Tesis Doctoral
Ingeniería Automática, Electrónica y de Telecomunicación

UAS planning and trajectory generation for safe and long-duration oceanic and coastal missions

Autor: Leopoldo Rodríguez Salazar
Directores: José Antonio Cobano Suárez
Aníbal Ollero Baturone

Departmento de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018

Tesis Doctoral
Ingeniería Automática, Electrónica y de Telecomunicación

UAS planning and trajectory generation for safe and long-duration oceanic and coastal missions

Autor:

Leopoldo Rodríguez Salazar

Directores:

José Antonio Cobano Suárez

Profesor (Universidad Pablo de Olavide)

Aníbal Ollero Baturone

Catedrático

Departmento de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

2018

Tesis Doctoral: UAS planning and trajectory generation for safe and long-duration oceanic and coastal missions

Autor: Leopoldo Rodríguez Salazar
Directores: José Antonio Cobano Suárez
Aníbal Ollero Baturone

El tribunal nombrado para juzgar la Tesis arriba indicada, compuesto por los siguientes doctores:

Presidente:

Vocales:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Date:

Leticia
If I lay here
would you lie with me
and just forget the world

Acknowledgements

The last three years were very challenging but also rewarding both in my professional and personal life. This thesis is an accomplished goal that could not happen without the support of so many people. Following the example of the Academy Awards, first, I would like to thank my director, Jose Antonio Cobano, who has helped me since day one, not only in aspects related to this thesis but others that have simplified the way through here, he was my mentor and also became my friend. His guidance, expertise, and vision have helped me to tackle different problems and sort all kinds of obstacles in this research and during my stay in Seville. I would also like to thank my thesis director and my tutor Prof. Anibal Ollero Baturone for his guidance, and to push me every day towards excellence and for being a role model of hard work. I want to thank Prof. Tor Arne Johansen for his work in managing the MarineUAS project that has provided us with vast experiences in the field and to organize broad activities that have enriched my professional life.

I want to thank my dearest friend Fotios Balampanis who was my partner in crime in this adventure, we were in the same boat, and sailed this rough waters successfully. I hope to share more experiences with you in the future.

I would like to give special thanks to Rafael Salmoral, founder of Proskytec; thanks to this project became a terrific friend. He was an integral part of the experiments of this thesis and has helped me enormously with different aspects of the project. I would like to thank Juan Braga, for his support and encouragement on publishing. Margarida Faria, for her great talks and for being a friend of Leticia and me. Thanks to Du Ho, for the great times in all the activities, and thanks to my colleague and friend Bahareh for her friendship and long or not so long conversations. Finally, I would like to thank the rest of MarineUAS fellows, because their companionship, expertise, and ideas helped me enormously in this goal and certainly would help me in the future. You guys are among the most brilliant people that I've ever met.

My most profound gratitude goes to Leticia, my beloved wife and the love of my life. For her unconditional love and for being my pillar, support and best friend in this adventure, words are not sufficient to thank her for keeping me focused on what matters, for not letting me fall, for telling me when I was wearing the sweater in the wrong way after a long night at work. We have a great future together, thanks for being part of this dream, because my dreams are also yours!

Thanks to my mom, who has been my greatest hero and my role model. Thanks again for not letting me miss a single day in elementary school, thanks for teaching me to read and for encouraging me to pursue my dreams. To my grandma that left a few years ago but her example and legacy are still here. Thanks to mom's and grandma's cooking lessons Mexico was not that far. And to all the Rodriguez Salazar family, and to the Aguilar Aguilar family, many thanks for your support and for always encouraging me.

Last but not least, I want to thank Otto and Kiwi, the most fabulous and spectacular dogs in the world. They were there every morning, every afternoon and every night, checking on my progress on the thesis, but most importantly protecting me from myself, from the outdoor noises, and from the bad neighbors.

Leopoldo Rodríguez Salazar
Sevilla, 2018

Agradecimientos

Los últimos tres años fueron retadores pero también fructíferos en mi vida profesional y personal. Esta tesis es una meta cumplida que no pudo existir sin el apoyo de muchísima gente. Al más puro estilo de los premios Óscar, primero, quisiera agradecer a mi director, Jose Antonio Cobano, que ayudó desde el primer día no solo con aspectos relacionados con esta tesis sino con muchos otros que han simplificado el camino hasta aquí. Su guía, conocimientos y visión me han ayudado a atacar distintos problemas y superar obstáculos durante esta investigación. Quisiera también agradecer a mi director de tesis y tutor, el profesor Aníbal Ollero, por guiarme y empujarme todos los días hacia la excelencia siendo un modelo a seguir de trabajo arduo y de obtención de éxitos. Agradezco también al profesor Tor Arne Johansen, por su trabajo como director del proyecto MarineUAS, que nos dio grandes experiencias a través de excelentes actividades que han enriquecido mi vida personal y profesional.

También quisiera agradecer a mi querido amigo Fotios Balampanis que fue mi camarada, navegando en el mismo barco y superando estas duras aguas exitosamente. Espero que las aventuras no terminen y solo sean el comienzo.

Quisiera agradecer especialmente a Rafael Salmoral, fundador de Proskytec y que se convirtió gracias a este proyecto en un buen amigo, por facilitar la campaña experimental de esta investigación, ayudándonos en múltiples aspectos del proyecto. También agradezco a Juan Braga por siempre aconsejarme publicar, a Margarida Faria por ser una gran amiga para mi esposa y para mi. A Du Ho por sus consejos y los momentos divertidos en las distintas actividades del proyecto y también agradezco a mi compañera y amiga Bahareh por sus largas y no tan largas conversaciones. Al resto de miembros del proyecto MarineUAS, estoy en deuda con ustedes, son de las personas más brillantes que conozco y su amistad y experiencia me ayudó enormemente en este sueño.

Mi mayor gratitud es para Leticia, el amor de mi vida, por su amor incondicional y por ser mi pilar, mi sustento y mi mejor amiga en esta aventura, las palabras no son suficientes para agradecerle que me hayas mantenido enfocado en lo que realmente importa, por no dejarme caer y por decirme cuando usaba el suéter al revés después de una desvelada. Tenemos un gran futuro juntos, gracias por compartir este sueño, pues mis sueños siempre son tuyos.

Gracias a mi mamá, *mi jefecita*, por ser mi héroe y mi modelo a seguir. Le agradezco haber hecho un esfuerzo por que no me perdiera ningún día de escuela, por enseñarme a leer y por enseñarme a nunca rendirme para cumplir mis sueños. A mi abuela, que ya no está con nosotros, su ejemplo perdura y perdurará para siempre. Gracias a mi abuela y mamá por enseñarme a cocinar, sin eso México estaría más lejos. A toda mi familia, Rodríguez Salazar y Aguilar Aguilar, gracias por sus mensajes de apoyo.

Finalmente quisiera agradecer a Otto y Kiwi, los mejores y más espectaculares perros del mundo. Estuvieron ahí, todas las mañanas, tardes y noches, evaluando mi progreso en

la tesis, pero más importante cuidándome de mi mismo, de los ruidos extraños y de los malos vecinos.

