It is described that when the goal is to maximize the flight range, an optimization algorithm which is set up to optimize the commanded airspeed in order to minimize the energy consumption as a function of ground distance covered, will inherently command the optimal course and airspeed in the presence of wind.

A simulation was performed where a particle swarm optimization method was utilized to determine the wind-optimized flight path, where an in-situ forecast 2D wind field was incorporated. The performed simulation shows that when comparing the wind-optimized flight path to the straight path, the length increased with 3.6 kilometers to a total of 145.4 kilometers. However, the flight time was reduced by eight minutes and the total consumed energy was reduced by 4.2%. These simulation results are valid for the chosen scenario utilizing the P31016 unmanned aircraft. In future work it should be particularly interesting to simulate a more diverse wind field. In addition it is warranted to validate the proposed model through field experiments.

Chapter 7

Long range path planning using an aircraft performance model for battery powered sUAS equipped with icing protection system

Earlier studies demonstrate that en-route atmospheric parameters, such as winds and icing conditions, significantly affect the safety and in-flight performance of unmanned aerial systems. Nowadays, the inclusion of meteorological factors is not a common practice in determining the optimal flight path. This study aims to contribute with a practical method that includes meteorological forecast information in order to obtain the most energy efficient path of a fixed-wing aircraft. The Particle Swarm Optimization based algorithm takes into consideration the aircraft performance, including the effects of en-route winds and the power required for active icing protection systems to mitigate the effects of icing. As a result, the algorithm selects a path that will use the least energy to complete the given mission. In the scenario evaluated with real meteorological data and real aerodynamic parameters, the battery consumption of the optimized path was 52% lower than the standard straight path.

7.1 Introduction

Small Unmanned Aerial Systems (sUAS) have become versatile tools that can be used in a broad spectrum of missions. The rapid growth of the use of sUAS is justified by their endurance, reduced cost, rapid deployment and flexibility. This flexibility is mainly due to the many types of sensors that can be mounted on sUAS, enabling them to be used in many different applications, such as surveillance, reconnaissance, search and rescue, delivery, photogrammetry, inspection, among others. In addition, they offer reduced risk for humans and impact on the environment, when compared to manned aircraft.

A next and necessary step for the continuous evolution of sUAS technology is to enable safe autonomous missions also in adverse weather conditions. For this to be possible, effects of wind and icing on the aircraft performance must be addressed, controlled and taken into consideration by the path planning algorithm to decide if it is worth it to face the adverse weather conditions or to take a detour in order to avoid exposing the sUAS to this.

Scientific literature on path planning of sUAS is abundant. In [66] a comparative analysis of four three dimensional path planning algorithms based on geometry search was done. The algorithms compared were Dijkstra, Floyd, A* and Ant Colony. Run time and path length were the two analyzed aspects. In [26], the author used the Voronoi diagram to produce routes minimizing their detection by radar, while in [160] the Rapidly Exploring Trees (RTTs) were used with a smoothing algorithm based on cubic spiral curves for collision-free path planning. Optimization techniques are also adopted, as Genetic Algorithms [106], MILP [118] and Particle Swarm Optimization [92], where the author used the method to minimize the UAS path's length and danger based on the proximity of threats.

Atmospheric winds usually constitute 20-50% of the airspeed of sUAS [13]. Therefore, it affects the aircraft's in-flight performance significantly. In [123], a sophisticated method was described where Model Predictive Control (MPC) was employed for path planning optimization including the effects of uniform wind. In [2], the author used Markov Decision Process to optimize the unmanned aerial vehicle's path, integrating the uncertainty of the wind field into the wind model. The goal of the algorithm was to minimize the energy consumption and time-to-goal. A similar approach was chosen in [77], where the Ant Colony Optimization (ACO) technique was used to optimize the path by minimizing the travel time considering the effects of an uniform wind.

Most of the works about path planning of sUAS that takes the wind into consideration use an uniform wind distribution. This information is often used in a simplified model when calculating the effects of the wind on the energy consumption. However, in [29], an aircraft performance was successfully included, with the assumption of a constant wind field. In recent literature a nonuniform wind distribution in addition to an aircraft performance model was used. That is the case in [70], where the flight path was optimized so that sUAS was guaranteed to be able to reach a pre-designated safe landing stop. This was done by continuously calculating the remaining range considering the remaining battery capacity in case of an engine failure. In that study a wind map with nonuniform wind distribution was used in the calculations of the maximum range of the sUAS. Also using a nonuniform wind distribution, [72] proposed a two-dimensional optimization algorithm to find the path between two points with the minimum energy consumption. By being aware of the wind map valid for a given altitude, it was possible to choose a path where the wind was used favorably for energy savings for that flight level.

One of the most important meteorological constraints for UAS mission planning is atmospheric icing. This hazard is also called in-cloud icing and occurs when an airframe travels through a cloud containing supercooled liquid droplets. When these droplets collide with the airframe they freeze and result in surface icing that grow over time into ice horns that can significantly alter the wing shape. Even small ice accretions have been shown to be able to decrease the aerodynamic performance of a wing dramatically [21] [97].

The icing hazard is a well-researched topic for general aviation, but little attention has been given to this topic until the recent years for UAS – although the issue has already been identified during the 1990s [129]. UAS icing is in many ways similar to icing on large aircrafts, but also exhibits significant differences when it comes to flight velocities, airframe size, mission profiles, and weight restrictions. In particular, the small UAS typically operate at Reynolds numbers an order of magnitude lower compared to general aviation which causes differences in the flow regime [138].

Modeling of icing effects on UAS have shown that icing results in a degradation of aerodynamic performance. Ice accretions on the leading edge of the lifting surfaces can decrease lift, increase drag, and initiate earlier stall [139]. The degree of the degradations seems strongly linked to the prevailing meteorological conditions. In addition, icing has also shown to have detrimental effects on static and dynamic stability. In summary, icing is a severe hazard, especially for small UAS, and it is common practice to avoid flying in icing at all costs.

An icing protection system (IPS) can be used to mitigate this restriction of the flight envelope. In the scope of this work an electro-thermal system designed at the Norwegian University of Science and Technology will be investigated [133] [62]. This system consists of heating zones on the leading edge of the lifting surfaces that are activated when the aircraft enters an icing cloud. The IPS can run in two different modes. In anti-icing mode, the system will continuously heat the leading edge to inhibit the build-up of any ice. In de-icing mode, the systems operates in a cyclic way, allowing for the accumulation of a small amount of ice over a time of 90 s, followed by the removal of the ice by activating the heating zones for 30 s. Typically, the de-icing mode will require lower power requirements compared to anti-icing, but will also results in performance degradation during the ice accumulation cycles [50].

As weather conditions often varies for geographic location and altitude, it is important that the path planning algorithm is able to allow altitude changes in during flight. Consequently, the terrain profile must be taken into consideration and treated as an obstacle by the algorithm. This was previously implemented by [119], where the PSO and Parallel GA optimization techniques were compared when used to find the best trajectory by minimizing a cost function based on the path length and average altitude, including a penalization in the cases when the path has parts under the terrain.

Electric batteries have variable potential according to the remaining capacity. [145] presented a simple model for open-circuit potential determination. With this model, it is possible to calculate the battery potential with respect to the current being drawn. Lately, [44] derived the model equations to calculate the rate of discharge for a constant-power.

In this study, a path planning algorithm is proposed to find an optimal path between a chosen origin and destination allowing both changes in course and altitude. This optimization is performed by an unique algorithm that simultaneously process several factors, some of which are novel and others that are normally individually studied by the literature. These factors include: the icing protection system usage, which is a very novel solution that enables sUAS to fly under icing conditions; horizontal wind, that is a major issue on sUAS operations and has only recently been studied; terrain profile, that is a fundamental factor that has already been included in many studies; the aircraft performance model, which brings more realistic and accurate calculations of the propulsion required power according to the aircraft platform and environmental parameters; and battery discharge properties, which is a relevant factor as sUAS are typically powered by electric batteries and the discharge rates vary according to the remaining capacity. Therefore, this work contributes to the field by proposing a tool that can be used to plan the sUAS mission and to evaluate the different possible scenarios, in order to assist the decision making. In addition, to demonstrate its applicability, this work also brings the analysis of the proposed solution for a mission scenario using real sUAS platform parameters and real terrain and weather data.

7.2 Aircraft performance model

In this chapter the aircraft performance model is presented with all the equations that are needed for the calculation of the required power to propel the aircraft in given atmospheric conditions and for a desired maneuver.

7.2.1 Pressure

The pressure (p in [Pa]) is calculated from the aircraft altitude by using the barometric formula with subscript 0, that is valid from sea level up to 11000 m of altitude:

$$p(h) = p_0 \left[\frac{T_0}{T_0 + L_0(h - h_0)} \right]^{\frac{g_0 M}{R L_0}} \tag{7.1}$$

where h is the altitude in [m], p_0 is the standard pressure at sea level of 101325 Pa, T_0 is the standard temperature at sea level of 288.15 K, L_0 is the standard temperature lapse rate for subscript 0 of -0.0065 K/m, h_0 is the altitude at sea level of 0 m, R is the molar gas constant of 8.314472 Jmol^{-1}K^{-1}, M is the molar mass of Earth's air of 0.0289644 kg/mol and g_0 is the gravitational acceleration at sea level of 9.80665 m/s^2.

The values of the constants are taken from the International Standard Atmosphere (ISA) mean sea level conditions [109].

7.2.2 Air Density

The density of air (ρ in [kgm^{-3}]) is an atmospheric property which significantly affects the aerodynamic forces.

To calculate the air density, the ideal gas law is used:

$$\rho(p,T) = \frac{p}{R_d T}, \tag{7.2}$$

where p is the pressure in [Pa] given by Eq. 7.1, T is the air temperature in [K] and R_d is the specific gas constant for dry air of 287.058 Jkg^{-1}K^{-1}.

7.2.3 Power Required

Assuming that the lift (Fig. 7.1) is high enough to compensate its opposite weight component to keep the aircraft in the air, it is necessary to provide a power high enough to: overcome the drag force and the weight's component which is tangent to the aircraft's trajectory; and to move the aircraft forward in its trajectory with an excess thrust. Therefore, the required propulsive power is given by multiplying the required thrust by the desired airspeed:

$$P_{req}(T_{req}, v_a) = T_{req} v_a \tag{7.3}$$

where P_{req} is the required propulsive power in [W] given by Eq. 7.3, v_a is the airspeed in [m/s] and T_{req} is the required thrust in [N], which is given by:

$$T_{req}(D,\theta) = D + W sin(\theta) \tag{7.4}$$

where D is the drag force in [N] given by Eq. 7.5, W is the aircraft weight in [N] and θ is the climb angle in [rad].

Hence, when the aircraft is cruising (θ is equal to zero), this results in $sin(\theta)$ being equal to zero. In this case, the weight is normal to the drag force and tangent to the lift.

As the drag force is dependent on the body's size (e.g. the wing surface), the air density and the airspeed, the equation for drag force D is derived by dimensional

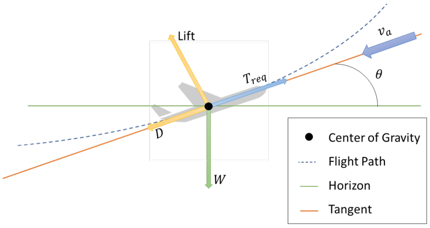


Figure 7.1: 2-D representation of an aircraft in a straight flight.

analysis following the Buckingham's π-Theorem:

$$D(\rho, v_a, C_D) = 0.5\rho v_a^2 S C_D \tag{7.5}$$

where ρ is the air density in [kgm^{-3}] given by Eq. 7.2, v_a is the airspeed in [m/s], S is the wing surface area in [m^2] and C_D is the drag coefficient, given by Eq. 7.8.

For an aircraft equipped with propellers, the engine's required power (P_{shaft}) is obtained dividing the propulsive required power (P_{req} in W, given by Eq. 7.3) by the propeller efficiency (η_p):

$$P_{shaft}(P_{req}) = \frac{P_{req}}{\eta_p}. \tag{7.6}$$

From the descent slope which no propulsion power is required, the aircraft's motor is assumed to be completely shut off and the on-board systems, except for the icing protection systems, are assumed to use insignificant amounts of energy. Therefore, the energy consumption in this case is assumed to be equal to the energy consumption of the icing protection requirements. This is possible for an electric powered aircraft that does not need to keep an engine running during the entire mission. However, the maximum descent angle needs to be chosen so that sufficient lift is provided for airspeed values that are in the range of predefined accepted values of the desired airspeed.

Aerodynamic Coefficients

To be able to calculate the drag force, which characterizes the power required to propel the sUAS, it is necessary to first calculate the drag and lift coefficients (C_D and C_L respectively).

In common A-to-B missions the aircraft is expected to primarily be flying in a horizontal straight flight, and performs a limited amount of turns. These turns depend on the path optimization, however the turns denote on a relatively small part of the entire path. Therefore, the effects of turns (circling flights) are not considered in the following calculations. This holds valid for "A-to-B" missions, and not for other mission types, such as loitering.

In addition, and with respect to the mission profile, the aircraft is assumed to follow a steady motion flight path, trust angle is zero, and the angle of attack is small, typically ranging between -4 and 10 deg.

The lift coefficient for straight flight is given by [53] as:

$$C_L(\theta, \rho, v_a) = \frac{2W\cos(\theta)}{\rho S v_a^2}, \tag{7.7}$$

where v_a is the airspeed in [m/s], W is the aircraft weight in [N], θ is the climb angle in [rad], ρ is the air density in [kgm^{-3}] given by Eq. 7.2 and S is the wing surface area in [m^2].

In this study, the drag coefficient (C_D) as a function the lift coefficient (C_L) was derived by a curve fitting process that aims to find a polynomial equation that represents the true drag polar, typically acquired from wind tunnel experiments or Computational Fluid Dynamics (CFD) simulations. To derive a valid polynomial equation, it is necessary to first define the range of the lift coefficient where the equation will be valid. This domain can be calculated by finding the lowest and highest lift coefficient ($C_{L_{min}}$ and $C_{L_{max}}$), respectively, for the mission and aircraft constrains, such as minimum and maximum accepted airspeed (v_a), minimum and maximum accepted climb angle (θ) and minimum and maximum air density (ρ). C_D is therefore a function of C_L:

$$C_D(C_L) = f(C_L), \tag{7.8}$$

where f is the fitted function.

7.2.4 Ground speed

The airspeed is the speed of the aircraft with relation to the mass of air in which it is flying. In this research, the airspeed is considered tangent to the aircraft's trajectory. Therefore, when the aircraft climbs or descends, it is possible to calculate the projection of the airspeed on the horizontal axis (Fig. 7.2), with the assumption of the absence of vertical wind, by:

$$v_h(v_{h_x}, v_{h_y}) = \sqrt{v_{h_x}^2 + v_{h_y}^2}, \tag{7.9}$$

with x and y components of the horizontal airspeed (v_h) given by:

$$\begin{aligned} v_{h_y}(v_a, \psi, \theta) &= v_a \cos(\psi)\cos(\theta), \\ v_{h_x}(v_a, \psi, \theta) &= v_a \sin(\psi)\cos(\theta), \end{aligned} \tag{7.10}$$

where v_a is the airspeed in [m/s], ψ is the heading in [rad] (Eq. 7.12) and θ is the climb angle in [rad].

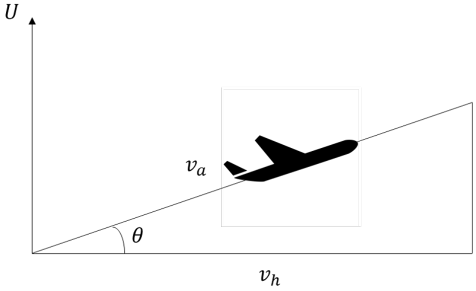


Figure 7.2: Representation of side view of the aircraft.

The presence of wind affects the aircraft's travelled trajectory (Fig. 7.3). The travelled trajectory is subject to the aircraft's ground speed (v_{gs} in [m/s]), which is the aircraft's speed relative to the ground and calculated by:

$$v_{gs}(v_{h_x}, v_{h_y}, v_{wind_x}, v_{wind_y}) = \sqrt{(v_{h_x} + v_{wind_x})^2 + (v_{h_y} + v_{wind_y})^2}, \tag{7.11}$$

where v_h is the horizontal airspeed in [m/s] and v_{wind} the wind speed in [m/s].