Leopoldo Rodríguez Salazar
Sevilla, 2018

Resumen

La presente tesis doctoral, muestra el diseño de un sistema para la extensión de la duración de vuelo de sistemas autónomos no tripulados de tamaño pequeño. Este sistema fue diseñado en el contexto de misiones de vigilancia marítima y costera como parte del proyecto europeo MarineUAS. En este contexto, se han identificado tres problemas: 1) la necesidad de la estimación precisa de un campo de viento y la capacidad de identificación de fenómenos como el viento cortante o las ráfagas continuas y discretas para que puedan ser utilizadas potencialmente para la extracción de energía para mejorar la duración de vuelo. 2) La necesidad de generar trayectorias suaves para la extracción de energía considerando la dinámica de las plataformas de vuelo y 3) la habilidad de seguir dichas trayectorias. Para el primer problema, el uso de un método de computación directa permite determinar el campo de viento (velocidad y tasa de cambio de la velocidad de viento) sin la utilización de un estimador óptimo. Sin embargo, también se consideraron varios métodos y a partir de un análisis extenso se presentan diferentes comparativas de estos métodos, en el que se muestran las ventajas y desventajas de los mismos. Adicionalmente, la identificación de distintos fenómenos de viento, cómo las ráfagas, o el viento cortante, se logra a través de un innovador método que ejecuta una serie de pruebas estadísticas basadas en la distribución de Weibull y en distintos modelos dinámicos que consideran no solo la distribución del viento sino la interacción con el océano y la superficie en las respectivas capas límite. Para el segundo problema, una aproximación biomimética permitió el uso de un algoritmo complejo para la réplica de trajectorias de vuelo dinámico de aves. En dicho algoritmo se consideran observaciones presentadas por distintos científicos que permiten generar trayectorias paramétricas que consideran además restricciones cinemáticas de la plataforma en el diseño de las mismas. El tercer problema toma en consideración la curva generada y utiliza la teoría del campo de vectores para diseñar un controlador que permite seguir dicha trayectoria de manera eficiente y en tiempo real, respetando las leyes de control de bajo nivel en el autopiloto y permitiendo flexibilidad. Como complemento a este último sistema, se propone la reconfiguración dinámica de las misiones para mejorar el consumo energético durante el tiempo de vuelo considerando el viento predominante. Uno de los principales objectivos fue integrar, utilizando la metodología de ingeniería de sistemas, las distintias funciones anteriormente mencionadas en el que la ejecución de la misión fuese la prioridad. El principal logro fue haber realizado una extensa campaña experimental que permitió la validación del sistema en diferentes niveles, en el que se combinaron pruebas computacionales de alto y bajo nivel así como pruebas de campo en distintos escenarios y con distintas plataformas, lo cual permitió explorar la versatilidad del sistema. Los resultados muestran que se pueden lograr misiones más eficientes con mejoras de hasta un 20 % en consumo de batería para misiones costeras. Finalmente, de los distintos análisis computacionales efectuados se concluye que el tiempo de ejecución de toda la función de extensión del vuelo es lo suficientemente pequeño para permitir la ejecución

en tiempo real, lo cual, combinando con el diseño versátil en cuestión de arquitectura computacional, permiten la portabilidad del sistema así como la futura integración de funciones adicionales.

Abstract

In this thesis a system that aims to extend the flight duration of small Unmanned Aerial Systems (UAS) is presented. The system was designed in the context of oceanic and coastal surveillance missions as part of the MarineUAS European project. Three main problems were identified: 1) the need to accurately estimate the wind field and the capability to identify features of interest, such as, wind shear, and gusts that may be suitable to allow energy extraction to improve flight duration. 2) the need to generate smooth trajectories that extract energy, considering the UAS platform dynamics and 3) the ability to follow such paths. For the first problem, the use of a direct computation method allows determining the wind field (wind velocity and wind rate of change) without the use of an optimal estimator. Nevertheless, different wind velocity estimation methods are compared, and the pros and cons of each are exposed; in addition, the identification of features is accomplished with a novel approach that performs a real-time statistical analysis of the distribution of the wind field estimates, allowing the characterization of the shear components and also any other potential features, like continuous and discrete gusts considering complex models that take into account not only the phenomena but the interactions with the ground and ocean through their respective boundary layers. For the second problem, a biomimetic approach is presented, replicating the trajectories of soaring birds by considering observations of these birds and the replication of their swooping maneuvers using smooth parametrized curves. This allows flexibility in the curve design and also the incorporation of dynamic constraints of the platform on it. The solution of the third problem takes into account the smooth curve that was generated and among it, a type 1 Bishop moving frame is generated. Then, a novel adaptive control method based on the vector-field theory approach is proposed to calculate the error equations and the respective control law, which permits the tracking of the designed trajectory for dynamic soaring. Furthermore, an additional step was added, in which the surveillance mission is re-configured on a waypoint-to-waypoint basis for a more efficient flight considering the identified wind field. The result was that the execution of soaring trajectories would not be executed during all the mission, but only in specific legs that fulfill specific characteristics.The primary goal was to design algorithms that implement these functions and to integrate these functionalities in a systems-engineering approach, in which the mission execution is the main priority. An extensive experimental campaign was performed at different levels, in which software-in-the-loop and hardware-in-the-loop tests, together with field tests, were executed to demonstrate the efficiency of the various functions separately and integrated. The field tests and the simulations consider different scenarios and UAS platforms, showing the performance of the system in different conditions. The results showed that the system could execute a more efficient mission, with savings of up to 20 % in battery consumption, with the so-called of the Flight-Duration-Enhancement-System (FDES). Finally, the computational analysis showed that the system could be executed in real-time with minimum latency despite the use of sophisticated

algorithms; this, together with the chosen software and hardware architectures allows portability to other hardware components and the possibility of incorporating additional functions.

Short Contents

Índice general

Notation

α	Angle of attack
β	Sideslip angle
χ_c	Control law for track angle
χ_k	Von Karman's constant for turbulent zone
χ_f	Track angle
$\Delta V_a\|_z$	Change of airspeed due to dynamic soaring
η	Load factor
ε	residual error of fitness function
γ_f	Flight path angle
γ	Flight path angle
γ_c	Control law for flight path angle
κ	Weibull scaling parameter
κ_c	Charnock's constant for turbulent zone
ω_p	Peak angular frequency
ψ	Heading angle, yaw angle
Φ	Cumulative standard normal distribution
ϕ	Pitch angle
ρ	Air density
σ_n^2	measurement noise covariance
$\sigma_u, \sigma_v, \sigma_w$	Intensities of the turbulence among the body frame axis
θ	Roll angle
τ	Period of a Rayleigh cycle
τ_i	Tension value for an interpolated spline
τ_{L_1}	L1-period
υ	Weibull scaling parameter