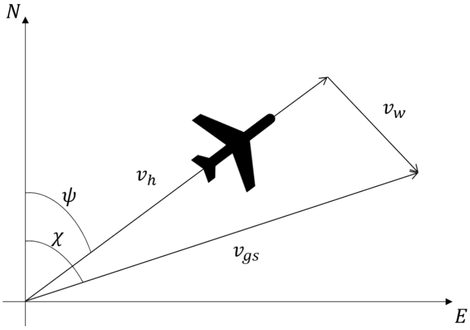


Figure 7.3: Wind triangle.

The heading (ψ) is the direction where the aircraft is pointing to. It is given by:

$$\psi(\chi, v_{wind}, v_a, \psi_{wind}) = \chi - \arcsin\left(\frac{v_{wind}}{v_a \sin(\psi_{wind} - \chi)}\right), \tag{7.12}$$

where the course (χ in [rad]) is the travelled direction relative to the ground, with the wind speed (v_{wind} in [m/s]) given by:

$$v_{wind}(v_{wind_x}, v_{wind_y}) = \sqrt{v_{wind_x}^2 + v_{wind_y}^2}, \tag{7.13}$$

and with the wind heading (ψ_{wind} in [rad]) given by:

$$\psi_{wind}(v_{wind_x}, v_{wind_y}) = \text{arctan2}(v_{wind_x}, v_{wind_y}). \tag{7.14}$$

7.3 Battery performance model

Modern electric batteries have become dominant power sources within sUAS, mainly because of their simplicity, and relatively high peak power output. Common battery types, such as lithium-based cells, are rechargeable and durable, which makes them suitable for sUAS operations.

Electric batteries' energy potential changes according to the remaining capacity. [53] modelled the battery potential (V_{oc} in [V]) based on [145] as (Fig. 7.4):

$$V_{oc}(C, V_o) = V_o - \left(\frac{\kappa C_{cut}}{C_{cut} - C}\right) + Ae^{-BC}, \tag{7.15}$$

where C_{cut} is the capacity discharged at cut-off in [Ah], C is the capacity discharged in [Ah], $A = V_{full} - V_{exp}$ and $B = 3/C_{exp}$ where V_{full} is the fully charged potential in [V]. Additionally, V_{exp} is the potential at the end of the exponential range in [V], and C_{exp} is the capacity discharged at the end of the exponential range in [Ah], with the Polarization Voltage (κ in [V]):

$$\kappa = \frac{(V_{full} - V_{nom} + A(e^{-BC_{nom}} - 1))(C_{cut} - C_{nom})}{C_{nom}}, \tag{7.16}$$

where V_{nom} is the potential at the end of the nominal range in [V], C_{nom} is the capacity discharged at the end of the nominal range in [Ah], and with battery constant potential (V_o in [V]):

$$V_o(I_{eff}, V_{oc}) = V_{full} + \kappa + (R_C I_{eff}) - A, \tag{7.17}$$

where R_C is the internal resistance in [Ohms] and I_{eff} is the effective discharge current in [A].

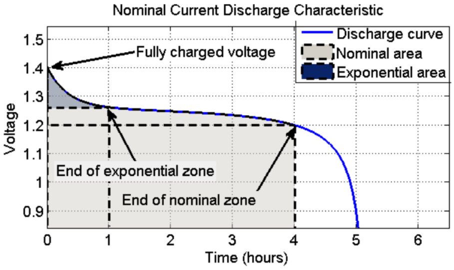


Figure 7.4: Battery discharge curve. (Source: [145])

In this study, the power is considered constant during the discretization step, and, therefore, the effective discharge current is the variable to be calculated. As a result, the Trembley's equations were manipulated to accommodate obtaining the effective discharge current for a given power. From Ohm's law, the effective current (I_{eff} in [A]) is given by:

$$I_{eff}(V_{oc}) = \frac{P_{eff}}{V_{oc}}, \tag{7.18}$$

with the potential (V_{oc} in [V]) being obtained by solving the nonlinear equation:

$$V_{oc}^{n+1} - \left(V_{full} + \kappa - A - \frac{\kappa C_{cut}}{C_{cut} - C} + Ae^{-BC} \right) V_{oc}^{n} - R_C I_{rated}^{1-n} P_{eff}^{n} = 0, \tag{7.19}$$

where n is the battery-specific Peukert's constant and I_{rated} is the maximum battery rated current in A.

Note that to obtain the potential (V_{oc}), it is necessary to solve the nonlinear Eq. 7.19. The valid solution will be in the range from the cut-off potential (V_{cut}) to the fully charged potential (V_{full}).

7.4 Meteorological and elevation data

This work aims to allow sUAS operations in adverse weather conditions. Therefore, meteorological forecast data needs to be considered. This data is used in the calculation of the total aircraft energy consumption, as it affects the aircraft's in-flight performance. Additionally, the meteorological conditions define when the icing protection systems are to be used, and how much power is required to mitigate the adverse effects of aircraft icing. Finally, the elevation data is of importance as the path planning algorithm optimizes the sUAS' altitude, and therefore it is vital to ensure a minimum terrain clearance in the aircraft's planned path.

7.4.1 Meteorological parameters

In Table 7.1 the downloaded parameters are shown. The wind and air temperature parameters are implemented directly in the form that they were supplied in. Other parameters were modified due to unit compatibility for usage in the calculation of other parameters, as described in the following sub-sections.

Table 7.1: List of downloaded parameters.

Parameter	Description	Units
v_{wind_x}	Meridional wind in x direction	m/s
v_{wind_y}	Meridional wind in y direction	m/s
T	Air temperature	K
q	Specific humidity	kg/kg
LWC	Atmospheric cloud condensed water content or Liquid Water Content	kg/kg

Relative Humidity

The specific humidity parameter can be downloaded from the meteorological service. However, in this work, the parameter used in the calculations is not the specific humidity but the relative humidity. This is because the aircraft is assumed to be in icing conditions and turn on the icing protection system when the temperature is below 0 deg C and the relative humidity is over 0.99. Therefore, the relative humidity (H) needs to be calculated and it is given by [46]:

$$H(e_a, e_{sat}) = \frac{e_a}{e_{sat}} \tag{7.20}$$

with the vapour pressure (e_a in [Pa]):

$$e_a(p, q) = \frac{qp}{0.622 + 0.378q} \tag{7.21}$$

where q is the specific humidity, p is the pressure in [Pa] given by Eq. 7.1 and with the saturated water vapour pressure (e_{sat} in [Pa]):

$$e_{sat}(T) = 10^{\frac{0.7859+0.03477(T-273.16)}{(1+0.00412(T-273.16))}} + 2 \tag{7.22}$$

where T is the temperature in [K].

LWC and MVD

The "mass fraction of cloud condensed water in air" can be also referred as "liquid water content (LWC)". In the icing protection system regression model, the LWC is one of the input parameters to estimate how much power is required by the system. The regression model uses the LWC concentration in [gm^{-3}] but the downloaded parameter is the LWC mixing ratio in [kg/kg]. Therefore, to convert LWC mixing ratio (LWC_m in [kg/kg]) to LWC concentration (LWC_c in [gm^{-3}]), the gas law for dry air is used:

$$LWC_c(p, LWC_m, T) = \frac{LWC_m p}{R_d T} \times 10^3, \tag{7.23}$$

where T is the temperature in [K], R_d is the specific gas constant for dry air of 287.058 Jkg^{-1}K^{-1}. and p is the pressure in [Pa] given by Eq. 7.1.

The Water Droplet Median Volume Diameter (MVD in [μm]) is another parameter used to calculate the power required by the icing protection system. It is approximated by following [143] and given by:

$$MVD(\lambda) = \frac{3.672 + \mu}{\lambda}, \tag{7.24}$$

with the shape parameter (μ) given by:

$$\mu = min\left(\frac{1000}{N_c} + 2, 15\right), \tag{7.25}$$

where N_c is the pre-specified droplet number of 100 cm^{-3} and with:

$$\lambda(LWC_c) = \left[\frac{\pi}{6}\frac{\rho_w N_c}{LWC_c}\frac{\Gamma(\mu+4)}{\Gamma(\mu+1)}\right]^{\frac{1}{3}}, \tag{7.26}$$

where Γ is the gamma function, ρ_w is the density of water of 1 gm^{-3} and N_c is equal to 100x10^{-6} m^{-3}.

Meteorological data download

The Norwegian Meteorological Institute hosts a webapp called THREDDS Data Server, where it is possible to have access to weather forecasts of several meteorological parameters. One of the services is the MetCoOp Ensemble Prediction System (MEPS) [102], from where the parameters used in this work were downloaded. This service provides data for the Scandinavian region with horizontal resolution of 2.5 km and from around 0.00986 to 0.99851 atm pressure levels (that can be converted to altitude) divided into 65 not equally spaced values. In the MEPS service, raw and post processed data are available for 10 ensemble members (set of forecast simulations) and for up to 66 hours of forecast. The models are run every 6 hours (00,06,12,18 UTC) and the first data file (00) is the most complete one and the only file containing all the necessary parameters for the development of this work. Therefore, the 00 file was downloaded for the esemble member 0 (mbr0) and the data from the forecast time slot 0 was used in the simulations. The time slot 0 reflects the instant information of the chosen date/time while the other time slots are hourly forecast.

The files are available in the Network Common Data Form (NetCFD) format and each file is up to 200 GB. However, it is possible to select which parts to download by using the Open-source Project for a Network Data Access Protocol (OPeNDAP). Therefore, the selected parameters can be downloaded only for the region of interest and for the desired pressure levels (altitudes).

7.4.2 Elevation data

The elevation data was downloaded from the Norwegian national website for map data (geonorge.no). Geonorge provides a catalog with a wide variety of map products, including elevation maps. These elevation maps are in the form of Digital Terrain Model (DTM) or Digital Surface Model (DSM) and can be visualized in the website or downloaded via WCS or WMS services. In this work, the DTM was used, which is available with 1 m and 15 m of resolution for the regions correspondent to UTM32, UTM33 and UTM35.

Elevation data download

To download the data with the WCS service, the web browser can be used as the WCS client. Therefore, the data is requested via HTTP through URL parameters. The commands to be used are: GetCapabilities; DescribeCoverage; and GetCoverage. The first one returns all service-level metadata and a brief description, the second one returns the full description and the third one returns the data itself. The URL parameters varies according to the product and are usually described in the information obtained by the GetCapabilities and DescribeCoverage commands.

7.5 Path Planning

The goal of this work is to find an optimum three dimensions path minimizing the energy consumption of a long range sUAS flight from an origin to a destination in adverse weather conditions. To achieve this an optimization technique is used to minimize a given cost function.

7.5.1 Optimization technique

In this study the Particle Swarm Optimization (PSO) [35] technique is used to minimize the cost function and therefore find the optimum path. PSO is a meta-heuristic optimization method where the particles (solutions) are updated every iteration based on the best global and local solutions. In this study the standard PSO was used with a modification to reduce the maximum absolute particle velocity by an ϵ factor. This was implemented to keep the search more local, and thereby avoiding too large movements in the solution domain per iteration, as it may be expected that the optimum solution is relatively close to the straight line path.

7.5.2 Optimization algorithm

The algorithm's block diagram is shown in Fig. 7.5. The orange blocks scenario-based input parameters, such as the meteorological and elevation data, the origin and destination, the number of control inputs and the PSO parameters. The other boxes outside the blue box are part of the pre-processing phase when the model is created and the initial solutions are generated.

The blue box contains the optimization loop. First the candidate solutions are evaluated with respect to the terrain. If part of the path is under terrain, the solution is discarded ($cost = \infty$). If not, the optimization will evaluate the icing conditions for each discretization step i.

If icing conditions are present in the step i ($H_i > 99\%$ and $T_i < 0 degC$), the deice and anti-ice required power are calculated (P_{deice_i} and $P_{anti-ice_i}$, respectively). For the deicing operations the engine's required power is calculated with an updated drag coefficient ($C^{*}_{D_i}$), which constitutes the average icing penalty, and therefore an updated required propulsive power $P^{*}_{shaft_i}$. For the anti-ice operations the aircraft's wings are kept clear from icing. Therefore the engine's required power is the same as without ice (P_{shaft_i}). However, the anti-icing system does require thermal energy. In this study the anti-icing system uses the main battery as power source (i.e. does not have a separate power source), and therefore induces a performance penalty during usage. The total power required by the deice and anti-ice systems, including the respective engine's required power, are compared and the solution that requires the least total power is chosen. If there are no icing conditions present the total required propulsion power remains unchanged (P_{shaft_i}).

The next step is to calculate the battery energy consumption in the step i taking into consideration the battery model and how much battery capacity is left. Finally, the total battery energy consumption is calculated by summing the battery energy consumption times the flight time of all steps. The total battery energy consumption is, therefore, used to update the particles' position in the domain. The new solutions are then evaluated. This process repeats for the chosen total number of iterations.

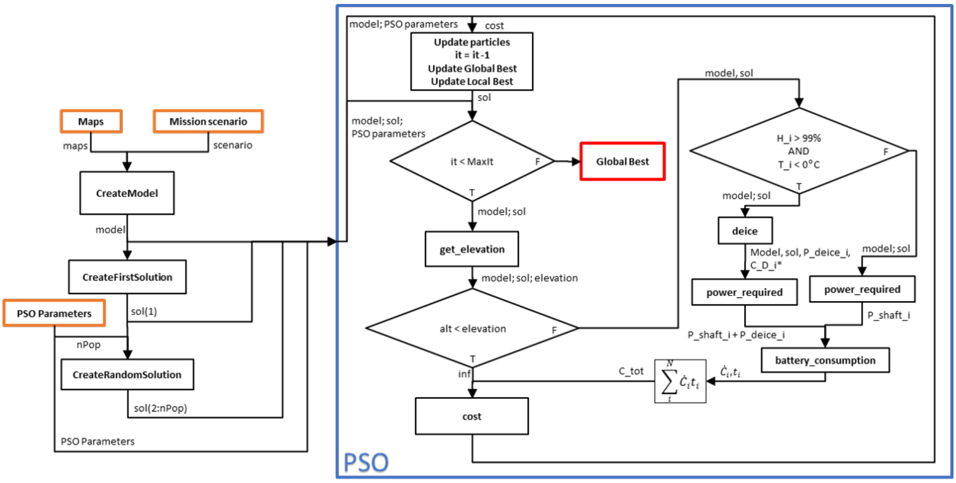


Figure 7.5: Algorithm block diagram.

7.5.3 Cost Function

The aim of the optimization algorithm is to minimize a cost function (Eq. 7.27), which represents the total energy consumption and is calculated by the sum of the battery discharge rate ($\dot{C}$ in [Ah/s]) in each discretization step, multiplied by the time in each discretization step:

$$minimize\, C_{tot}(\dot{\mathbf{C}}, \mathbf{t}) = \sum_{i=1}^{N} \dot{C}_i t_i \tag{7.27}$$

where $\dot{\mathbf{C}}$ and $\mathbf{t}$ are the vectors with all $\dot{C}_i$ and t_i, respectively. C_{tot} is the total discharged capacity in [Ah], i is the index of the discretization step, N is the number of discretization steps in the path, t_i is the time in [s] at the i-th step given by:

$$t_i(L_{step}, v_{gs_i}) = L_{step}/v_{gs_i}, \tag{7.28}$$

where v_{gs_i} is the ground speed in the discretization step i in [m/s] given by Eq. 7.11, with the step's length (L_{step} in [m]) given by:

$$L_{step}(L) = \frac{L}{N}, \tag{7.29}$$

and with the total length of the path (L in [m]) given by:

$$L(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{7.30}$$

where x_i and y_i are east and north positions in the ENU frame and i is the index of the discretization step.

The the rate of discharge ($\dot{C}$ in [Ah/s]) given by:

$$\dot{C}_i(I_{tot_i}) = \frac{I_{tot_i}}{3600}, \tag{7.31}$$

with the total current (I_{tot} in [A]) given by:

$$I_{tot_i}(P_{tot_i}, V_{oc_i}) = \frac{P_{tot_i}}{V_{oc_i}}, \tag{7.32}$$

where V_{oc} is the battery's potential in [V] (Eq. 7.19) and with the total required power (P_{tot} in [W]) given by:

$$P_{tot}(P_{shaft}) = P_{shaft}, \tag{7.33}$$

if there are not icing conditions occurring, or

$$P_{tot}(P^{*}_{shaft}, P_{deice}) = P^{*}_{shaft} + P_{deice}, \tag{7.34}$$

when there are icing conditions occuring, and the deice solution is the one requiring the least power, or

$$P_{tot}(P_{shaft}, P_{anti-ice}) = P_{shaft} + P_{anti-ice} \tag{7.35}$$

if there are icing conditions, while the anti-ice solution requires the least power. Here, P_{shaft} is the engine's required power in [W] (Eq. 7.6), P^{*}_{shaft} is the engine's required power when using the deice solution in [W] (Eq. 7.40), $P_{anti-ice}$ is the anti-ice solution required power in [W] and P_{deice} is the deice solution required power in [W].

Finally, C_i is the total capacity discharged in [Ah] until instant i and given by:

$$C_i(\dot{\mathbf{C}}_i, \mathbf{t}_i) = \sum_{i=0}^{i} \dot{C}_i t_i, \tag{7.36}$$

where $\dot{\mathbf{C}}_i$ is the vector of $\dot{C}$ from $\dot{C}_0$ to $\dot{C}_i$ and $\mathbf{t}_i$ is the vector of t from t_0 to t_i. C_0 is the initial discharged capacity.

Note that when parts of the path are not above the terrain, or if the total energy consumption is higher than the battery's capacity, this candidate solution receives an infinite penalty to ensure it is disregarded as a candidate solution.

7.5.4 Control inputs

The required control inputs are horizontal plane waypoints (x,y), airspeeds (v_a) and climb angles (θ). The number of waypoints (O) and airspeeds/climb angles (K) are chosen by the user when defining the scenario. Note that the airspeed and climb angle changes were chosen to occur at the same time for algorithm simplicity.

7.5.5 Model and Mission Parameters

ENU frame

The ENU frame was chosen as the coordinate system of the optimization algorithm. Therefore, all information in World Geodetic System 1984 (WGS84), which is in the format of latitude, longitude and altitude, must be converted to the ENU frame. Also, the resulting waypoints of the optimized solution must be converted to WGS84 in order to be fed into the sUAS' flight control system.

In this work, when using the ENU frame, the x axis points east, the y axis points north and the z axis points up.

It is also necessary to define the origin (0,0,0) of the ENU frame. As the region around the origin is less affected by the frame conversion error, the origin was chosen to be in the geographical midpoint between origin and destination at sea level.

Domain

The candidate solutions' waypoints are limited to be away from the straight path up to a maximum distance. This maximum distance was defined as one third of the length of the straight path between the origin and destination. Therefore, the optimization algorithm can only find candidate solutions containing waypoints within this domain region.

The boundaries of airspeed (v_a) and climbing angle (θ) must also be defined according to the aircraft platform's constraints. In addition, these boundaries should be fine tuned for values around the expected optimization resulting values, in order to achieve faster convergence.

Discretization strategy

The cost function (Eq. 7.27) is evaluated for each discretization step of the path and the total cost is the sum of the energy consumption in each step. Therefore the number of steps will affect the resolution of the optimization algorithm, and the processing time. The number of discretization steps (N) is defined by the multiplication factor (F) and the number of airspeed and climb angle changes (K):

$$N = KF - 1 \tag{7.37}$$

These parameters are presented for a scenario example in Fig. 7.6. In this example, A is the origin and B is the destination. There is one waypoint between origin and destination ($O = 1$). There are three airspeed and climb angle changes ($K = 3$), and three of multiplication factor ($F = 3$). Therefore there are eight discretization steps (N).

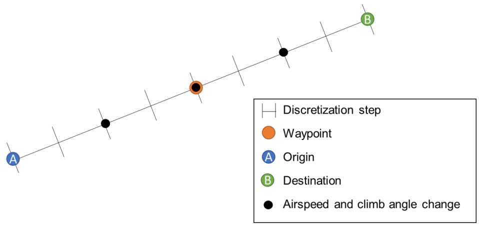


Figure 7.6: Example of a path and its division.

Additionally, the first particle in the PSO algorithm has to be initiated with a candidate solution. A good candidate initial solution for the first particle is a straight path from origin to destination, climbing with constant climb angle to the altitude a few meters above the highest peak, then cruising close to the destination, and finally descending with constant negative climb rate to the destination.

Also, the other particles (candidate solutions) of the population must be initiated. To not distract the optimization algorithm from the region around the first candidate solution, which is expected to contain an optimal solution, the particles are chosen to be variations of the first particle following the exponential probability distribution. Therefore, the values of the set of variables of the other particles are close to the values of the first initial solution set of variables.

7.6 Case study

In this section, the chosen mission case and operational profiles that were evaluated are described, and the aircraft and battery parameters used in the optimization algorithm are explained.

7.6.1 Aircraft platform

The P31016 (Fig. 7.7) is a small battery-powered aircraft that is powered by a 6.0 kilowatt brushless motor. The propulsion efficiency (η_p) is assumed constant at 50% with an ideal electrical discharge pattern. The aircraft has a wing surface (S) of 0.81 m^2 and has a typical mission-ready weight of 171.5 N (W).

Figure 7.7: P31016 concept battery-powered fixed-wing unmanned aircraft

Based on the aircraft flight envelope, the airspeed (v_a) was set ranging from 20 m/s to 30 m/s, and the climb angle (θ) ranging from -10 to 10 degrees. Considering the aircraft performance these limits were chosen to avoid the optimization algorithm explores too high climb angles and airspeed.

The aircraft performance data was generated with the flow solving module FENSAP, which is part of FENSAP-ICE [11]. Three-dimensional CFD simulations were performed on the P31016 (Fig. 7.7) at Reynolds number (Re) of 1.2×10^6 with angles of attack (AOA) corresponding to the set envelope limitations and using a numerical setup described in Table 7.3. The results for drag and lift of the P31016 are presented in Fig. 7.8. The simulations indicate that the flow separation starts from the trailing-edge at AOA of 8 deg. Drag forces increase unproportionally after the onset of stall, whereas lift is decreased as the separation intensifies with higher AOAs.

This data was used to fit the drag polar curve (Fig. 7.9). The curve was fitted for a lift coefficient range calculated based on the aircraft and mission constrains. These constrains are: minimum and maximum airspeed (v_a), minimum and maximum

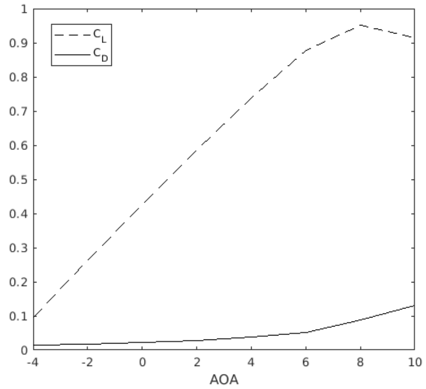


Figure 7.8: AOA (Angle-Of-Attack) vs C_D and C_L from CFD simulations

climb angle (θ) and minimum and maximum air density (ρ). The air density was calculated according to the minimum and maximum expected relative humidity (H), temperature (T) and pressure (p) in the meteorological data. The minimum and maximum resulting lift coefficient for these constrains were 0.3436 and 1.0371 respectively. The fitted curve of the drag polar for this range is given as:

$$C_D(C_L) = 0.1407C_L^2 - 0.07989C_L + 0.02496, \tag{7.38}$$

where C_L is the lift coefficient and C_D is the drag coefficient.

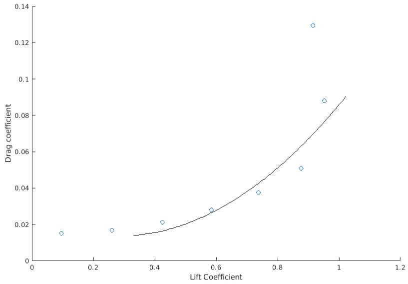


Figure 7.9: Drag polar fitted curve

7.6.2 Icing protection System model

The power requirements for de-icing and anti-icing, as well as the performance penalties during de-icing are generated using numerical simulation methods. Two icing codes are used for this. LEWICE is an icing code that has been developed by NASA over several decades for general aviation [155]. It is a widely validated code [156], but it has been shown that there may be limitations for the application of small UAS [59] [60]. The code is based on a panel-method, that can simulate ice accumulation, anti-icing, and de-icing with very low computational resources. ANSYS FENSAP-ICE is an icing code using modern computational fluid dynamics (CFD) methods [55]. The code is very flexible and has in the past been used for UAS applications [56] but still lacks a dedicated validation for icing at small Reynolds numbers [61].

In this work, the LEWICE is used to generate a model for the anti-icing and de-icing loads, whereas FENSAP-ICE is used for the de-icing performance penalties. The low computational requirements of the panel-method of LEWICE allow to simulate a large number of different meteorological icing conditions in short time, in the order of minutes on a typical desktop computer. The same computations would take several days on a high-performance computing (HPC) cluster with FENSAP-ICE.

A total of 112 different icing cases have been simulated with LEWICE to generate a dataset for anti-icing with LEWICE. The boundary conditions of the meteorological cases are based on the icing envelope of 14 CFR Part 25, App. C [1] used for the airworthiness certification of commercial aircrafts. The simulation cases cover the intermittent maximum (IM) icing and continuous maximum (CM) icing envelope. The range of values for each icing parameter is shown in Table 7.2. Simulations were performed in 2D using the mean aerodynamic chord (MAC = 0.275 m) of the wing. For all simulation it was assumed that only 20% of the leading-edge area of the lifting surfaces was protected (surface temperature of +5 deg C). Runback icing, generated by the refreezing of melted ice from the heated zones, was not included in this study. This was done for reasons of simplification and lack of dedicated studies of runback icing on UAS. Runback icing itself may be a significant source of aerodynamic performance degradation of any IPS [154].

Table 7.2: Range of values for each icing parameter

Parameter	**Range of values**
Airspeed	[20, 30, 40, 50] m/s
Angle of attack AOA	[0] deg
Chord c	[0.275] m
Temperature T_C	[-2, -5, -10, -30] deg C
Median (droplet) volume diameter MVD	[15, 20, 30, 40] μm
Liquid water content LWC concentration	[0.04 ... 2.82] gm^{-3}

The de-icing power requirements have been assumed to be 60% lower than the anti-icing loads. In contrast to the anti-icing, the minimum power requirement for de-icing can not be directly simulated with a steady-state assumption. This means that transient simulations that prescribe a power supply to the leading-edge is required. Such simulations were carried out with LEWICE and confirmed that the aforementioned assumption provides sufficient power for successful de-icing. It should be noted however, that this assumption is a gross simplification, but is deemed sufficient for the purpose of this work.

The 112 simulation cases from LEWICE for the anti-icing and de-icing power requirements ($P_{anti-ice}$ and P_{deice}, respectively) were used to generate linear models that are used for the path planning optimization. Forth order linear regression models were used and have been found to be able to predict the power loads depending on airspeed (v_a in [m/s]), temperature (T_C in [deg C]), Liquid Water Content concentration (LWC_c in gm^{-3}) and Median Volume Diameter (MVD in [μm]) with good accuracy (R^2= 0.977).

The data for the de-icing performance degradation was obtained with FENSAP-ICE in 2D and then extrapolated for the entire aircraft. First, 90 s of ice accretion were simulated with FENSAP-ICE with the numerical parameters specified in Table 7.3. The degradation of lift and drag was then averaged over a full de-icing cycle of 120 s. Again the 14 CFR Part 25, App. C icing envelopes (CM & IM) were applied. In order to reduce the number of simulations, only the cruise velocity of 25 m/s and a single MVD of 20 μm was considered.

Table 7.3: Numerical parameters setup

Parameter	**Setup**
Flow conditions	Steady-state, fully turbulent
Turbulence model	Spalart-Allmaras
Droplet distribution	Monodisperse
Artificial	Second order
Viscosity	Streamline upwind

The aerodynamic degradation occurring during de-icing is presented in Fig. 7.10. A linear model (Eq. 7.39) was selected for the drag (R^2 = 0.81).

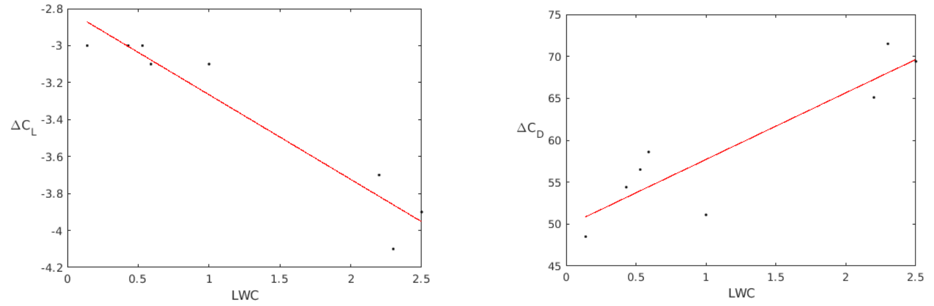
$$C_D^*(C_D, LWC_c) = C_D + C_D(0.0785\, LWC_c + 0.4973). \tag{7.39}$$

Therefore, the required power to propel the aircraft when the de-icing solution is used (P_{pshaft}^* in [W]) needs to be calculated using the degraded drag coefficient (C_D^*):

$$P_{pshaft}^*(\rho, v_a, C_D^*, \theta) = \frac{(0.5\rho v_a^2 S C_D^* + W sin(\theta)) v_a}{\eta_p}. \tag{7.40}$$

Battery parameters

The P31016 is assumed to be equipped with a commercial 10-cells LiPo battery with 26.4 Ah capacity (C_{cut}). Following Tremblay's model, the potential parameters



(a) LWC [gm^{-3}] vs ΔC_L [%]

(b) LWC [gm^{-3}] vs ΔC_D [%]

Figure 7.10: Degradation on lift and drag.

of a 10-cells LiPo battery are approximately: 41.8, 39.67 and 37.67 Ampere-hour of fully charged (V_{full}), end of exponential range (V_{exp}) and end of nominal range (V_{nom}) respectively. The capacity parameters are approximately: 2.64 and 20.4 Ampere-hour of end of exponential range (C_{exp}) and end of nominal range (C_{nom}), respectively. In addition, from the battery's manual it is found that the internal resistance (R_c) is 0.015 Ohms and the maximum rated discharge current (I_{rated}) to be 660 Ampere. The potential curve of this battery with respect to the capacity discharged for 10 Ampere of constant current is shown in Fig. 7.11.

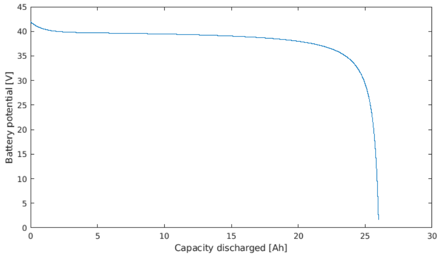


Figure 7.11: Battery potential times capacity discharged

Note that for all cases, the battery was assumed to be fully charged in the beginning of the mission. Therefore, C_0 (the initial capacity discharged of the battery) was assumed to be equal to 0 Ah.

7.6.3 Mission Case

The region of Northern Norway was chosen for the evaluation of the proposed solution. The meteorological and elevation data were obtained for the area of the white rectangle of Fig. 7.13. In this area, one mission case was defined to be investigated and the weather of the date of 20th of January of 2019 was chosen as the reference weather. For this area and date, the parameters of liquid water content concentration (LWC_c in [gm^{-3}]) and temperature in deg C are related as shown in Fig. 7.12 if icing conditions are met.

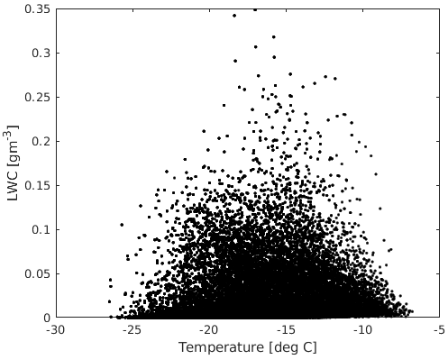


Figure 7.12: LWC_c and temperature distribution.

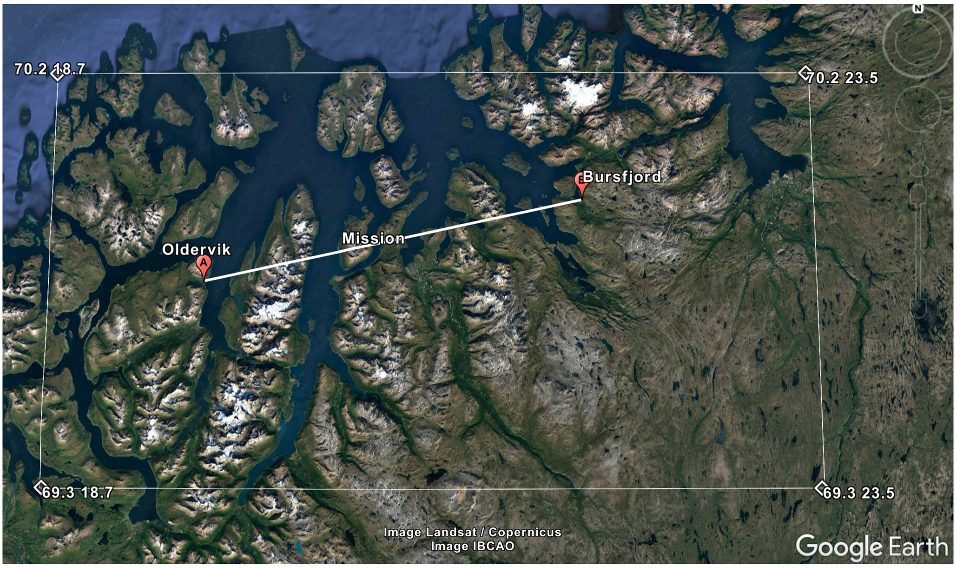


Figure 7.13: Mission case.