A	Amplitude of the wave
A_σ	Shear gradient coefficient
a_T	Tangential acceleration
a_C	Centrifugal acceleration
$\mathbf{a}$	Acceleration vector
B	Binormal vector of Frenet-Serret frame
$\mathbf{b}$	Accelerometer bias
C_L	Lift coefficient
C_{D_n}	Drag coefficients
C_{D_0}	Zero-lift drag coefficients
D	Drag polar

d_m	gust length
E	System energy
$\bar{E}(\mathbf{q}(t))$	Total energy over trajectory
e_0	Oswald efficiency
e_l	Lateral error in tracking
e_o	Longitudinal error in tracking
e_w	Forward error in tracking
g	gravity constant
h	Altitude above sea level
H_u,H_v,H_w	Dryden transfer functions
$\mathbf{J}(t)_{\mathbf{w}}$	Matrix of temporal wind gradient
$\mathbf{J}_{\mathbf{w}}$	Matrix of spatial wind
$k(t)$	Curvature expressed in linear time
L	Lift
L_u,L_v,L_w	spacial wavelengths
$\mathscr{L}$	Lyapunov function
m	UAS mass
m_Ω,n_Ω	Curvilinear coordinates
$\tilde{M}_i$	Slope unitary vectors for an interpolated splines
N	Normal vector of Frenet-Serret frame
n Accelerometer measurement noise	
p,q,r	Rotational rates in the body frame
$\mathbf{Q}(\mathbf{X},\mathbf{X})$	Covariance matrix
$\mathbf{q}(\mathbf{X},x)$	Covariance vector
$q(x,x)$	Covariance value
S	Wing area
s	Arc length
T	Thrust
T	Tangent vector of Frenet-Serret frame
u,v,w	Æirspeed components
V_a	True airspeed magnitude
V_g	Ground speed magnitude
V_ζ	Sinking speed through the air
$\mathbf{w}$	Wind vector field
W_{20}	Reference wind velocity at 20 ft.
W_λ	Logarithmic shear effect in the vertical direction
W_φ	Pump effect due to the wavelength
W_p,$W_p(x)$	Predicted wind speed
W_u	Mean velocity profile in the turbulent zone
W_m	Gust magnitude
W_{ds}	minimum wind speed for dynamic soaring
W_{mp}	Most probable wind speed at a specific altitude
w_S^{NED}	Steady ambient wind in the NED frame

$w_{i(x,y,z)}$	Wind velocity expressed in the inertial frame
w_{shear}	Horizontal wind gradient
$\tilde{W}_k$	Friction velocity in the turbulent zone
x,y,z	Position components of the platform
z_0	Transition altitude or reference altitude for wind shear
$z_{0,k}$	Reference altitude for the friction velocity
z_T	Altitude where the shear velocity is zero

1 Introduction

I love the feeling of the fresh air on my face and the wind blowing trough my hair.

EVEL KNIEVEL

This thesis presents a novel system and algorithms to estimate the wind vector and identify wind features with an Unmanned Aerial System (UAS) to plan and generate energy-efficient trajectories for safe and long-duration oceanic and coastal missions. The main objectives are:

- Estimation of the wind field from the measurements obtained in the UAS with the sensors onboard.
- Identification the wind features that are suitable for energy exploitation.
- Study of the mechanics of the dynamic soaring to implement the necessary methods.
- Generation of energy-efficient trajectories for energy gain.
- Efficient tracking of the generated trajectories.
- Integration of the functionalities in a systemic manner, defining the software and hardware architectures with well-stated requirements.
- Validation the system by defining appropriate tests for the software, hardware and its integration, considering bench tests and outdoor experiments.

1.1 Motivation

The use of Unmanned Aerial System (UAS)s has been growing in multiple applications worldwide. However, there are endurance limitations, especially for small and micro unmanned aerial vehicles since they need additional cargo space for extending the propulsion energy. For larger fixed-wing UAS's this has been addressed and successfully

tested and implemented. There are many examples of this success for medium sized vehicles. For instance, in 2007, the Aerosonde Unmanned Aerial Vehicle (UAV) flew for more than 7.5 h, while monitoring hurricane Noel in very harsh windy conditions, with gusts over 115 km/h, with a total flight time of 17 h [1]. Another example of long duration flight is the Remote Aerial Vehicle for Environment Monitoring (RAVEN), which platform was capable of flying for about 10 h with a payload of 5 kg. The European Space Agency (ESA) has several ongoing research efforts, such as the Ship's Sulfur Trails Emissions Aerial Measurements (STEAM) [2], and the Remote Airborne Platform with Satellite Oversight Dependency (RAPSODY), in which long flight duration platforms are required. In the case of the RAPSODY, the selected platform Tekever AR5 provides with an endurance of 10 h for oceanic flights [3]. These platforms are all fuel-powered, which generally were derived from gasoline engines designed for other applications (lawnmowers, karts, chainsaws and string trimmers, etc.), which brought as a consequence intensive maintenance programs and costs. It was until 2010 that the first engine designed specifically for UAVs was created. This represented an increase in reliability that comes together with an increase of the platform costs, and despite the reduction of the maintenance programs, the maintenance of fuel-powered UAVs is still significantly high. The bottom-line while observing these cases of success with UAV is that there is an increasing need for alternatives to power up the UASs.

The research of this thesis focuses on small UASs. Current UAS technology has advanced in such a way that an inexperienced user can plan a route with relatively good accuracy. As the reliability has increased, applications using small UASs are proliferating. Besides, nonlinear natural effects, such as winds, can be compensated even with Commercial-Off-The-Shelf (COTS) components. Nevertheless, to reject wind effects disturbances efficiently, the use of a sensor that can provide wind measurements is sometimes limited by the platform payload and the cost. This leads to inefficient attitude compensations, producing drift and sometimes missing waypoints, which may result likely in higher energy consumptions [4]. Therefore, wind vector estimation and identification of wind features by using small UAS play an essential role in extending the duration of flight by generating energy-efficient trajectories.

This thesis analyzes the potential use of Atmospheric Energy Harvesting (AEH) as a means to increase the efficiency in flights for UASs. The system was intended to be utilized for maritime surveillance and coastal monitoring, and the design follows a systems-engineering approach with extensive tests and simulations to prove the feasibility of a Flight Duration Enhancement System (FDES), that has a trajectory generation and following system based on the wind information.

1.2 Related Work

Some years ago, state-of-the-art small UAVs of less than 25 kg were used primarily by the military in different applications. However, they have become significant tools for other sectors, such as the consumer and commercial sector. The regulation of these platforms is also very recent; in the United States, the U.S. Department of Transportation's Federal Aviation Administration (FAA) has just finalized the operational rules for the use of small UAVs, which is an opportunity of growth for commercial applications. According to [5], in 2017, more than 600 000 drones flew officially in U.S. soil, and over 2.5 million around the world. This unprecedented grew taken the interest of different sectors due to the challenges in current state-of-the-art propulsion systems. In the cargo space of small UAVs, batteries or internal combustion engines have been primarily the energy source for their propulsion. It is well known that these technologies limit the use and efficiency of the UASs as their technology keeps evolving. Therefore, there are more demanding requirements of higher autonomy range and reliability.

There are numerous efforts on exploring alternate technologies for powering UASs, such as solar-battery UASs, which have shown flight times that goes to even weeks or months. These systems are perfect for high altitude flights, since flying over the clouds will maximize the energy from the solar panels. However, they still need batteries for storing energy for night flight, and also they are not efficient in low altitude flights. The exploration of combining technologies such as solar, batteries and fuel cell for UASs have been explored since the late 1990's [6]. Enrico Cestino in 2006 has presented a comprehensive project of a solar-battery driven UAS [7] and Lee et al. have shown a hybrid solar-battery-fuel cell system for powering an Unmanned system in 2014. These technologies are indeed promising. Nevertheless, other alternatives could be explored, which utilize additional external sources of energy, such as the wind.