Operational Profiles

For the mission case, twelve different operational profiles (OP) were evaluated as described below.

Note that all the operational profiles start at 250 m of altitude, regardless of the altitude of the take off spot. Therefore, it is assumed that before starting the autopilot, the aircraft will be taken by the pilot to 250 m of altitude. Also, when reaching the destination, the aircraft must be landed by the pilot. Take off and landing maneuvers are not considered in this work.

The straight paths are assumed to have constant airspeed of 28 m/s, which is around the value of the best cruise airspeed for the P31016.

- OP 01: Horizontal straight path between origin and destination, climbing to a few meters above the highest peak, flying at constant altitude until close to the destination, then descending until the destination. Evaluated under no icing conditions.
- OP 02: Optimized path without considering icing conditions. Evaluated under no icing conditions.
- OP 03: Optimized path considering icing conditions, using deice or anti-ice (best option) when needed. Evaluated under no icing conditions.
- OP 04: Optimized path considering icing conditions, using only anti-ice when needed. Evaluated under no icing conditions.
- OP 05: Horizontal straight path between origin and destination, climbing to a few meters above the highest peak, flying at constant altitude until close to the destination, then descending until the destination. Evaluated under icing conditions, using deice or anti-ice (best option) when needed.
- OP 06: Optimized path without considering icing conditions. Evaluated under icing conditions, using deice or anti-ice (best option) when needed.
- OP 07: Optimized path considering icing conditions, using deice or anti-ice (best option) when needed. Evaluated under icing conditions, using deice or anti-ice (best option) when needed.
- OP 08: Optimized path without considering icing conditions. Evaluated under icing conditions, using only anti-ice when needed.
- OP 09: Horizontal straight path between origin and destination, climbing to a few meters above the highest peak, flying at constant altitude until close to the destination, then descending until the destination. Evaluated under icing conditions, using only anti-ice when needed.
- OP 10: Optimized path considering icing conditions, using only anti-ice when needed. Evaluated under icing conditions, using deice or anti-ice (best option) when needed.
- OP 11: Optimized path considering icing conditions, using only anti-ice when needed. Evaluated under icing conditions, using only anti-ice when needed.

- OP 12: Optimized path considering icing conditions, using deice or anti-ice (best option) when needed. Evaluated under icing conditions, using only anti-ice when needed.

7.7 Results

Table 7.4 show the results for the mission case, where the sUAS flies from Oldervik to Bursfjord. In icing conditions, the operational profile seven has the lowest battery energy consumption (7.05 Ah), as expected. Compared to the operational profile one, which consumes 14.82 Ah of battery, it brings a reduction of 52.43 % on the battery energy consumption. Also, in this mission case, if only the anti-ice is used and the sUAS is flying straight (OP 09), the battery energy consumption is equal to 20.54 Ah, almost three times more than the optimized path that both deice and anti-ice are available. This is due to the absence of path optimization and to the fact that the anti-ice system requires more power.

In addition, if the path is optimized without taking the ice into consideration, the expected battery energy consumption is of 6.28 Ah (OP 02). However, if the sUAS actually experiences icing conditions during this flight, the battery energy consumption is of 10.74 Ah (OP 06), against 7.05 Ah when the path is optimized taking into consideration the weather forecast (OP 07). Therefore, this shows the importance of using the weather information to optimize the path.

Table 7.4: Mission case operational profiles results

	Straight	Opt. without ice	Opt. with anti-ice	Opt. with deice	Eval. with anti-ice	Eval. with deice	Battery Cons. [Ah]	Length [km]	Time [min]	Length in ice [km]	Time in ice [min]
OP 01	x						8.08	91.47	44.68	0.00	0.00
OP 02		x					6.28	91.82	45.59	0.00	0.00
OP 03			x	x			6.52	97.49	49.80	0.00	0.00
OP 04			x				6.65	94.84	50.36	0.00	0.00
OP 05	x				x	x	14.82	91.47	44.68	49.39	23.32
OP 06		x			x	x	10.74	91.82	45.59	34.89	16.27
OP 07			x	x	x	x	7.05	97.49	49.80	3.90	1.96
OP 08		x			x		14.32	91.82	45.59	34.89	16.27
OP 09	x				x		20.54	91.47	44.68	49.39	23.32
OP 10			x		x	x	7.09	94.84	50.36	3.79	1.71
OP 11			x		x		7.46	94.84	50.36	3.79	1.71
OP 12			x	x	x		7.48	97.49	49.80	3.90	1.96

All optimized paths were longer than the straight path. Also, the flight time was slightly longer in all cases. This is due to the fact the optimization takes the wind into consideration so it is able to change the path to find a better wind profile and/or to change the airspeed accordingly. Therefore, the flight duration is longer but the battery energy consumption is lower.

Figure 7.14 shows the straight path (OP 05) and Fig. 7.15 the optimized path (OP 07) of the mission case. The paths are represented by the black line and the blue dots at ground level shows the projection of the paths in the latitude-longitude planes. The red lines represent the terrain. It is possible to notice that in the optimization (Fig. 7.15), the path is optimized both vertically and sideways, so that the ice is avoided when possible by placing it under or above the icing clouds (light blue dots). Also, when close to the destination, the descent maneuver is started as soon as possible, so energy savings are enhanced.

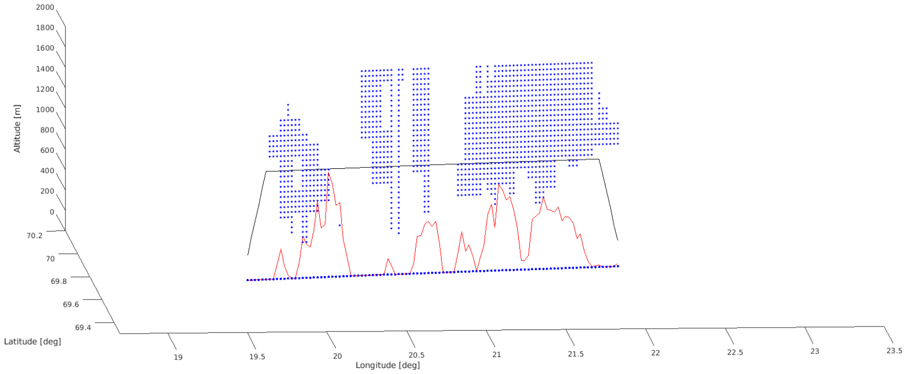


Figure 7.14: Straight path.

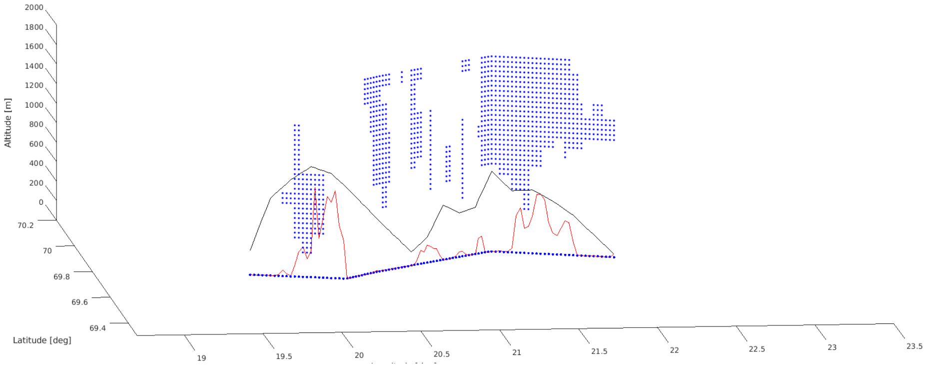


Figure 7.15: Optimized path.

The two peaks on the battery consumption (Fig. 7.16) between 5 and 10 minutes and between 35 and 40 minutes are due to the icing conditions. In the first moment that the sUAS is flying under icing conditions, the power required by the deice system is 477 W and the increase on the power required to propel the aircraft is 184 W, totalizing 661 W. The increase on the propulsion required power is due to the drag coefficient penalty brought by the deice system. If the anti-ice solution was used, where there is no penalty on the drag, the required power would be around 1150 W. Therefore, the deice solution requires less power in total (deice system plus propulsion power). The predominance of the deice solution over the anti-ice will repeat in almost every case investigated in this work. This is due to the mission constrains and to the fact that, according to the deice and anti-ice regression models used in this work, the anti-ice will only have an advantage in maneuvers with high drag.

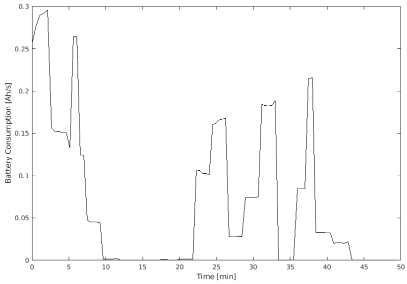


Figure 7.16: Battery Consumption.

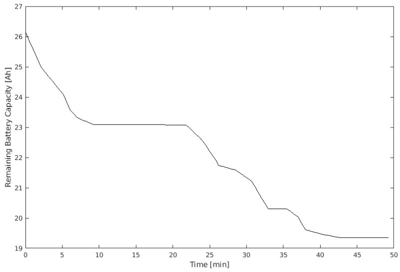


Figure 7.17: Battery Discharged.

Finally, Fig. 7.18 shows the optimized airspeed along the path (OP 07). It is possible to notice that the airspeed is kept around the known best cruise airspeed of the aircraft, which is around 28 m/s.

It should be noted that several simplifications have been applied to some of the simulation input of this study regarding the icing protection system and icing effects that may have a significant influence on the overall results:

- No runback icing effects
- Simplified de-icing load calculation
- Simplified simulation of the aerodynamic degradation during de-icing

These simplifications were introduced in order to limit the amount of expensive computational simulations. Since this work is focussing mostly on the path planning

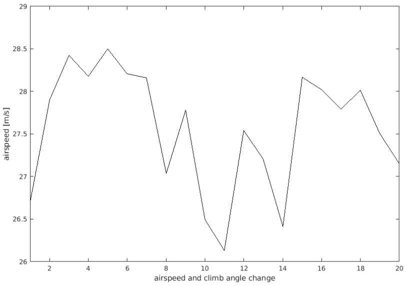


Figure 7.18: Airspeed of optimized path.

method, these simplifications were considered sufficient for this study. For future work, a greater level of detail can easily be included to the required input data.

7.8 Conclusion

This chapter presented a path planning algorithm for small UAS equipped with icing protection systems. An aircraft performance model was used to calculate the power required to propel the aircraft. A battery model was also included in the calculations to give a more precise battery consumption. The goal of the algorithm was to find an optimum path that uses the least energy, taking into consideration the atmospheric parameters, such as wind, liquid water content, relative humidity and temperature of a given time. Climb/decent angles, airspeed and waypoints were the optimization variables. The investigated mission case was to fly between two towns in Northern Norway in a given date of the winter season. Twelve operational profiles were compared and the proposed solution, that takes the icing conditions into consideration when optimizing the path, achieved 52% of battery savings when compared to the standard straight path, proving itself to be a very useful solution for path planning in icing conditions. In addition, it was verified that, for the sUAS used in this work, the deice solution will require less power to protect the sUAS from icing in the majority of situations, compared to the anti-ice solution.

Part III

Chapter 8

Concluding Remarks and Recommendation for Future Work

In this thesis, real-time and off-line solutions for UAV path planning were described and discussed. Each chapter describes the improvements of the previous solution and a new application.

In part one, in the applications where the mission included a moving maritime vessel, it was shown that the inclusion of the ship's kinematic model in the Model Predictive Control (MPC) optimal control problem (OCP) significantly increases the system performance. Regarding the UAV kinematic model, the use of a coordinated turn model, considering wind and where the control inputs corresponded to the autopilot's commanded controls, allowed the UAV to execute more agile maneuvers and, consequently, achieve better performance. Therefore, this solution was superior than the previous one where a simplified model was used and the calculated waypoints fed to the autopilot control unit.

In addition, the Particle Swarm Optimization algorithm was implemented with CUDA C programming language, which benefits from parallel computing and, therefore, achieves faster convergence. This results in improved performance because MPC or optimization parameters such as number of horizon steps or number of iterations can be increased. Also, integrating this algorithm to the DUNE framework, which was used to make the interface between the MPC algorithm and the Ardupilot autopilot control unit, was proven to be a suitable solution that simplifies the implementation of UAV path planning applications.

Software-In-The-Loop (SITL) simulations were proven of fundamental importance when evaluating real-time path planning solutions. Challenges such as communication delays, actuator limitations and interfacing may significantly impact the results. Therefore, numerical simulations without proper SITL simulations (where the UAV flight dynamics is also considered) may achieve results that are far from what is expected from a real flight. SITL simulations have the objective to bring the results closer to reality, allowing a fairer evaluation.

Regarding the off-line path planning solutions for long range missions discussed in the second part of this thesis, it was shown that planning the UAV mission by taking the wind effects into consideration can significantly reduce the energy consumption and, consequently, increase the mission endurance. In the path planning algorithm, the aircraft aerodynamics model should be included in the cost-function in order to achieve accurate estimations of performance. Also, to estimate the expected range of the UAV it is fundamental to consider the wind effects.

For UAVs equipped with electro-thermal icing protection systems, path planning can significantly reduce energy consumption. Also, finding quasi-three-dimensional (multiple altitudes) paths, where the altitude can change along the path, is beneficial to avoid operating under severe icing conditions or to mitigate the effects of wind. In this kind of application, it is important to limit the optimization search domain to the area above the ground by taking elevation maps into consideration.

The main activities of recommended future work are grouped as the following.

- **Optimization algorithm improvements**: Particle Swarm Optimization (PSO) was the technique used for the optimization of both real-time and off-line path planning. The algorithms were implemented with the feature of parallel processing, in order to achieve faster convergence. However, it is probably possible to reduce even more the computational time by optimizing the algorithm implementation. Improvements such as structuring better the memory sharing and using pointers and references are desirable, as well as the conversion of floating point numbers into integers. The simplification or approximation of computational expensive mathematical expressions, such as trigonometric functions, also have the potential to significantly reduce the processing time. One straightforward example is the Euclidean distance that is used many times by all the solutions presented in this thesis and can be replaced by taxicab distance, which results in a faster algorithm. Also, modifications on the structure of the PSO as the ones proposed in recent literature should be experimented. In addition, other optimization techniques may be tested and their performance compared with PSO.
- **Collision avoidance**: In the real-time application, where multiple UAVs are expected to be employed, a basic collision avoidance constraint was implemented. This solution was an abrupt constraint that discards the solution if the distance between the UAVs is smaller than the desired safe distance. It may be beneficial to improve the collision avoidance constraint implementing an algorithm that applies a smoother penalty which is proportional to the distance between UAVs, such as potential field-based collision avoidance solutions.

- **DUNE's maneuver integration**: The real-time algorithms use DUNE to get navigation state, suervisory control, and to send the control inputs to the Ardupilot autopilot control unit. The off-line solutions assume that the waypoints, airspeed and altitude will be manually fed to the mission planner software. Both solutions work but are not ideal. To increase the operational safety, it is necessary to integrate the solutions to DUNE's plan management. Therefore, the systems will be part of DUNE's maneuvers list. This allows a centralized control management and fast response in case of failures, as well as correct logging and automatic start of contingency plans in case the UAV has an unexpected behavior.
- **Hardware-In-The-Loop simulations**: When the embedded software is ready and tested with Software-In-The-Loop simulations, it is considered important to test it in the UAV's on-board hardware before a flight in order to anticipate and correct any problems that would result in accidents or malfunctions. The hardware should be configured and integrated as for a flight. An on-board computer, such as the NVidia Jetson, should have the DUNE software installed with the path planning algorithm. Also, additional sensors, such as cameras, should be connected. This on-board computer should be connected to the Ardupilot. Therefore, the network connections (and corresponding delays), interfacing, as well as sensors and actuators, are prepared and can be tested. Also, different scenarios should be simulated and investigated in order to evaluate the robustness of the solutions proposed.
- **Field tests**: Finally, after the Hardware-In-The-Loop simulations, the system should be ready to be tested in field tests. In these field tests, the payload, such as on-board computer and sensors, should be mounted on experimental aircraft platforms. The performance of the system should be evaluated and eventual corrections applied with the objective to achieve a reliable system that can be used in future research and applications.

Appendices

Appendix A

Object Classification in Thermal Images using Convolutional Neural Networks for Search and Rescue Missions with Unmanned Aerial Systems

In recent years, the use of Unmanned Aerial Systems (UAS) has become commonplace in a wide variety of tasks due to their relatively low cost and ease of operation. In this paper, we explore the use of UAS in maritime Search And Rescue (SAR) missions by using experimental data to detect and classify objects at the sea surface. The objects are chosen as common objects present in maritime SAR missions: a boat, a pallet, a human, and a buoy. The data consists of thermal images and a Gaussian Mixture Model (GMM) is used to discriminate foreground objects from the background. Then, bounding boxes containing the object are defined and used to train a Convolutional Neural Network (CNN). The CNN achieves the average accuracy of 92.5% when evaluating a testing dataset.

A.1 Introduction

Maritime Search and Rescue (SAR) operations are usually based on the drifting trajectory, which is influenced by the water streams and winds. In such operations, it is common to estimate the drift by deploying buoys with GPS sensors to transmit their positions [73]. Since changes in the environment at the search region are common, the search parameters might change many times during the mission, leading to the necessity of the reconfiguration of the mission itself. The search is usually performed using manned aircraft and vessels and is limited by the costs, the availability of human resources, and the mental and perception limitations of the human operators. All these limitations impose that a method for automatic classification of objects would be beneficial to the SAR mission as an additional assistance to the operators, due to its ability to process multiple inputs at higher speeds and with an invariable reliability rate, as it is not subject to exhaustion.

The use of Unmanned Aerial Systems (UAS) has grown rapidly, especially because of their high endurance, reduced cost, rapid deployment and flexibility. They also offers reduced risk for humans and impact on the environment compared to manned aircraft. Therefore, intelligent autonomous UAS equipped with image recognition capabilities to classify vessels, wrecks, people and objects pose are well suited tools to assist maritime SAR operations.

In these missions, it is fundamental to identifying key objects in aerial images using the techniques of object detection, classification and tracking. However, it might be more challenging to solve these classic computer vision problems when using UAS, especially because of real-time requirements and relatively fixed view angles. Moreover, running computationally intensive algorithms, such as image processing algorithms and deep neural networks with many filters and convolution layers, can be an additional challenge due to UAS power consumption limitations, and space and weight constraint for embedded hardware.

Leira et al. [91] used thermal camera images captured by UAS to detect, classify, and track objects at the sea. The solution presented arises as a useful tool for SAR operations. The object detection algorithm used relies on static filter parameters and thresholds, which are determined manually a posteriori. The classifier used is based on the object area, the average object thermal radiation, and its general shape. However, there are a number of scenarios where this classification would be challenging, e.g., when motion blur is present or when the object is moving across an image with varying sensor intensity, which can be caused by an uneven scene radiance or sensor noise. Therefore, a deep learning algorithm could be a more effective tool for the object classification, since it can handle variations on the images affected by environmental changes, as long as these effects are widely present in the dataset.

Convolutional Neural Networks (CNN) are the state-of-the-art deep learning tools for classification of images. Using convolution and pooling layers, it is possible to efficiently extract the most relevant features of the images. Some works were done with CNN and UAS, as in [128], where bounding boxes of images captured by a camera mounted on a UAS at a high altitude were classified in real-time into four classes: building, ground, tree and road. In [83], ground animals were detected using CNN in aerial images captured by a camera mounted on a low-cost UAS in Namibia and the step of object detection for bounding boxes prediction was also explored in the work. Sea animals were detected in aerial images in [98], where the bounding boxes were defined by the confidence of each pixel of being the center of a window containing a mammal and then a CNN is used to classify the images. Regarding the use of CNN to classify objects in aerial images taken by a UAS in maritime environments, a work was done by [30], where RGB images were used and bounding boxes were classified into two classes: *boat* or *notboat*. SAR, CNN and UAS are used together in [14], where near real-time object detection was performed by a UAS for avalanche SAR missions. A pre-trained CNN did the object detection and a Support Vector Machine (SVM) was used to classify the proposed human bodies. All of these works were done using datasets of RGB images, but there are also some works using CNN with thermal images, as in [76], to monitor machine health and in [80], to detect pedestrians. However, there were not found works using CNN to classify objects at the sea in aerial thermal imagery and this is particularly important in night time low visibility SAR operations.

In this paper, a CNN is trained to classify boats, buoys, people and pallets in images captured by a thermal camera mounted on a fixed-wing UAS. The foreground objects were detected by modeling the background as a mixture of Gaussian distributions and subtracting the foreground [136]. This method is computationally cheaper than other object proposal methods such as sliding windows [111] or selective search [147] because it is particularly suitable for thermal images at the sea, as there are two modes present in the distribution: the radiance reflected from the sky, and the heat emitted from the sea [19]. Subsequently a window was fitted around the objects and padded to ensure that the full objects were included in the window. One other novelty brought by this study is the use of the estimated observed area as an extra feature in the fully connected layer of the CNN.

A.2 Dataset

The dataset consists of images captured by a thermal camera mounted on a fixed-wing UAS. The thermal camera used is a FLIR Tau2, which provides analogue video data at a 640 × 512 pixels resolution. The lens has a focal length of 19 mm, which produces a 32° × 26° angle of view. The analogue video data is converted to digital using a 16 bit analogue-to-digital converter. In order to create 8 bit images, the 16 bit images are normalized between 0 and 255 for the smallest and largest intensity in the full dataset. The UAS was also equipped with an Inertial Measurement Unit (IMU) and Global Navigation Satellite System (GNSS) unit, in order to find the surface area of the objects in the images (see section A.2.3).

Four different objects were placed in the ocean: a 26 feet boat, an euro pallet, a human wearing an immersion suit, and a buoy with a 60 cm diameter. The objects are chosen as common objects present in maritime SAR missions, where e.g. pallets are a common object to search for when trying to locate fish aggregating devices. The objects can be seen in higher resolution visual light camera images in figure A.1. The human varied between different actions during the experiment: floating horizontally on the surface (creating a large, long surface), swimming (creating a medium sized surface varying in shape), and standing vertically (creating a small surface, down to 20 cm across).

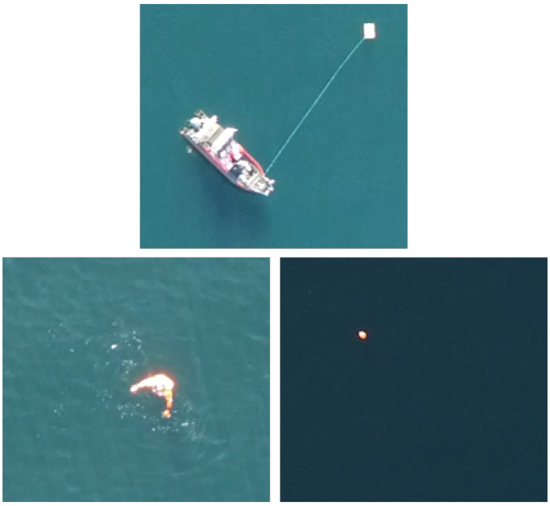


Figure A.1: The different objects present in the scene, as captured by a higher resolution, visual light camera. Top: boat and pallet, bottom left: human, bottom right: buoy. The images were captured at different altitudes.

The total dataset consists of around 22,000 images that were captured during a time span of 50 minutes. The objects were only fully inside in the camera field of view in a limited subset of the full dataset, leading to a smaller number of images used in the CNN.

Various imperfections were present in the images. Several images contain motion blur caused by the dynamics of the UAS. This effect is minor for larger objects such as the boat, however for smaller objects such as a human head sticking out from the water, it can greatly affect the shape, size, and intensity of the object. See figure A.2 for an example of how motion blur changes the object size and dimensions. The pixel intensity is also varying throughout each image, which makes the same object take on intensities between 97 to 110 in an example 8 bit image sequence. This might be caused by noise in the uncooled thermal image sensor, internal camera intensity calibrations, or varying scene radiance. The background varies between 81 and 98 in the same image sequence. See figure A.3 for an average of all images without objects, where the intensity variation can be seen.

In order to find the objects in the images and label them, their boundaries were first found (section A.2.1). The objects were then automatically labeled based on the physical area of the boundary (see section A.2.1 for definition of the physical

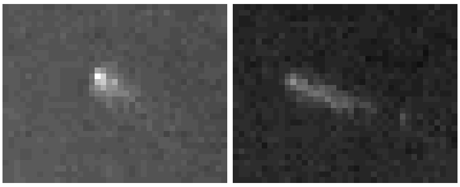


Figure A.2: The same object (human) without motion blur (left) and with motion blur (right). The shape, size, and intensity is greatly affected.

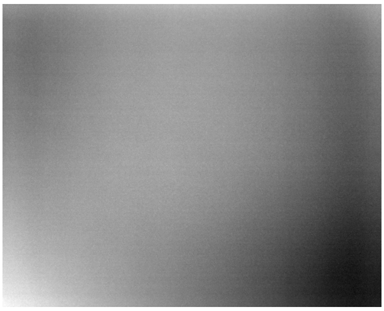


Figure A.3: The mean image without any objects, with its intensity stretched to show the varying image intensity.

area) and finally manually corrected (section A.2.2). The number of labeled objects in the dataset used in the CNN is summarized in table A.1.

Table A.1: Number of labeled objects in the dataset

Boats	620
Pallets	739
Humans	313
Buoys	276

A.2.1 Bounding Boxes

In order to discriminate the foreground objects from the background in the images, the background pixels were modeled using an adaptive background Gaussian Mixture Model (GMM) [136]. The GMM provides robust foreground segmentation and is suitable for thermal images at the sea since there are two modes present in the distribution: the radiance reflected from the sky, and the heat emitted from the sea. It can also model the varying sensor noise, but might fail when the thermal camera is performing sudden noise corrections. The algorithm was implemented using the Background Subtraction Library [132]. A study by Borghgraef et al. [19]

showed that more advanced algorithms, such as ViBe and the behaviour subtraction algorithm, outperformed the GMM for detecting objects at the sea surface in thermal images. However, this was for a static camera at a highly slant angle, which means that the study is not completely applicable to the scenario of this paper. For this application, the GMM is chosen as a good balance between robustness and simplicity. The bounding box was then defined as the smallest box that encloses the boundary of the object. The bounding boxes of all objects were then padded to the size of the largest bounding box found in the dataset. See figure A.4 for a sample boundary and bounding box.

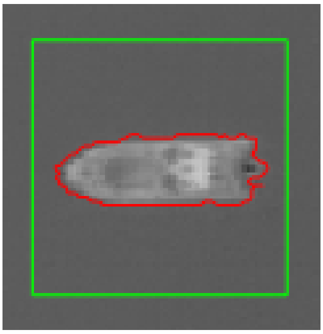


Figure A.4: The border around the extracted foreground object (red), and the bounding box (green).

A.2.2 Labeling

In order to use the extracted foreground objects in the supervised learning algorithm, each object needs to be properly labeled. The objects were first assigned one of three labels based on their observed area in square meters (see section A.2.3) - boat, pallet, or human/buoy. Each label was then manually verified and adjusted if deemed incorrect.

Due to the low ground resolution and their similar dimensions, discriminating humans from buoys was not possible only using the size as a criterion or by looking at individual images due to the varying shapes of the human and other effects, e.g., motion blur. A manual classification was therefore done by analyzing the shape of each object appearing in a sequence of images, taking into consideration that the buoy is completely round while the human has a more elliptical and varying shape. See figures A.5 and A.6 for examples with a boat, a pallet, a human, and a buoy in the images.

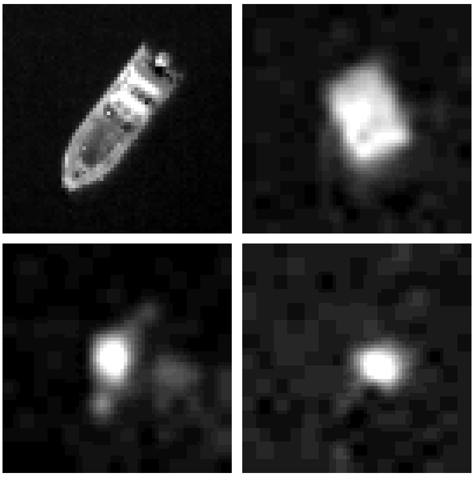


Figure A.5: The different objects which were labeled. Top left: boat, top right: pallet, bottom left: human, bottom right: buoy. The images are scaled to show more detail.

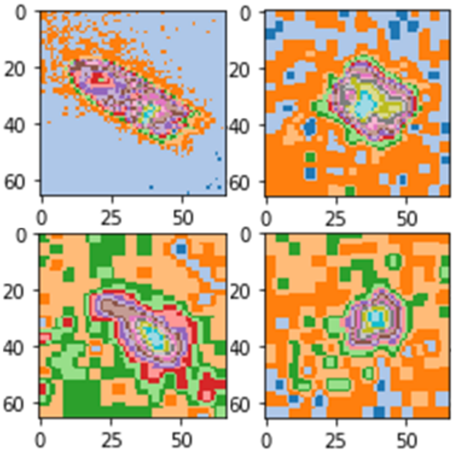


Figure A.6: Objects in a different color map, in order to aid in manually discriminating humans from buoys. Top left: boat, top right: pallet, bottom left: human, bottom right: buoy.

A.2.3 Object Area

The observed area of each object in square meters is used as an extra feature in the fully connected layer of the CNN. The real observed area is defined as the area of the object as seen by the camera, when projected at the plane spanning the North and East axes (NE-plane) at an altitude of zero ($D = 0$). See figure A.7 for a visual description of the observed area of an object.

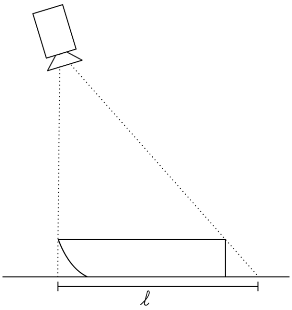


Figure A.7: The observed length, l, of an object. The observed area is the corresponding feature in two dimensions.

The pinhole camera model [114] is used to calculate the observed area of the boundary of each object. First, the observed area of the center pixel within the boundary is calculated, which is then multiplied by the number of pixels within the boundary. In order to perform these calculations, it is necessary to know the roll and pitch angles and altitude of the camera. This data is obtained from the IMU and GNSS data, and is represented in the form of the extrinsic camera matrix. The intrinsic matrix is calculated from the camera specification. No lens distortion is considered – due to the relatively small angle of view of the lens, the distortion will likely be low and not affect the results in a significant way.

The observed area distributions for each object is shown in figure A.8. It can be seen that boats and pallets can be almost completely classified based on their observed area (with minor overlap between pallets and humans), while there are major overlaps between humans and buoys. This is however an artifact of this dataset – in other datasets, buoys and boats can take on a variety of sizes. As previously mentioned, humans can take on a wide variety of sizes due to the different poses.

The real observed area of a buoy with a diameter of 60 cm should be 0.28 m^2. As can be seen in figure A.8, the area is biased towards higher values. One reason for this is that a 60 cm circle can appear in 16 pixels (figure A.9), when the observed area of each pixel is 17.9 cm – which is the case when flying at an altitude of 200 m with no roll or pitch using the camera system used in the experiment performed.

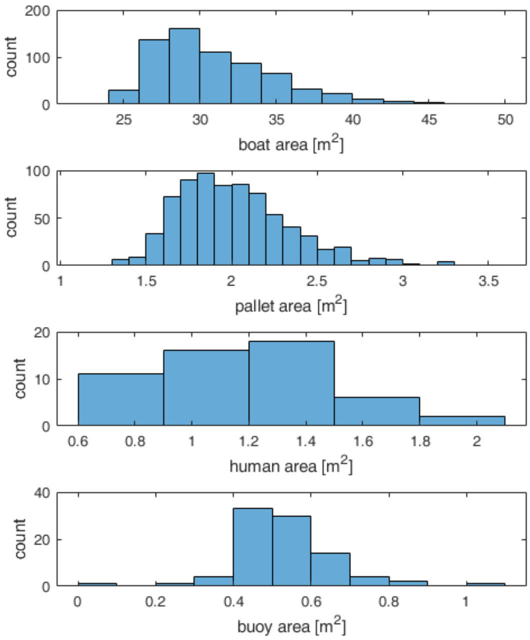


Figure A.8: Distribution of observed areas for the objects present. From top to bottom: boats, pallets, humans, buoys.