1.2.1 Atmospheric Energy Harvesting

Utilizing wind energy to extend the flight duration, AEH, has been widely researched [8, 9]. Such process can be divided into two main lines of work, the dynamic soaring case and the static soaring case. This division is based on the phenomena that is exploited to extract energy and potentially extend the flight duration. Several features are of interest for AEH. Thermal vortices and wind shear or gradients are ideal for energy gain. The thermal vortices have been extensively studied for static soaring : in [10], the authors present a method for Thermal localization and exploitation. In [11], a technique that integrates the use of thermal soaring with other soaring methods is shown. Cutler [9] and

Chakrabarty [8] have presented successful results in generating static soaring trajectories for flight duration extension. Ákosl et al. [12] have proposed a comprehensive study of the thermal soaring phenomena and a GPS independent control strategy soaring in a thermal updraft. The study of thermal soaring goes back to the late 1990's in which there have been examples of simulations of thermal detection and exploitation [13], and there have been successful implementations of thermal soaring exploitation applications, such as the one presented by the U.S. Naval Research Laboratory in 2008 [14] and by NASA's Autonomous Soaring Project Allen2007 in which a small UAV was programmed to detect whether it is in an updraft and use that updraft by circling. During the project, the UAV flew 17 times, gained an average altitude of 173m in 23 updrafts, and ascended 844m in one strong thermal. In one flight of this project, the UAV added 60 minutes to its endurance by soaring autonomously.

The process of utilization of vertical gradients for energy gain, namely dynamic soaring, has also been widely studied. Montella et al. [15] and Bird et al. [16] have proposed solutions for dynamic soaring exploitation in which a set of trajectories are proposed. Langelaan [17, 18] has introduced different methodologies for atmospheric energy harvesting, focusing on an integral solution but mostly in the wind field estimation utilizing a standard sensor suite, i.e., Inertial Measurement Unit (IMU) and Global Navigation Satellite System (GNSS). On its first method, Langelaan compares the predictions, the dynamic model and actual measurements of the aircraft motion. The second method utilizes estimates of the wind acceleration obtained with measurements from the GNSS velocity and comparing it with measurements from the airspeed at the vehicle. These estimates are used to calculate circular flight patterns that will be able to exploit the energy.

The basics of the dynamic soaring case has been studied in detail by numerous authors. Starting from Rayleigh [19], in which a series of observations of birds that perform soaring trajectories are described. These observations were continued by Pennywick [20, 21] and by Lissaman [22], who was the first to propose equations of motion to the soaring birds. Richardson [23, 24], from the Department of Physical Oceanography of Woods Hole, has proposed vast studies on the conditions of necessary soaring and his observations are the most complete because he has determined optimum circling periods, and suggested a model that captures the essential dynamics of wind-shear soaring and also provides estimates of the minimum shear but as a function of a desired cruising speed among large legs.

The wind velocity estimation is crucial for the Atmospheric Energy Harvesting. Johansen et al. [25] together with Wenz et al. [26, 27, 28] have proposed a method for estimation of wind velocity, the angle of attack, and the sideslip Angle for UASs. In the proposed work,

the authors utilize at first an Extended Kalman Filter (EKF) using frequency separation and kinematic, aerodynamic and wind models which evolved into a Moving Horizon Estimator (MHE). The results of these research are auspicious but only when typical flight patterns are utilized. One key advantage is that the use of the EKF avoids the need for prior knowledge about the aircraft. There have been other methods, that use different types of estimators, such as the one presented by Larrabee et al. [29] that considers the use of multiple UASs in a leader-follower formation. In here the leading vehicle is equipped with an optical flow sensor and a pitot-tube for accurate wind estimation using and an Unscented Kalman Filter (UKF). There is still room for improvement for more complex flying patterns, and one of the main disadvantages of the approaches that require an optimal estimator (Extended Kalman Filter (EKF), UKF, Moving Horizon Estimator (MHE)) is that the system needs to be excited on different axis for convergence. This condition is better for the MHE, in which the authors observed less parameter drift, and also an input noise can be incorporated into the system model, which is the main advantage compared to the EKF.

There are more straightforward approaches for wind estimation. They are easier to implement in commercial systems, which may not produce robust results concerning wind estimation. The method presented by Premerlani perhaps is the simplest method to provide with a wind estimation [30], in which the GNSS/INS information is used to determine a Direction Cosine Matrix (DCM) that is used to derive a 2D wind speed measurement. However, the work presented by Shensherao Biradar in [4] show that there is constant velocity error even in low dynamic environments. This error results in a cross-track drift with a magnitude sometimes more considerable than the distance between two waypoints, and also the loss of precision in the wind estimation. Therefore, the platform needs to be excited dramatically to keep the errors relatively low. This method has been implemented as the base method for popular Commercial-Off-The-Shelf (COTS) autopilots. Since accurate wind estimation is not considered crucial for the hobby community, this has not been improved.

The wind field estimation alone is not sufficient to perform any form of AEH. Some features need to be identified to determine if a set of underlying conditions are present to extract energy. The identification of shear has been studied in the meteorological field, and in order fields, such as wind turbine positioning. Liu, et al. [31] presents a study in which the wind distribution is analyzed to determine the wind power and how other environmental conditions, such as solar radiation, relative humidity, dew point affects wind power. Liu et al. proposed a way to fit the data of different wind measurements into a Weibull distribution. This is accomplished with the utilization of Genetic Algorithm (GA), used to determine

the parameters of the Weibull distribution. Furthermore, there have been techniques for gust detection, which were implemented for UASs: Afridi et al. [32] have proposed a system for gust detection that aims for the real-time gust detection with a machine learning tool. Yeo et al. [33] proposed the utilization of onboard flow measurements for gust detection. Etele [34] has written a complete theory on gust modeling for UAV, which is the basis of a control strategy designed to cope with the unwanted effects of gusts. Wenz et al. [26, 28, 27] and Johansen et al. [25] have proposed the approximation of continuous gusts with a Dryden wind model, which is used for the estimation of the aerodynamic coefficients.

Once the wind is accurately estimated, the conditions necessary for dynamic soaring have to be determined. Bencantel et al. [35] stated a series of conditions that are necessary for dynamic soaring, which are translated into a minimum shear required, which is proportional to the towing area, air density, and gravity acceleration and is inversely proportional to the aircraft mass. The results show that a more efficient aircraft concerning the aerodynamic parameters requires less shear to perform dynamic soaring.

1.2.2 Trajectory Generation and Tracking

Although there have been multiple approaches to solve the guidance and control problem for Unmanned Aerial Systems, there has been few that have proposed strategies for static or dynamic soaring. Nevertheless, few authors have suggested control strategies that deal with this issue. This section discusses the different approaches utilized and shows a mention of the static soaring case since it is a widely studied form of energy harvesting.

For thermal soaring, Acevedo et al. [36] have proposed the utilization of multiple gliders in a distributed approach for thermal detection with a model-based estimation that enables determining what the optimal time in a thermal to extend the duration of the flight is. Cobano et al. [10] have proposed the utilization of a trajectory planner based on the RRT* (Optimal Rapidly-exploring Random Trees) for accurate thermal exploitation. Clarke and Chen [37] evaluated different methods for trajectory generation for thermal exploitation, and have recommended the use of the Piggott [38] and the Reichmann [39] techniques for thermal centering. The Reichmann's method can be summarized into three premises: 1) as climb improves, then the flying circle shall be flattened. 2) as climbing deteriorates, the circle must be steeped, and 3) if the climb is constant, keep constant bank. These techniques have been widely used for manned gliders and were also taken into consideration for autonomous vehicles [40], in which a control system was designed to achieve the behavior mentioned above.

In the case of dynamic soaring, several research efforts deal with the planning and the trajectory generation. Lawrance [41] presented a method that utilizes Gaussian Process Regression (GPR) to generate a wind map that works together with a path planning algorithm to generate a trajectory that aims for energy gain. The trajectory considers an optimal solution without actually looking for a specific type of soaring but for maximizing a reward function that contains information of the wind field. Gao [11] has stated techniques for both thermal exploitation and shear exploitation in the context of surveillance missions for UAS's. Gao has proposed the execution of a dynamic soaring cycle, i.e., traveling across the shear layer of the wind. To merge the cycles, Gao presented a Dubins-path-based trajectory planner. Akhtar [42] addressed the problem of generating problems for energy exploitation with dynamic soaring with a direct method in real time, in which the trajectories are dependent on the coefficients of a polynomial that define the paths. The process proves to have advantages (mainly those related to real-time implementation) over the classic approach presented by Zhao [43], who consider the dynamic soaring problem a nonlinear optimal control problem and has generated off-line solutions for different wind gradient profiles. Zhu [44] suggested the determination of a so-called equilibrium position, in which the potential energy gained while flying throughout the vertical wind gradient, is higher than the wasted kinetic energy of flying upwards. Syama et al. [45] stated an off-line theory, in which given a suitable wind profile, specific characteristics of the aircraft are incorporated into a dynamic optimization routine that utilizes numerical methods to determine the optimal trajectory. Bower [46] defined a trajectory optimization method for dynamic soaring, with the objective of finding a solution for optimal periodical trajectories in different wind fields, which are stored in the database. However, Bower focused part of his work on determining the platform characteristics which are optimal for dynamic soaring. This work was also considering large-area oceanic missions, which required the study of the interaction between the sea and the ocean. The trajectory planning and control could be executed in a multi-stage approach, for instance, with the utilization of a receding horizon control to track such trajectory. In low level, there is an inner-controller that deflects the control surfaces. The details on the implementation of the receding horizon control can be found in [47]. Montella and Spetezer [15] proposed a reinforcement-learning approach that controls the aircraft while flying in shear conditions. Again this was a multi-stage approach with a predefined trajectory that is flown with a sample-based controller. The selected method for trajectory tracking was proposed by Miao et al. [48], in which the generated curve is framed with a moving frame, e.g., the Frenet-Serret frame or the Type 1 bishop frame. After the curve framing process, the error is determined and is used to define a Lyapunov function that monitors the positional and angular errors between the

body frame of the vehicle and the moving frame of the curve. Finally, this function is utilized to provide with flight path angle commands and also with track angle commands.

1.3 MarineUAS framework

The work presented in this thesis is part of the MarineUAS project, which is a program part of the Marie Sklodowska-Curie actions of the European Commission to strategically strengthen research training on Autonomous Unmanned Aerial Systems for Marine and Coastal Monitoring. It is a comprehensive researcher training program across a range of industrial and education partners in several countries, including, Norway, Spain, Portugal, Sweden and the Czech Republic. The program is designed to have a high impact on the training of individual Early Stage Researchers (ESR's) and their knowledge, skills and their future careers. MarineUAS has established a unique cooperative environment. It takes benefit of the partners' extensive and complementary expertise, operational field experience, and experimental facilities [49].

Figure 1.1 Research and training activities on the MarineUAS project include the realization of tests in harsh environments, training on RPAS piloting, aiming to design the most innovative solutions for the problems of coastal and oceanic monitoring.

The research lines of the project include the design of fault-tolerant control for coastal monitoring, iceberg detection, performance optimization, model-based diagnosis, cooperative motion planning, among others. The project aims to build a solid foundation for long-term European excellence and innovation in this field by sharing research infrastructures for field testing and disseminating the research and training outcomes and best practice of MarineUAS into the doctoral schools of the partners, as well as by fostering long-term partnerships and collaboration.

The motivation of this research is aligned with the motivation of the project, that considers the complexity of monitoring the coasts and Exclusive Economic Zones (EEZ) of the European Union, that stated in the European Strategy for Marine and Maritime Research, that there is a need to protect and manage the vulnerable natural environment and marine resources in a sustainable manner. The conditions for monitoring these resources are sometimes harsh and demanding. Therefore, innovative ideas and approaches, including the use of UASs, are necessary to deal with this objective.

1.4 Thesis contribution and summary

This thesis presents the development and implementation of a system capable of extending the flight duration of UASs by utilizing the available energy from the atmosphere by two means: the first one is an off-board wind-based waypoint sequencing planning or an on-board win-based waypoint sequencing reconfiguration. The second is the selection of portions of the given flight plan, that is suitable to execute dynamic soaring trajectories in a safe manner.

Compared to the reviewed methods in Section 1.2.2, the system focuses on the mission and optimizes portions of it to gain energy and reduce the fuel/battery consumption by harvesting energy from the atmosphere. Also, there was a large experiments campaign, which included flights in a variety of locations, conditions and scenarios. The objective was to validate the presented methods at different stages. One critical aspect is that the system was developed with a systems-engineering approach, in which a set of requirements is defined and cascaded to every function, which permits to determine clear means for validation and verification of the different system functions. The goal was to prioritize the use of COTS components and the real-time implementation of the selected methods, which drive most of the design decisions of the research.

These are the main contributions of this thesis:

- The development of a fully functional system for the implementation of smart AEH in a surveillance mission context.
- The combination and improvement of different methods for wind estimation and feature identification that are normally utilized in other fields.
- A novel methodology for trajectory generation with parametric curves that are suitable for dynamic soaring.
- The integration of a novel adaptive control algorithm for tracking complex curves that allow the realization of dynamic soaring.

- The consideration of a simpler reconfiguration of the waypoints sequencing as a means to achieve more efficiency combined with the dynamic soaring.

This thesis consists of seven chapters, with complementary information in two appendices. Next, an overview of the contents of each chapter is presented.

- Chapter 2 explains the wind models utilized for the wind field estimation and the characterization of features of interest (shear and gusts), together with the sea-air interaction; it also explains in detail the methodology for determining the wind relevant characteristics that are necessary to dynamic soaring.
- Chapter 3 describes the methodology employed for trajectory generation, the bio-inspiration for the selected type of trajectories, the parametrization technique and the trajectory generation algorithm.
- Chapter 4 sets out the methodology for path planning and trajectory tracking. It starts from the methodology to create or reconfigure a wind efficient flight path, followed by the methodology for the selection of the control points for the parametrized trajectory and finalizing with the proposed tracking methodology, including a definition of a tracking error definition and the control methods.
- Chapter 5 explains the integration of the different methods described in the previous chapters into a functional architecture. Moreover, the software and hardware architectures are described, together with the architecture of testing benches that were utilized for validation and verification activities.
- Chapter 6 shows a large experimental campaign, that was performed throughout the development of the preceding system, including the results and the analysis from the Hardware- and Software-In-The-Loop experiments, as well as the field experiments.
- Chapter 7 has the conclusions about the research and presents directions for further research.