This gives an observed area of 0.52 m^2. Additionally, the motion blur causes the object to appear larger than it really is.

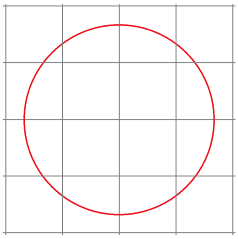


Figure A.9: A buoy with a diameter of 60 cm can appear in 16 pixels, when the observed width and height of each pixel is 17.9 cm.

A.3 Convolutional Neural Network

Traditionally, supervised learning based image analysis combines feature extraction with classical machine learning methods [149]. Convolutional Neural Net-

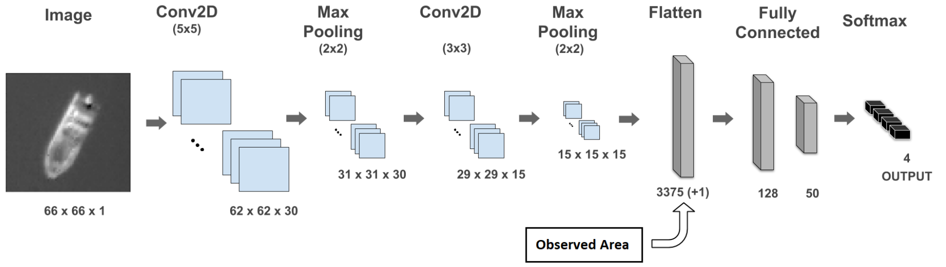


Figure A.10: CNN architecture. If the Estimated Object Area is used, one more element is added in the flatten layer, resulting in an output of 3.376 elements.

work (CNN) is an alternative trend for image classification that has been proven to produce high accuracy in image classification tasks [39] without requiring any task-specific feature engineering [103]. It is considered the most successful machine learning model in recent years [90] and the most eminent method in computer vision [27], in part because it consists of a powerful image feature extractor [16].

A CNN is based on neuroscience research about the processes that mammalian visual cortex uses to recognize images [49]. Typically several basic stages compose a CNN. Each stage consists of concatenation of convolution, normalization, activation (nonlinear), and pooling layers [151]. In this work, two distinct architectures were used. The difference between them was the employment of the observed object area as an input of the fully connected layer.

A.3.1 Architecture

Regarding the architecture (see Figure A.10), the proposed network starts with an input layer containing the image window. This layer is followed by the convolution layer which produces 30 feature maps from filters of size 5×5. The convolutional layer has a set of learnable filters called kernels. By the convolution between one kernel and a chunk of values from the layer, a feature map is generated, which consists of a presence representation of a specific feature in the image. The next layer is a max pooling with a filter of size 2×2, whose purpose is extracting the hierarchical features of the input image [94]. It works by mapping the bigger value from a 2×2 chunk to only one value in the next layer. Pooling helps to make the representation approximately invariant to small translations of the input [49]. This function is also responsible for reducing the width and the height of the feature map. Reducing this dimension, the computational demand is reduced due to the reduction of the number of parameters, which helps to avoid over-fitting. Then, it comes another convolution layer with 15 feature maps of size 3×3, and finally one more max pooling layer composed by 2×2 filters.

The next layer is a flatten layer used to adjust the tensor dimensions to the fully connected layers. At this point, the two architectures become different. One network has the estimated object size as an input and the other one does not. Then, it follows 2 fully connected layers composed respectively by 128 and 50 neurons using the rectifier activation function. The last fully connected layer is used to provide the predicted classification, using the softmax activation function. This is the most common solution for the regulation of the output values within the range from 0 to 1 [93], which assigns a multinomial probability distribution to the output vector [162]. It enhances the discriminative modeling power of the CNN, providing the probability of the input to belong to each possible class, namely: boat, buoys, human or pallet.

A.3.2 Dropout

One technique widely used to improve the performance and avoid the overfitting (which is often a serious problem for a CNN [161]) is the dropout. The term “dropout” refers to dropping out units (hidden and visible) in a neural network during the training phase. By dropping a unit out, it means temporarily removing it from the network in the current epoch, along with all its incoming and outgoing connections [135].

The dropout parameter controlled in this paper was the independent probability of deactivating a neuron. This parameter was tested with 2 values: 0.2 and 0.5.

A.3.3 Cross-validation

In order to evaluate the generalization capacity of the classifiers, it is preferable that the dataset used in the evaluation process is different from the one used during the training process. Typically, the formation of the training and test set is based on non-repetitive sampling techniques, such as the k-fold cross validation method [122]. Cross-validation is a robust statistical technique for estimating the true risk function [4] (or the generalization error), the most important operational performance metric of a trained network [82].

In this paper, the database is divided into five sets of equal size. During each execution of the algorithm, one set is chosen to be excluded in the training phase, which will be the corresponding test set. This process is repeated five times, and the performance metric is inferred for each of the datasets that were left out of the training process. The value of the overall performance metric will be defined by the average of the values obtained for each of the five executions.

A.3.4 Stopping criterion

In the neural network training phase, a stopping criterion has to be used to stop the training of the neural network. To ensure that the training was stopped in a way to provide an appropriate generalization, a validation based early stopping is used in this work [113]. A small part of the training dataset is sorted out to be used as a validation set. At each epoch, the performance index is evaluated for the new training set and for the validation set. When the performance metric of the validation set stops to decrease, i.e., when the training starts to over-fit, the training is stopped.

A.4 Results

After testing the convergence for different parameters, the first CNN was chosen and trained for the 5-folds of images in 8 bit format, without taking into consideration the estimated size of the objects. The maximum number of epochs was 500 and the early stopping was set to stop the training after 50 validation evaluations without improvements. The validation split was 0.18 and dropout was 0.50. After doing 10 executions to get an indicative statistical performance, the average accuracy was 92.0% with 0.50% of standard deviation. This result shows that the configuration of the training algorithm was well set, so that the performances of all executions for all folds were similar.

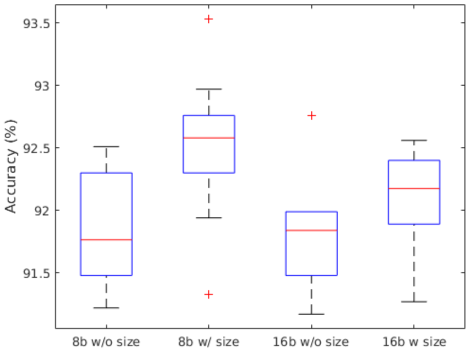


Figure A.11: Accuracy for different configurations.

When using the estimated objects size as an extra input of the fully connected layer, the resulted accuracy is higher as shown in figure A.11 for 8 bit and 16 bit images, achieving 92.5% and 92.1% of accuracy, respectively. Regarding the classification, it is possible to notice that classifying between buoys and humans is a challenge as seen in the Confusion Matrix (table A.2). However, the use of the estimated object size helps the CNN to get better results (table A.3). When looking to the Confusion Matrix for buoys, there are fewer cases when the buoy is classified as a human. There is even a case of a boat being classified as a pallet when the estimated object size was not used (table A.2).

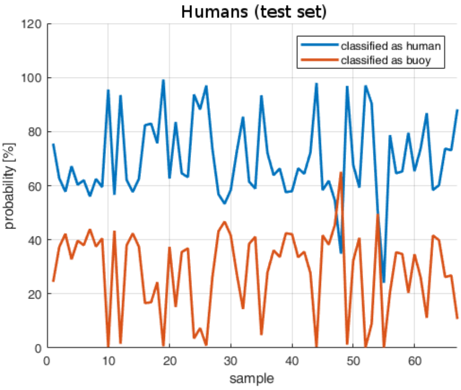


Figure A.12: Probability that a human is either a human (blue) or buoy (red).

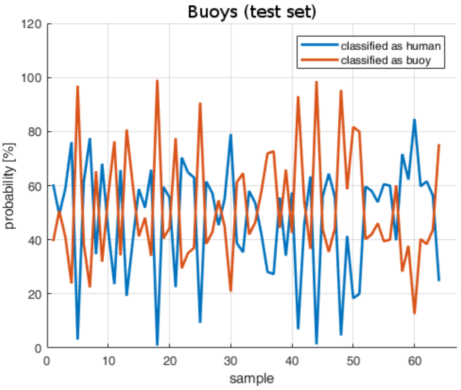


Figure A.13: Probability that a buoy is either a human (blue) or buoy (red).

The ability of the CNN to classify humans vs. buoys is further investigated in figures A.12 and A.13, where the probability of each human and buoy test sample being either a human or a buoy are shown. In the humans samples, it is possible to notice that it is easier for the CNN to differentiate them from buoys. However, when analyzing the classification probabilities of the buoys samples, it shows that it is very challenging to the CNN to decide if it is a buoy or a human.

Regarding the comparison of the performance between 8 and 16 bit images, even with the fact that the 16 bit images have more detail, the accuracy was sightly higher for the configuration with 8 bit images. This might be caused by the reduction of noise when reducing the bit depth, e.g., small intensity variations in the sea surface.

Table A.2: Confusion Matrix for 8 bit images without using object size.

		Predicted			
		Boat	Human	Buoy	Pallet
True	Boat	128	0	0	1
	Human	0	36	18	0
	Buoy	0	15	45	1
	Pallet	0	0	1	145

Table A.3: Confusion Matrix for 8 bit images using object size.

		Predicted			
		Boat	Human	Buoy	Pallet
True	Boat	114	0	0	0
	Human	0	41	13	0
	Buoy	0	7	64	0
	Pallet	0	0	1	150

Images of three vessels from a different dataset (Figure A.14) were used to evaluate the performance of the CNNs.

The observed areas of the vessels were estimated and chosen to be 200 m^2, 25 m^2 and 10 m^2 for the big, medium and small boat, respectively.

When using the CNN where the observed area is not used, all the three samples achieved 100% of probability of being a boat. Regarding the CNN where the observed area is used as an input, the big and medium vessel achieved 100% of probability of being a boat and the small vessel was classified as a pallet.

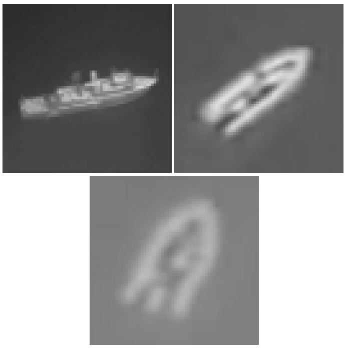


Figure A.14: Images of vessels from an external dataset. Top left: big vessel, top right: medium vessel, bottom: small vessel.

A.5 Discussion

To obtain the images of the objects, the bounding boxes were defined as the smallest box that encloses the boundary of the object. Then, the bounding boxes of all objects were padded to the size of the largest bounding box found for the set of objects of the same class. However, in the SAR mission, it is not possible to know the class of the object a priori, therefore, another strategy need to be used to define the bounding boxes. One solution could be to define a specific number of pixels pad the boundaries of the detected object with, to ensure that the whole object will be inside the bounding box. This number of pixels should be defined by the altitude of the UAS when the image is being captured and also the estimated size of the object. Thus, effects by the distance to the scene would also be mitigated.

The observed area, as well as its appearance in the thermal images, are greatly affected by motion blur. For larger objects this does not pose a major problem, but for objects being just a few pixels in size, the difference can be of major concern. An actively stabilized gimbal and carefully chosen exposure times based on the UAS dynamics could prevent this. Another mitigating solution would be to collect a larger dataset in order to be able to properly classify objects even with motion blur.

The major difficulty of the CNN is to properly distinguish between humans and buoys, which is likely due to the low resolution of the thermal image sensor and relative high altitude, resulting in the objects being represented by very few pixels in the images. In real world maritime SAR missions, however, a buoy being classified as a human would not be a major issue, as the operator would still be notified, and could dismiss the notification from the CNN. Incorrectly classifying a human as a buoy could potentially cause a missed person, but could be solved by lowering the human probability threshold for notifying the operator.

In the dataset used in this work, all boat samples have similar observed areas. Therefore, when evaluating the classification performance for images of vessels from an external dataset, the result was superior when using the CNN where the object area was not considered as an input. However, the generalization power of the CNN containing the observed area can be improved by using a dataset with more samples of boats of different sizes. Also, in general, it is beneficial to have more data, especially at different angles and altitudes.

A.6 Future Work

In the mission carried out to gather the data used for this work, an Electro-Optical (EO) camera was also equipped in the UAS to capture RGB images. However, the thermal and the RGB images were not obtained during the same flight, so it is not possible to use the two images together as inputs of the same CNN.

Therefore, a future step is to develop a CNN to classify the objects in the RGB images, as done by [30]. Then, investigating a method to use both datasets together, for example, trying to use the results of each independent CNN multiplying the probability of each sample to be one of the classes.

For the CNN proposed by this work, the classification of images of vessels obtained in another mission in totally different conditions was evaluated. However, it is important to evaluate the classification for images of humans, buoys and pallets as well. Thus, it would be possible to estimate how well the CNN could perform in a real mission.

Another aspect that requires evaluation is how to improve the discrimination between humans and buoys, especially in the case of buoys, where the calculated probability of a buoy sample being a buoy is very close to the probability of being a human. Examples of this approach would be to use an actively stabilized and sweeping gimbal together with a lens with higher focal length, in order to get a higher ground resolution.

A.7 Conclusion

In this paper, the algorithm for detecting and classifying objects at the sea surface in thermal camera images taken by Unmanned Aerial Systems (UAS) has been discussed. The algorithm uses a Gaussian Mixture Model (GMM) in order to discriminate foreground objects from the background in the images. Then, bounding boxes around the objects are defined and used to train and test a Convolutional Neural Network (CNN). The observed area of the objects was also estimated and used as an input. The CNN was evaluated using the k-fold method with 5 folds and achieved an average of 92.5% of accuracy. Images of vessels from an external dataset were also evaluated and all of them achieved 100% of probability of being a boat when using the CNN where the observed area was not used. The results and the robustness of the CNN algorithm prove it to be a useful tool to assist maritime SAR operations, and be a central part in a future fully autonomous UAS operation in SAR missions.

Appendix B

Particle Swarm Optimization

Particle Swarm Optimization (PSO) [35] is a technique that uses a population of solutions that explores the hyperspace of a problem at a defined speed, which is adjusted according to the best individual historical solution p_{best}, and with the best historical global solution g_{best}. This evaluation is performed by calculating the cost function. Calculating the cost function according to the position of the particle makes it possible to identify whether the new position is better than that previously occupied by the particle. Thus, at each iteration a new velocity, i.e., the movement in the domain space, is adjusted as a function of p_{best} and g_{best}. This is done so that each particle explores the hyperspace optimally, as it takes into consideration the historical performance of the population. This procedure is illustrated in figure B.1. Through this method the movement of each particle is considered to naturally evolve into the optimal (solution) position.

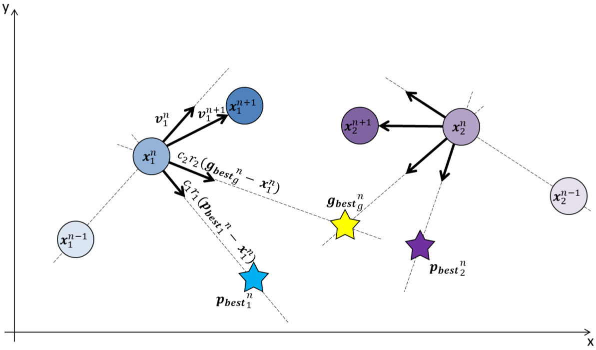


Figure B.1: Behavior of two particles in an arbitrary two-dimensional space

This technique is notable for its simplicity as the behavior of each particle, and therefore the set of presumed solutions, is defined by only two iterative equations.

These determine the position x_i^n and velocity v_i^n of the particle i at time n, resulting in:

$$\begin{aligned} v_i^{(n+1)} = v_i^n + \; & c_1 \; r_1 ({p_{best_i}}^n - x_i^n) \\ & + \; c_2 \; r_2 ({g_{best_g}}^n - x_i^n) \end{aligned} \tag{B.1}$$

$$x_i^{(n+1)} = x_i^n + v_i^{(n+1)} \tag{B.2}$$

where c_1 and c_2 are called “acceleration coefficients”, which are related to the local and global portion, respectively; and with r_1 and r_2 representing the stochastic factor of these accelerations. These are usually chosen as a uniformly distributed random value between 0 and 1.

The PSO algorithm corresponds to the pseudocode shown in Algorithm 1:

Algorithm 1 PSO

```
Initialize a swarm with random positions and velocities
while Stop criteria is not satisfied do
    for Each particle $i$ do
        Calculate the new velocity
        Update the position
        Evaluate the cost function $f(x_i)$
        if $f(x_i) < f(p_{best\,i})$ then
            $p_{best\,i} \leftarrow x_i$
        end if
        if $f(x_i) < f(g_{best\,g})$ then
            $g_{best\,g} \leftarrow x_i$
        end if
    end for
end while
```

Several authors proposed modifications to the basic algorithm. In this study two small modifications proposed by the original creators of the algorithm are adopted; position and velocity boundary constraints as described in [37], and linear inertia weight as described in [36].

The PSO algorithm evolves by updating the particle position for each iteration in relation to the velocity vector. Such updates have stochastic gains, where it is undesirable that the particles move uncontrollably. A particle that has a high velocity in relation to the total domain size, may eventually jump to a distant point inside the domain. This results in the particle no longer performing a minutely search for the optimum. To avoid this problem, the concept of position and velocity constraints was developed.

Another fundamental strategy is to limit the search domain in relation to the optimization problem in question. To prevent the particle from exploring distant regions away from the region that has the optimal solution, or to prevent from bringing solutions outside the problem domain.

Here the basic idea is to avoid for the particle to leave the domain where the optimal solution resides.

The constraints can be implemented through:

$$v_i = \begin{cases} V_{max} & \text{if } v_i > V_{max} \\ -V_{max} & \text{if } v_i < -V_{max} \\ v_i & \text{otherwise} \end{cases} \tag{B.3}$$

The following conditions are added to the algorithm:

$$x_i = \begin{cases} X_{max} & \text{if } x_i > X_{max} \\ X_{min} & \text{if } x_i < X_{min} \\ x_i & \text{otherwise} \end{cases} \tag{B.4}$$

The final modification is to permit a better control of the search domain. The inertia weight, indicated in the following equation as w^n, is applied to the current velocity v_i^n, during the process of calculating the new velocity of the particle:

$$\begin{aligned} v_i^{(n+1)} &= w^n v_i^n + c_1\, r_1 ({p_{best_i}}^n - x_i^n) \\ &\quad + c_2\, r_2 ({g_{best_g}}^n - x_i^n) \end{aligned} \tag{B.5}$$

When a constant value is chosen for the inertia weight, high values imply high velocities, which can make the particle to traverse the entire search domain more quickly; while low values slow down, limiting the search domain of the particle to its neighborhood. Initially, a constant value was proposed for the inertia weight. However, proposals of dynamic values that varied linearly appeared later.

In this specific case the consensus is that initially it is more convenient for the particle to have a global search power, and only afterwards perform a more local exploration. In the linear inertia weight, if N is the maximum number of iterations, nd w_{ini} and w_{fin} are the values of the initial and final inertia weight, the inertia weight for the iteration n is determined by:

$$w^n = (w_{ini} - w_{fin}) \frac{(N-n)}{N} + w_{fin} \tag{B.6}$$

References

[1] F. A. Administration. 14 CFR parts 25 and 29, appendix c, icing design envelopes, 2002. DOT/FAA/AR-00/30.

[2] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith. Wind-energy based path planning for unmanned aerial vehicles using markov decision processes. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 784–789. IEEE, 2013.

[3] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa. LSAR: Multi-UAV collaboration for search and rescue missions. *IEEE Access*, 7:55817–55832, 2019.

[4] S. Amari, N. Murata, K. R. Muller, M. Finke, and H. H. Yang. Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks*, 8(5):985–996, Sep 1997.

[5] J. Anderson. *Aircraft performance and design.* McGraw-Hill international editions: Aerospace science/technology series. WCB/McGraw-Hill, 1999. ISBN 9780070019713.

[6] J. Anderson. *Introduction to Flight.* Anderson series - International edition. McGraw-Hill, 2012. ISBN 9780071086059.

[7] F. A. A. Andrade. Route planning of unmanned aerial vehicles for maritime situational awareness. *Revista de Villegagnon (English Edition)*, 10:146–150, 2015.

[8] F. A. A. Andrade, R. Storvold, and T. A. Johansen. Autonomous UAV surveillance of a ship's path with MPC for maritime situational awareness. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 633–639. IEEE, 2017.

[9] F. A. A. Andrade, C. D. Rodin, A. R. Hovenburg, T. A. Johansen, and R. Storvold. Path planning of multi-UAS communication relay by decentralized MPC. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pages 1–8. IEEE, 2018.

[10] F. A. d. A. Andrade, A. Reinier Hovenburg, L. Netto de Lima, C. Dahlin Rodin, T. A. Johansen, R. Storvold, C. A. Moraes Correia, and D. Barreto Haddad. Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control. *Sensors*, 19(19):4067, 2019.

[11] ANSYS. ANSYS FENSAP-ICE user manual 18.2, 2017.

[12] C. Baraniuk. The crop-spraying drones that go where tractors can't. *BBC News*, 2018. URL `https://www.bbc.com/news/business-45020853`.

[13] R. W. Beard and T. W. McLain. *Small unmanned aircraft: Theory and practice*. Princeton university press, 2012.

[14] M. B. Bejiga, A. Zeggada, and F. Melgani. Convolutional neural networks for near real-time object detection from UAV imagery in avalanche search and rescue operations. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.

[15] S. Benders, A. W. Wenz, and T. A. Johansen. Adaptive path planning for unmanned aircraft using in-flight wind estimation. In *2018 International Conference on Unmanned Aerial Systems*, 2018.

[16] C. Bentes, D. Velotto, and B. Tings. Ship classification in TerraSAR-X images with convolutional neural networks. *IEEE Journal of Oceanic Engineering*, 43(1):258–266, Jan 2018.

[17] J. Berndt. JSBSim: An open source flight dynamics model in c++. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 4923, 2004.

[18] S. C. Boraz. Maritime domain awareness: Myths and realities. *Naval War College Review*, 62(3):137–146, 2009.

[19] A. Borghgraef, O. Barnich, F. Lapierre, M. Van Droogenbroeck, W. Philips, and M. Acheroy. An evaluation of pixel-based methods for the detection of floating objects on the sea surface. *EURASIP Journal on Advances in Signal Processing*, 2010(1), Feb 2010.

[20] J. Braga, R. Praveen Jain, A. Pedro Aguiar, and J. Sousa. Self-triggered time coordinated deployment strategy for multiple relay UAVs to work as a point-to-point communication bridge. In *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2017 Workshop on*, pages 0–0. IEEE, 2017.

[21] M. B. Bragg, A. P. Broeren, and L. A. Blumenthal. Iced-airfoil aerodynamics. *Progress in Aerospace Sciences*, 41(5):323–362, 2005.

[22] Ø. Breivik, A. A. Allen, C. Maisondieu, and J. C. Roth. Wind-induced drift of objects at sea: The leeway field method. *Applied Ocean Research*, 33(2): 100–109, 2011.

[23] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[24] E. F. Camacho, D. R. Ramírez, D. Limón, D. M. De La Peña, and T. Alamo. Model predictive control techniques for hybrid systems. *Annual reviews in control*, 34(1):21–31, 2010.

[25] A. Chakravarty. Four-dimensional fuel-optimal guidance in the presence of winds. *Journal of Guidance*, 8:16–22, 1985. doi: 10.2514/3.19929.

[26] P. Chandler, S. Rasmussen, and M. Pachter. UAV cooperative path planning. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1255–1265, 2000.

[27] K. Chen, K. Fu, M. Yan, X. Gao, X. Sun, and X. Wei. Semantic segmentation of aerial images with shuffling convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(2):173–177, Feb 2018.

[28] M. Coombes, W.-H. Chen, and P. Render. Reachability analysis of landing sites for forced landing of a UAS inWind using trochoidal turn paths matthew. In *2015 International Conference on Unmanned Aerial Systems (ICUAS)*. ICUAS 2017, June 2015.

[29] M. Coombes, W.-H. Chen, and P. Render. Landing site reachability in a forced landing of unmanned aircraft in wind. *Journal of Aircraft*, 54:1415–1427, 2017. doi: 10.2514/1.C033856.

[30] G. Cruz and A. Bernardino. Aerial detection in maritime scenarios using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 373–384. Springer, 2016.

[31] B. Di, R. Zhou, and H. Duan. Potential field based receding horizon motion planning for centrality-aware multiple UAV cooperative surveillance. *Aerospace Science and Technology*, 46:386–397, 2015.

[32] C. Dixon and E. W. Frew. Maintaining optimal communication chains in robotic sensor networks using mobility control. *Mobile Networks and Applications*, 14(3):281–291, 2009.

[33] U. DoD. Department of defense dictionary of military and associated terms. *Joint Publication*, pages 1–02, 2007.

[34] T. Donateo, A. Ficarella, L. Spedicato, A. Arista, and M. Ferraro. A new approach to calculating endurance in electric flight and comparing fuel cells and batteries. *Applied Energy*, 187:807–819, 2017. ISSN 03062619. doi: 10.1016/j.apenergy.2016.11.100. URL `http://dx.doi.org/10.1016/j.apenergy.2016.11.100`.

[35] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

[36] R. C. Eberhart and Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 84–88. IEEE, 2000.

[37] R. C. Eberhart, Y. Shi, and J. Kennedy. *Swarm intelligence*. Elsevier, 2001.

[38] A. Filippone. *Flight Performance of Fixed and Rotary Wing Aircraft*. Butterworth-Heinemann, 2006. ISBN 978-0-7506-6817-0.

[39] G. Forslid, H. Wieslander, E. Bengtsson, C. Wählby, J. M. Hirsch, C. R. Stark, and S. K. Sadanandan. Deep convolutional neural networks for detecting cellular changes due to malignancy. In *International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017.

[40] E. J. Forsmo, E. I. Grøtli, T. I. Fossen, and T. A. Johansen. Optimal search mission with unmanned aerial vehicles using mixed integer linear programming. In *2013 International conference on unmanned aircraft systems (ICUAS)*, pages 253–259. IEEE, 2013.

[41] A. Franco, D. Rivas, and A. Valenzuela. Optimization of unpowered descents of commercial aircraft in altitude-dependent winds. *Journal of Aircraft*, 49 (5):1460–1470, 2012. ISSN 0021-8669. doi: 10.2514/1.C031737.

[42] A. Franco, D. Rivas, and A. Valenzuela. Optimal aircraft path planning considering wind uncertainty. *European Conference for aeronautics and Space Sciences (EUCASS)*, 2017. doi: 10.13009/EUCASS2017-254.

[43] E. W. Frew and T. X. Brown. Networking issues for small unmanned aircraft systems. *Journal of Intelligent and Robotic Systems*, 54(1-3):21–37, 2009.

[44] B. Fuller and E. Mark. Generic battery rate-effect model. Technical report, NAVAL UNDERSEA WARFARE CENTER DIV NEWPORT RI AUTONOMOUS AND DEFENSIVE . . . , 2012.

[45] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of mathematics and artificial intelligence*, 31(1-4): 77–98, 2001.

[46] A. E. Gill. *Atmosphere—ocean dynamics.* Elsevier, 2016.

[47] A. R. Girard, A. S. Howell, and J. K. Hedrick. Border patrol and surveillance missions using multiple unmanned air vehicles. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 1, pages 620–625. IEEE, 2004.

[48] B. Girardet, L. Lapasset, D. Delahaye, and C. Rabut. Wind-optimal path planning: Application to aircraft trajectories. *13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*, pages 1403–1408, 2014. doi: 10.1109/ICARCV.2014.7064521.

[49] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016.

[50] Z. Goraj. An overview of the deicing and anti-icing technologies with prospects for the future. In *24th international congress of the aeronautical sciences*, volume 29, 2004.

[51] A. Grancharova, E. I. Grøtli, D.-T. Ho, and T. A. Johansen. UAVs trajectory planning by distributed MPC under radio communication path loss constraints. *Journal of Intelligent & Robotic Systems*, 79(1):115, 2015.

[52] K. Gryte, R. Hann, M. Alam, J. Roháč, T. A. Johansen, and T. I. Fossen. Aerodynamic modeling of the skywalker x8 fixed-wing unmanned aerial vehicle. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 826–835. IEEE, 2018.

[53] S. Gudmundsson. *A Biomimetic, Energy-Harvesting, Obstacle-Avoiding, Path-Planning Algorithm for UAVs.* PhD thesis, Embry-Riddle Aeronautical University, 2016.

[54] S. Gudmundsson. *A Biomimetic, Energy-Harvesting, Obstacle-Avoiding, Path-Planning Algorithm for UAVs.* PhD thesis, Embry-Riddle Aeronautical University, 2016.

[55] W. G. Habashi, F. Morency, and H. Beaugendre. FENSAP-ICE: a comprehensive 3d simulation tool for in-flight icing. In *7th International Congress of Fluid Dynamics and Propulsion, Sharm-El-Sheikh, Egypt, December*, pages 1–7, 2001.

[56] W. G. Habashi, M. Aubé, G. Baruzzi, F. Morency, P. Tran, and J. C. Narramore. FENSAP-ICE: a fully-3d in-flight icing simulation system for aircraft, rotorcraft and UAVS. In *24th International Congress of The Aeronautical Sciences*, pages 2004–7, 2004.

[57] F. J. Hale and A. R. Steiger. Effects of wind on aircraft cruise performance. In *1978 AIAA Aircraft Systems and Technology Conference*. American Institute of Aeronautics and Astronautics, Aug. 1978.

[58] M. Hals and M. Skjønhaug. Optimization of coordinated path planning for autonomous vehicles in ice management. Master's thesis, NTNU, 2017.

[59] R. Hann. UAV icing: Comparison of LEWICE and FENSAP-ICE for ice accretion and performance degradation. In *2018 Atmospheric and Space Environments Conference*, page 2861, 2018.

[60] R. Hann. UAV icing: Comparison of LEWICE and FENSAP-ICE for anti-icing loads. In *AIAA Scitech 2019 Forum*, page 1286, 2019.

[61] R. Hann. UAV icing: Ice accretion experiments and validation, 2019.

[62] R. Hann, K. Borup, A. Zolich, K. Sorensen, H. Vestad, M. Steinert, and T. Johansen. Experimental investigations of an icing protection system for UAVs. Technical report, SAE Technical Paper, 2019.

[63] M. R. Haque, M. Muhammad, D. Swarnaker, and M. Arifuzzaman. Autonomous quadcopter for product home delivery. In *2014 International Conference on Electrical Engineering and Information & Communication Technology*, pages 1–5. IEEE, 2014.

[64] J. Haugen. *Autonomous aerial ice observation*. PhD thesis, NTNU, 2014.

[65] J. Haugen and L. Imsland. Monitoring moving objects using aerial mobile sensors. *IEEE Transactions on Control Systems Technology*, 24(2):475–486, 2015.

[66] Z. He and L. Zhao. The comparison of four UAV path planning algorithms based on geometry search algorithm. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 2, pages 33–36. IEEE, 2017.

[67] E. Honkavaara, H. Saari, J. Kaivosoja, I. Pölönen, T. Hakala, P. Litkey, J. Mäkynen, and L. Pesonen. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture. *Remote Sensing*, 5(10):5006–5039, 2013.

[68] B. Houska, H. J. Ferreau, and M. Diehl. ACADO toolkit—an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[69] A. R. Hovenburg, F. A. A. Andrade, R. Hann, C. D. Rodin, T. A. Johansen, and R. Storvold. Long range path planning using an aircraft performance model for battery powered sUAS equipped with icing protection system. Submitted for publication.

[70] A. R. Hovenburg, F. A. de Alcantara Andrade, C. D. Rodin, T. A. Johansen, and R. Storvold. Contingency path planning for hybrid-electric UAS. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 37–42. IEEE, 2017.

[71] A. R. Hovenburg, T. A. Johansen, and R. Storvold. Mission performance trade-offs of battery-powered sUAS. In *2017 International Conference on Unmanned Aerial Systems (ICUAS)*. ICUAS 2017, June 2017.

[72] A. R. Hovenburg, F. A. de Alcantara Andrade, C. D. Rodin, T. A. Johansen, and R. Storvold. Inclusion of horizontal wind maps in path planning optimization of UAS. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 513–520. IEEE, 2018.

[73] I. IAMSAR. International aeronautical and maritime search and rescue manual. *Mission coordination*, 2, 2007.