Several publications in international conferences and indexed journals have been generated in the course of this thesis. These publications are:

- Leopoldo Rodriguez, Jose A. Cobano, Anibal Ollero *Wind characterization and mapping using fixed-wing small unmanned aerial systems*, 2016 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 178-184, Arlington (VA, USA), June 7-10, 2016.

- Leopoldo Rodriguez, Jose A. Cobano, Anibal Ollero, *Wind field estimation and identification having shear wind and discrete gusts features with a small UAS*, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5638- 5644, Deajeon (Korea), October 9-14, 2016.

- Leopoldo Rodriguez, Jose A. Cobano, Anibal Ollero, *Small UAS-Based Wind Feature Identification System, Part 1: Integration and Validation Sensor*, Special issue on UAV- Based Remote Sensing, December, 2016.

- Leopoldo Rodriguez, Jose A. Cobano, Anibal Ollero, *Qualitative Analysis of Guidance and Control Methods for Wind Gradients Exploitation with Small Fixed Wing UAS*. The 2017 International Workshop on Research, Education and Development on Unmanned Aerial Systems, RED-UAS 2017, 102-107, Linköping (Sweden), October 3-5, 2017.

- Leopoldo Rodriguez, Jose A. Cobano, Anibal Ollero, *Architecture of a Flight Endurance Enhancement System for Maritime Operations with Fixed Wing UAS*, ROBOT 2017: Third Iberian Robotics Conference, pp. 171-182, Seville (Spain), November 22-24, 2017.

- Leopoldo Rodriguez, Jose A. Cobano, Anibal Ollero, *UAS Soaring Trajectories Considering the Atmospheric Boundary Layer for Oceanic Long Duration Missions*, 2017 MTS/IEEE OES OCEANS conference, Anchorage (Alaska, USA), 2017.

- Leopoldo Rodriguez, Fotios Balampanis, Jose A. Cobano, Ivan Maza, Anibal Ollero, *Energy-Efficient Trajectory Generation with Spline Curves Considering Environmental and Dynamic Constraints for Small UAS*, 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1739-1745, Vancouver (BC, Canada), September 24–28, 2017.

- Leopoldo Rodriguez, Jose A. Cobano, Anibal Ollero, *Smooth Trajectory Generation for Wind Field Exploitation With a Small UAS*, 2017 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1241-1249, Miami (FL, USA), June 13-16, 2017.

- Leopoldo Rodriguez, Fotios Balampanis, Jose A. Cobano, Ivan Maza, Anibal Ollero, *Wind Efficient Path Planning and Reconfiguration of UAS in Future ATM*, Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle (Washington, USA), 2017.

- Fotios Balampanis, Leopoldo Rodrıguez, Jose A. Cobano, Ivan Maza, Anibal Ollero, *A path enhancement experimental framework for heterogeneous non holonomic flying vehicles* Submitted to IROS 2018 - Under Review.
- Leopoldo Rodriguez, Jose A. Cobano, Alberto Galán, Guillerno Heredia, Anibal Ollero *Wind estimation and trajectory generation methods for UAS long duration missions*, Submitted to Journal of Intelligent Robotic Systems (JINT 2018) - under review

2 Wind estimation and identification

The wind of change blows straight into the face of time.

Scorpions

Knowledge of the environment is critical for UAS mission planning and execution. In particular, the atmosphere plays a vital role while flying; information about the wind phenomena could help to improve the safety of the mission and even to a possible increase of the flight duration. Hence, different methods to understand the wind interaction with the UAV shall be studied to provide with information to avoid any hazardous condition and to increase (to take advantage of) the available energy.

This chapter describes a multi-stage approach for the characterization of the wind dynamics, starting from the identification of the incident wind field on the aircraft while flying followed by a characterization of the features that allow the creation of a wind map for features of interest that can be exploited for energy gain. As stated in Chapter 1, there have been numerous research efforts on estimating the wind vector of a moving aerial vehicle. The most common approach is the utilization of low-cost navigation sensors, such as the combination of IMU, GNSS, and an air probe through the use of a Bayesian filter or even directly with a simplified model. For this research, a simple approach namely, *direct computation method* is selected as the means for estimating the wind vector. Furthermore, this method is compared with existing and popular approaches which have been proposed not only for soaring applications but others, such as icing detection. Also, the knowledge of the wind field, i.e., wind velocity, the rate of change of wind velocity and the wind gradient, is of particular interest to perform safer and more efficient flights. The additional information on the wind may allow better navigation and to identify potential areas to harvest or to avoid. This identification process starts with the local wind vector estimator; then this information can be stored and utilized to create a full wind map with identified features of interest that can be used online or off-line for different purposes.

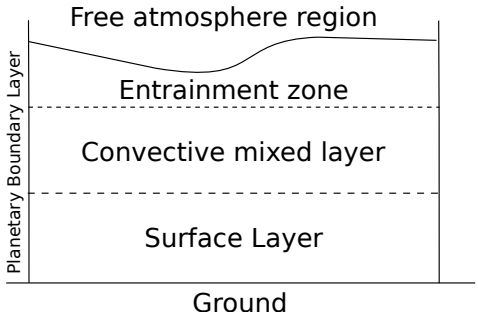


Figure 2.1 Structure of the Planetary Boundary Layer (PBL) [11].

2.1 Wind models

The atmosphere is divided into five layers based on temperature: troposphere, stratosphere, mesosphere, thermosphere, and exosphere. The phenomena studied in this research occurs in the troposphere, which contains up to 75% of the air on the planet. This layer extends from the surface up to an altitude of that goes between 9 km and 16 km of height, depending if it is measured at the poles or the equator respectively. The troposphere can be divided in the Planetary Boundary Layer (PBL), also called Atmospheric Boundary Layer (ABL), and the free atmospheric region. According to Gao [11], the thickness of the PBL varies with temperature, so that in extreme cold it can go up to 100 m and in very hot equatorial areas can be of several kilometers. Also, the PBL can be divided into three layers where different phenomena occur: the surface layer, the convective mixed layer, and the entrainment zone (see Figure 2.1).

There is a significant presence of wind gradients in the surface layers, which are determined by different factors. These include surface friction, or interaction with the ocean waves, daytime solar heating, radiative night cooling, humidity, stability conditions, etc. This zone and the interaction with the surface and the ocean is where most of the research on this chapter is focused because these conditions affect the flight characteristics of small UAS. Conjointly, the mixed layer is also of interest due to the presence of convective wind patterns, namely thermals or thermal columns, which are suitable for energy harvesting.

The forthcoming models are expressed in different reference frames :

1. The Inertial frame, I: it has its origin at the center of the earth. The x axis points to the vernal equinox, the z axis points to the north polar axis, and y points to the resultant right-handed direction.