[74] W. International. 3W-28i HFE FI test data. `http://3w-international.com/Drone_Engines_Sale/engine-details-test-data/engine-data-3W-28i-HFE-FI.php`, 2016.

[75] M. Jakob, E. Semsch, D. Pavlicek, and M. Pechoucek. Occlusion-aware multi-uav surveillance of multiple urban areas. In *6th Workshop on Agents in Traffic and Transportation (ATT 2010)*, pages 59–66. Citeseer, 2010.

[76] O. Janssens, R. Van de Walle, M. Loccufier, and S. Van Hoecke. Deep learning for infrared thermal image based machine health monitoring. *IEEE/ASME Transactions on Mechatronics*, 2017.

[77] A. L. Jennings, R. Ordonez, and N. Ceccarelli. An ant colony optimization using training data applied to UAV way point path planning in wind. In *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pages 1–8. IEEE, 2008.

[78] T. A. Johansen and T. Perez. Unmanned aerial surveillance system for hazard collision avoidance in autonomous shipping. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1056–1065. IEEE, 2016.

[79] T. A. Johansen, A. Zolich, T. Hansen, and A. J. Sørensen. Unmanned aerial vehicle as communication relay for autonomous underwater vehicle—field tests. In *Globecom Workshops (GC Wkshps), 2014*, pages 1469–1474. IEEE, 2014.

[80] V. John, S. Mita, Z. Liu, and B. Qi. Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks. In *Machine Vision Applications (MVA), 14th IAPR International Conference on*, 2015.

[81] Y. Kang and J. K. Hedrick. Linear tracking for a fixed-wing UAV using nonlinear model predictive control. *IEEE Transactions on Control Systems Technology*, 17(5):1202–1210, 2009.

[82] G. N. Karystinos and D. A. Pados. On overfitting, generalization, and randomly expanded training sets. *IEEE Transactions on Neural Networks*, 11 (5):1050–1057, Sep 2000.

[83] B. Kellenberger, M. Volpi, and D. Tuia. Fast animal detection in UAV images using convolutional neural networks. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[84] S. Kim, P. Silson, A. Tsourdos, and M. Shanmugavel. Dubins path planning of multiple unmanned airborne vehicles for communication relay. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(1):12–25, 2011.

[85] S. Kim, H. Oh, J. Suk, and A. Tsourdos. Coordinated trajectory planning for efficient communication relay using multiple UAVs. *Control Engineering Practice*, 29:42–49, 2014.

[86] *Maritime Broadband Radio - MBR.* Kongsberg, (accessed March 23, 2018). URL `https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/BCCBAC3EA4EA6785C1257E280039BD63`.

[87] M. Kościelski, R. Miler, and M. Zieliński. Maritime situational awareness (MSA). *Zeszyty Naukowe Akademii Marynarki Wojennej*, 48:79–88, 2007.

[88] J. D. Kraus and R. J. Marhefka. Antennas for all applications. *Antennas for all applications, by Kraus, John Daniel; Marhefka, Ronald J. New York: McGraw-Hill, c2002.*, 2002.

[89] J. W. Langelaan, N. Alley, and J. Neidhoefer. Wind field estimation for small unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 34 (4):1016–1030, 2011. doi: 10.2514/1.52532.

[90] S. J. Lee, T. Chen, L. Yu, and C. H. Lai. Image classification based on the boost convolutional neural network. *IEEE Access*, 6, 2018.

[91] F. S. Leira, T. A. Johansen, and T. I. Fossen. Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera. In *2015 IEEE aerospace conference*, pages 1–10. IEEE, 2015.

[92] S. Li, X. Sun, and Y. Xu. Particle swarm optimization for route planning of unmanned aerial vehicles. In *Information Acquisition, 2006 IEEE International Conference on*, pages 1213–1218. IEEE, 2006.

[93] S. Lim and D. Lee. Stable improved softmax using constant normalisation. *Electronics Letters*, 53(23):1504–1506, 2017.

[94] N. Liu, J. Han, T. Liu, and X. Li. Learning to predict eye fixations via multiresolution convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(2):392–404, Feb 2018.

[95] S. Long. Drones and precision agriculture: The future of farming. *Microdrones*, 2017. URL `https://www.microdrones.com/en/content/drones-and-precision-agriculture-the-future-of-farming/`.

[96] T. Luna. New technology making drones easier, more affordable. *The Boston Globe*, 2015. URL `https://www.bostonglobe.com/business/2015/12/08/very-drone-christmas/bjoMHPmiidy0WHQXy6LjSN/story.html?p1=Article_Recommended_ArticleText#comments`.

[97] F. T. Lynch and A. Khodadoust. Effects of ice accretions on aircraft aerodynamics. *Progress in Aerospace Sciences*, 37(8):669–767, 2001.

[98] F. Maire, L. Mejias, and A. Hodgson. A convolutional neural network for automatic analysis of aerial imagery. In *Digital lmage Computing: Techniques and Applications (DlCTA), 2014 International Conference on*, 2014.

[99] M. Marwa, S. M. Martin, B. C. Martos, and R. P. Anderson. Analytic and numeric forms for the performance of propeller-powered electric and hybrid aircraft. In *55th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 2017.

[100] M. J. Mears. Energy harvesting for unmanned air vehicle systems using dynamic soaring. *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, January 2012. doi: 10.2514/6.2012-851.

[101] K. Merical, T. Beechner, and P. Yelvington. Hybrid-electric, heavy-fuel propulsion system for small unmanned aircraft. *International Journal of Aerospace*, 7:126–134, 2014. doi: 10.4271/2014-01-2222.

[102] M. Müller, M. Homleid, K.-I. Ivarsson, M. A. Køltzow, M. Lindskog, K. H. Midtbø, U. Andrae, T. Aspelien, L. Berggren, D. Bjørge, et al. AROME-MetCoOp: A nordic convective-scale operational weather prediction model. *Weather and Forecasting*, 32(2):609–627, 2017.

[103] T. Nakazawa and D. V. Kulkarni. Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing*, 2018.

[104] D. R. Nelson, D. B. Barber, T. W. Mclain, and R. W. Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23: 519–529, 2007. doi: 10.1109/TRO.2007.898976.

[105] N. Nigam. The multiple unmanned air vehicle persistent surveillance problem: A review. *Machines*, 2(1):13–72, 2014.

[106] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras. Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(6):898–912, 2003.

[107] M. Novak. U.S. border patrol flew more drone missions last year than ever before. *GIZMODO*, 2018. URL `https://bit.ly/2Ok2V88`.

[108] M. R. O'Leary. *The dictionary of homeland security and defense*. iUniverse, 2006.

[109] I. C. A. Organization. *Manual of the ICAO Standard Atmosphere: Extended to 80 Kilometres (262 500 Feet)*, volume 7488. International Civil Aviation Organization, 1993.

[110] D. Palma, A. Zolich, Y. Jiang, and T. A. Johansen. Unmanned aerial vehicles as data mules: An experimental assessment. *IEEE Access*, 2017.

[111] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[112] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa. The LSTS toolchain for networked vehicle systems. In *2013 MTS/IEEE OCEANS-Bergen*, pages 1–9. IEEE, 2013.

[113] L. Prechelt. Early stopping-but when? *Neural Networks: Tricks of the trade*, pages 553–553, 1998.

[114] S. Prince. *Computer Vision: Models, Learning, and Inference*. Computer Vision: Models, Learning, and Inference. Cambridge University Press, 2012. ISBN 9781107011793.

[115] F. F. Ramírez, D. S. Benitez, E. B. Portas, and J. A. L. Orozco. Coordinated sea rescue system based on unmanned air vehicles and surface vessels. In *OCEANS 2011 IEEE-Spain*, pages 1–10. IEEE, 2011.

[116] E. v. Rees. A quick introduction to drone photogrammetry. *medium.com*, 2019. URL `https://medium.com/@ericvanrees/a-quick-introduction-to-drone-photogrammetry-381a15478fdc`.

[117] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi. Uav photogrammetry for mapping and 3d modeling–current status and future perspectives. *International archives of the photogrammetry, remote sensing and spatial information sciences*, 38(1):C22, 2011.

[118] A. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference, 2002. Proceedings of the 2002*, volume 3, pages 1936–1941. IEEE, 2002.

[119] V. Roberge, M. Tarbouchi, and G. Labonté. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, 9(1):132–141, 2013.

[120] C. D. Rodin, F. A. A. Andrade, A. R. Hovenburg, T. A. Johansen, and R. Storvold. A survey of design considerations of optical imaging stabilization systems for small unmanned aerial systems. Submitted for publication.

[121] C. D. Rodin, L. N. de Lima, F. A. de Alcantara Andrade, D. B. Haddad, T. A. Johansen, and R. Storvold. Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[122] L. Rokach. *Pattern classification using ensemble methods*, volume 75. World Scientific, 2010.

[123] A. Rucco, A. P. Aguiar, F. L. Pereira, and J. B. de Sousa. A predictive path-following approach for fixed-wing unmanned aerial vehicles in presence of wind disturbances. In *Robot 2015: Second Iberian Robotics Conference*, pages 623–634. Springer, 2016.

[124] V. San Juan, M. Santos, and J. M. Andújar. Intelligent UAV map generation and discrete path planning for search and rescue operations. *Complexity*, 2018, 2018.

[125] D. T. Sandwell. Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data. *Geophysical Research Letters*, 14(2):139–142, 1987. ISSN 1944-8007. doi: 10.1029/GL014i002p00139.

[126] A. Satariano. Facebook halts aquila, its internet drone project. *The New York Times*, 2018. URL `https://www.nytimes.com/2018/06/27/technology/facebook-drone-internet.html`.

[127] R. Sengupta, J. Connors, B. Kehoe, Z. Kim, T. Kuhn, and W. Jared. Final report - autonomous search and rescue with scaneagle, 2010.

[128] C. Sheppard and M. Rahnemoonfar. Real-time scene understanding for uav imagery based on deep convolutional neural networks. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[129] R. Siquig. Impact of icing on unmanned aerial vehicle (UAV) operations. Technical report, NAVAL ENVIRONMENTAL PREDICTION RESEARCH FACILITY MONTEREY CA, 1990.

[130] R. Sisk. Coast Guard commandant boosts plan to outfit cutters with ScanEagle drones. *Military.com*, 2019. URL `https://bit.ly/2ZqptFv`.

[131] E. Skjong, S. A. Nundal, F. S. Leira, and T. A. Johansen. Autonomous search and tracking of objects using model predictive control of unmanned aerial vehicle and gimbal: Hardware-in-the-loop simulation of payload and avionics. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 904–913. IEEE, 2015.

[132] A. Sobral. BGSLibrary: An opencv c++ background subtraction library. In *IX Workshop de Visão Computacional (WVC'2013)*, Rio de Janeiro, Brazil, Jun 2013. URL `https://github.com/andrewssobral/bgslibrary`.

[133] K. L. Sørensen. Autonomous icing protection solution for small unmanned aircraft: An icing detection, anti-icing and de-icing solution. 2016.

[134] K. L. Sørensen, A. S. Helland, and T. A. Johansen. Carbon nanomaterial-based wing temperature control system for in-flight anti-icing and de-icing of unmanned aerial vehicles. In *2015 IEEE Aerospace Conference*, pages 1–6. IEEE, 2015.

[135] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[136] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[137] R. F. Stengel. *Flight dynamics*. Princeton University Press, 2015.

[138] K. Szilder and S. McIlwain. In-flight icing of UAVs-the influence of reynolds number on the ice accretion process. Technical report, SAE Technical Paper, 2011.

[139] K. Szilder and W. Yuan. The influence of ice accretion on the aerodynamic performance of a UAS airfoil. In *53rd AIAA Aerospace Sciences Meeting*, page 0536, 2015.

[140] V. Tasikas. Unmanned aerial vehicles and the doctrine of hot pursuit: A new era of coast guard maritime law enforcement operations. *Tul. Mar. LJ*, 29: 59, 2004.

[141] A. Team. Ardupilot. `ardupilot.org`, 2019. [Online; accessed 10-July-2019].

[142] M. Team. MAVLink micro air vehicle protocol. `mavlink.io/en`, 2019. [Online; accessed 10-July-2019].

[143] G. Thompson, B. E. Nygaard, L. Makkonen, and S. Dierer. Using the weather research and forecasting (WRF) model to predict ground/structural icing. In *13th International Workshop on Atmospheric Icing on Structures, METEOTEST, Andermatt, Switzerland*, 2009.

[144] L. Traub. Range and endurance estimates for battery-powered aircraft. *Journal of Aircraft*, pages 703–707, 2011. ISSN 0021-8669. doi: 10.2514/1.C031027.

[145] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche. A generic battery model for the dynamic simulation of hybrid electric vehicles. In *2007 IEEE Vehicle Power and Propulsion Conference*, pages 284–289. Ieee, 2007.

[146] P. Trodden and A. Richards. Multi-vehicle cooperative search using distributed model predictive control. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 7138, 2008.

[147] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.

[148] A. Valenzuela and D. Rivas. Analysis of wind-shear effects on optimal aircraft cruise. *Journal of Guidance, Control and Dynamics*, 39(9):2148–2155, 9 2016. ISSN 0731-5090. doi: 10.2514/1.G000251.

[149] M. Valkonen, K. Kartasalo, K. Liimatainen, M. Nykter, L. Latonen, and P. Ruusuvuori. Dual structured convolutional neural network with feature augmentation for quantitative characterization of tissue histology. In *International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017.

[150] G. Wallace. Amazon requests FAA approval of delivery-drone plans. *CNN Business*, 2019. URL https://edition.cnn.com/2019/08/08/tech/amazon-faa-drones/index.html.

[151] Y. Wang, J. Lin, and Z. Wang. An energy-efficient architecture for binary weight convolutional neural networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(2):280–293, Feb 2018.

[152] A. Weintrit. *Marine navigation and safety of sea transportation: navigational problems*. CRC Press, 2013.

[153] L. A. Weitz. *Derivation of a Point-Mass Aircraft Model used for Fast-Time Simulation*. MITRE Corporation, 2015.

[154] E. Whalen, A. Broeren, M. Bragg, and S. Lee. Characteristics of runback ice accretions on airfoils and their aerodynamics effects. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, page 1065, 2005.

[155] W. Wright. User's manual for LEWICE version 3.2. 2008.

[156] W. Wright and A. Rutkowski. Validation results for LEWICE 2.0" and CD-ROM, 1999.

[157] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su, and Y. Yu. Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by adaptive grasshopper optimization algorithm. *Aerospace Science and Technology*, 70:497–510, 2017.

[158] J. Wu, H. Wang, Y. Huang, Z. Su, and M. Zhang. Energy management strategy for solar-powered UAV long-endurance target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 2018.

[159] X. Xue, W. Qin, Z. Sun, S. Zhang, L. Zhou, P. Wu, et al. Effects of N-3 UAV spraying methods on the efficiency of insecticides against planthoppers and cnaphalocrocis medinalis. *Acta Phytophylacica Sinica*, 40(3):273–278, 2013.

[160] K. Yang and S. Sukkarieh. 3D smooth path planning for a UAV in cluttered natural environments. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 794–800. IEEE, 2008.

[161] N. Yang, H. Tang, H. Sun, and X. Yang. DropBand: A simple and effective method for promoting the scene classification accuracy of convolutional neural networks for vhr remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 15(2):257–261, Feb 2018.

[162] D. Yu and L. Deng. *Automatic speech recognition: A deep learning approach.* Springer, 2014.

[163] A. Zolich, T. A. Johansen, K. Cisek, and K. Klausen. Unmanned aerial system architecture for maritime missions. design & hardware description. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 342–350. IEEE, 2015.

[164] A. Zolich, A. Sœgrov, E. Vågsholm, V. Hovstein, and T. A. Johansen. Coordinated maritime missions of unmanned vehicles—network architecture and performance analysis. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–7. IEEE, 2017.

[165] A. Zolich, D. Palma, K. Kansanen, K. Fjørtoft, J. Sousa, K. H. Johansson, Y. Jiang, H. Dong, and T. A. Johansen. Survey on communication and networks for autonomous marine systems. *Journal of Intelligent & Robotic Systems*, pages 1–25, 2018.

ISBN 978-82-326-4248-9 (printed version)
ISBN 978-82-326-4249-6 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Doctoral Thesis

Fabio Augusto de Alcantara Andrade

Doctoral theses at NTNU, 2019:326

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

NTNU

Doctoral theses at NTNU, 2019:326

Fabio Augusto de Alcantara Andrade

Real-time and offline path planning of Unmanned Aerial Vehicles for maritime and coastal applications

NTNU
Norwegian University of
Science and Technology