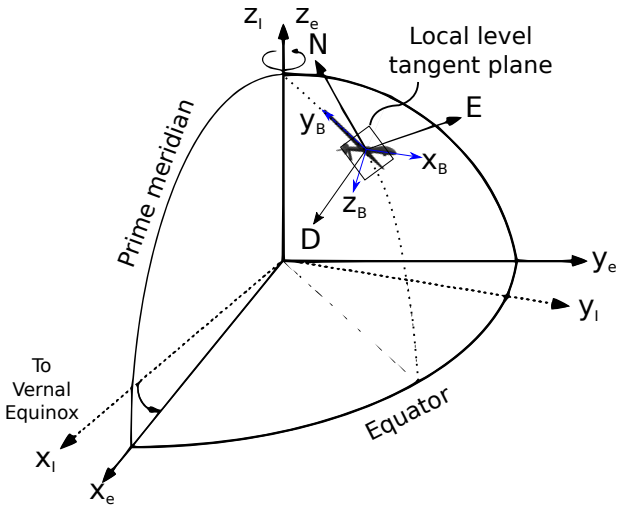


Figure 2.2 Different reference frames utilized in the definition of the wind models.

2. The Earth-Centered-Earth-Fixed (ECEF) frame, *e*: it has its origin at the center of the earth. The *x* axis points to the intersection of the prime meridian with the equator, the *z* axis points to the north polar axis and *y* points to the resultant right-handed direction.

3. The navigation frame, or North-East-Down (NED) frame: it has its origin at the aircraft center of mass. Its axes are oriented along the geodetic directions defined by the Earth's surface: one axis points to the polar direction. Another axis points east along a latitude curve, and the last axis points towards the earth surface.

4. The body frame *B*: it has its origin at the platform UAV center of mass, the *x* axis points to the right wing, the *z* axis points towards the bottom of the vehicle

5. The wind frame *W*: it is a convenient frame to express the aerodynamic forces and moments acting on an aircraft, having its origin on the aircraft aerodynamic center with the *x* axis pointing forward and aligned to the velocity vector. The positive *y* axis passes through the right wing and the axis points downward. The aircraft flow angles (angle of attack and sideslip) define the relationship between the wind and the body axis system.

These frames are illustrated in Figure 2.2 but the wind frame. Note that the wind frame is illustrated in section 2.3

Langelaan, Alley, and Neidhofer, in [18], proposed an approach that estimates the wind vector together with the gradients and acceleration. This calculation was considering the

wind as a vector field $\mathbf{w}$ that varies both in space and time, in which its components are expressed in the inertial frame I:

$$\mathbf{w}_I(x,y,z,t) = \begin{bmatrix} w_{Ix}(x,y,z,t) \\ w_{Iy}(x,y,z,t) \\ w_{Iz}(x,y,z,t) \end{bmatrix} \tag{2.1}$$

The field variations directly influence the forces and movements of any given UAV that is flying through it both in space and time. The UAV experience changes of the velocity that can be expressed as a function of the instantaneous wind vector at a point defined by its coordinates (x,y,z):

$$\frac{d}{dt}\mathbf{w}(x,y,z) = \begin{bmatrix} \dot{w}_{Ix} \\ \dot{w}_{Iy} \\ \dot{w}_{Iz} \end{bmatrix} = \begin{bmatrix} \dot{w}^{+}_{Ix} \\ \dot{w}^{+}_{Iy} \\ \dot{w}^{+}_{Iz} \end{bmatrix} + \nabla\mathbf{w} \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{bmatrix} \tag{2.2}$$

where $(\dot{x}_I, \dot{y}_I, \dot{z}_I)$ are the velocity components of the platform in the inertial frame and $\nabla\mathbf{w}$ is the wind gradient of the wind. The components $(\dot{w}^{+}_{Ix}, \dot{w}^{+}_{Iy}\dot{w}^{+}_{Iz})$ denote the time rate of change of wind velocity at the point (x,y,z). The gradient can be divided into two components, one spatial and one temporal. The study of the influence at the wind velocity point changes with the gradient effect, $\nabla\mathbf{w}$, depending on the application of the wind field. This research work focuses on the instantaneous wind velocity, the net wind acceleration, and mostly on the effects of the spatial gradients.

The instantaneous wind velocity at a point can be treated as the sum of two components: the first one is defined as the steady ambient wind, $\mathbf{w}_S$, and the other one represents the temporary variation of the wind velocity $\mathbf{w}_G$, which are typically discrete and continuous gusts, lee-eddies:

$$\mathbf{w} = \mathbf{w}_S + \mathbf{w}_G \tag{2.3}$$

The steady state component is typically expressed in the NED frame:

$$\mathbf{w}_S^{\text{NED}} = \begin{pmatrix} w_{n_s} \\ w_{e_s} \\ w_{d_s} \end{pmatrix} \tag{2.4}$$

2.1.1 Temporary wind disturbances

The temporary wind disturbances are typically expressed in the body frame:

$$\mathbf{w}_G = \begin{pmatrix} u_G \\ v_G \\ w_G \end{pmatrix} \tag{2.5}$$

The temporary wind disturbances, or gusts w_G, are very complex processes. However, the model can be simplified as a frozen Dryden turbulence model. A common representation of this model is the one found in the U.S. DoD military specification MIL-F-8785C [50], which is an approximation of the von Kármán turbulence spectrum given by the Dryden transfer functions in the frequency domain, which is defined in [50]:

$$H_u(s) = \sigma_u \sqrt{\frac{2V_a}{L_u}} \frac{1}{s + \frac{V_a}{L_u}} \tag{2.6a}$$

$$H_v(s) = \sigma_v \sqrt{\frac{3V_a}{L_v}} \frac{s + \frac{v_a}{\sqrt{3}L_v}}{\left(s + \frac{V_a}{L_v}\right)^2} \tag{2.6b}$$

$$H_w(s) = \sigma_w \sqrt{\frac{3V_a}{L_w}} \frac{s + \frac{v_a}{\sqrt{3}L_w}}{\left(s + \frac{V_a}{L_w}\right)^2} \tag{2.6c}$$

in which σ_u, σ_v and σ_w are the intensities of the turbulence along the body frame axes; L_u, L_v and L_w are spacial wavelengths and V_a is the magnitude of the airspeed of the vehicle.

The number of variables in the simplified model shown in (2.6) is vast. Therefore, inferring these values from wind measurements may not be a straightforward process, even with the potential use of an estimator it is difficult to provide with a high level of confidence a sufficient number of variables to characterize the model.

2.1.1.1 Discrete gust

The Discrete gust model is another feature that needs to be identified. The description of this phenomena is the sudden increase of wind speed for a short period of time. The most common model for its representation uses the implementation of the $1 - \cos$ shape

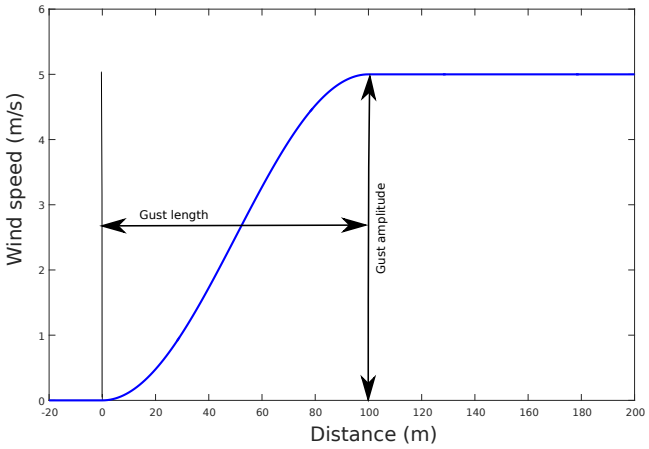


Figure 2.3 Discrete gust profile that shows a growth over the wind on a short period of time from the initial wind speed of the gust magnitude, and a permanent increase at the end of the gust length.

and its mathematical form is as follows:

$$W_{\text{gust}} = \begin{cases} 0 & x < 0 \\ \frac{W_m}{2}(1 - \cos\frac{\pi x}{d_m}) & 0 \leq x \leq d_m \\ W_m & x > d_m \end{cases} \tag{2.7}$$

where W_m is the magnitude of the gust and d_m is the gust length and x is the distance traveled. The discrete gust profile, as represented by the model in (2.7) is illustrated in Figure 2.3.

2.1.2 Spatial wind gradient

The time gradient has influence in all the extension of the PBL, however, the lower part has a significant influence on the spatial wind gradient. In particular, the dominant wind feature is the one that refers to the planar wind speed variation with height, which is the wind shear. This pattern occurs as the wind speed decelerates close to the ground due to the friction [11]. Nevertheless, the wind shear has slightly different behaviors if it occurs in the earth's surface or in the ocean. To begin with the analysis, the earth's surface case is considered, so that the wind shear can be modeled by the following equation:

$$w_{\text{x}} = W_{20}\frac{\ln\frac{h}{z_0}}{\ln\frac{6.096}{z_0}}, 1\,\text{m} < h < 300\,\text{m} \tag{2.8}$$

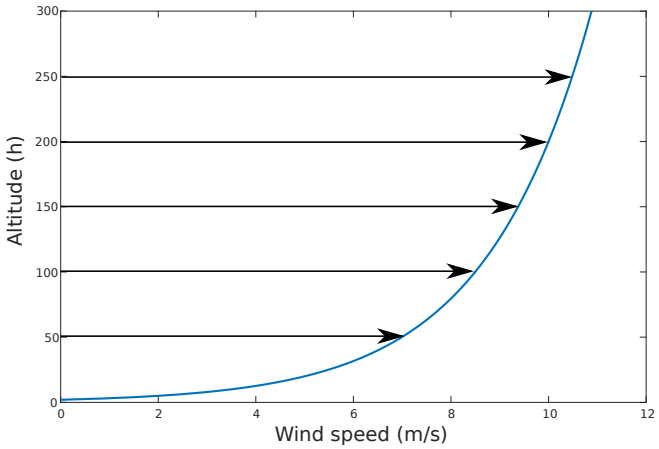


Figure 2.4 Typical shear profile that shows the increase of wind speed over as the altitude increases. There is an exponential relationship between the two variables..

where w_x is the mean wind speed, W_{20} is the wind speed at 20 ft (6.096 m), and z_0 is the roughness length or transition altitude, which varies depending on the flight phase and the terrain if it is close to the ground. A value of 0.0457 m (0.15 ft) is selected, as the intended flights of the UAS's missions are not higher than 300 m and it is a typical value utilized in aviation for final approach phase of commercial aircraft as stated in [50]. Finally, h is the actual altitude of the vehicle. The graphical representation of the wind shear phenomena is shown in Figure 2.4.

The model shown in (2.8) is utilized for the characterization of the feature. However, the model is not sufficient to understand completely the dynamic of the wind shear since it does not consider the influence of the ocean on it, which is necessary for this research work. Zhao [43] and Holton [51] show a generalized form of the horizontal wind gradient (wind shear), which is applicable to the dynamic soaring case:

$$w_{\text{shear}} = \frac{w_{\text{shear}}^{\max}}{z_0}\left[Ah + \frac{1-A_\sigma}{z_0}h^2\right] \tag{2.9}$$

where $w_{x_{\max}}$ is the maximum horizontal wind speed and A_σ is a coefficient that determines the slope shape of the shear, in which it is required that $0 < A_\sigma < 2$; if $0 < A_\sigma < 1$ the shape will be exponential and if it is $1 < A < 2$ the wind profile will be logarithmic and if $A_\sigma = 1$ the wind profile becomes a straight line.

The roughness length z_0 may vary depending on the terrain over the mission. In meteorology there are standard values of z_0 (see Table 2.1).

Table 2.1 Roughness class and length for different land cover types .

Roughness class	Roughness length z_0	Land cover types
0	0.0002 m	Water surfaces: seas and lakes
0.5	0.0024 m	Smooth surface in open terrain
1	0.03 m	Open agricultural land without fences and hedges
1.5	0.055 m	Agricultural land with few buildings and isolated hedges up to 8 m high
2	0.1 m	Agricultural land with few buildings and closer hedges up to 8 m high
2.5	0.2 m	Agricultural land with many trees, bushes and plants
3	0.4 m	Towns, villages, agricultural land with many or high hedges, forests and uneven terrain
3.5	0.6 m	Large towns with high buildings
4	1.6 m	Large cities with high buildings and skycrapers

In (2.8), the selected value of $z_0 = 0.0457$ as a baseline for the wind shear study is indeed an accurate value for typical landscapes on UAS's mission, however for oceanic missions, the value need to be adjusted to be very low, which would be translated as a significant increment on the airspeed all over the profile.

The rate of change of the wind velocity with respect of the vertical component z is given by:

$$\frac{dw_{\text{shear}}}{dz} = \begin{cases} \frac{w_{\text{shear}}^{\max}}{z_0} A + 2 * \frac{w_{\text{shear}}^{\max}}{z_0} \frac{1-a}{z_0} z & \text{for} 0 \leq z \leq z_T \\ 0 & \text{for} z > z_T \end{cases} \tag{2.10}$$

where z_T is the altitude where $w_{\text{shear}} = 0$:

$$z_T = \frac{-Az_0}{2(1-A)} \tag{2.11}$$

2.2 Ocean influence on the wind field

The ocean influence on the gradient model or wind field is relevant. The main application of the algorithms developed in this work are intended to be applied into UAS's for maritime

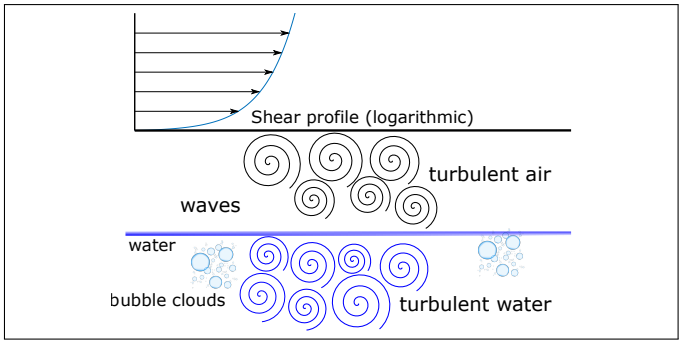


Figure 2.5 Sea-Air interaction in the ABL with shear profile above the turbulent zone.

and coastal applications. The dynamics of the wind are indeed affected by the atmosphere-ocean interface and its interaction have been widely studied by meteorologists [46, 52, 53, 54]. It has been proved that there is correlation between the wind speed measured at an approximate distance of 10 m from the surface of the ocean and the wave spectrum [46]. The repetition of a wave pattern is a very complex problem. Therefore, it is assumed that the useful energy for dynamic soaring is located in the energy spectrum, such that the number of parameters are reduced.

The interaction zone of the ocean and the atmosphere is the ABL. This interaction zone is formed typically by a turbulent zone with direct wave movement and water spray, and a shear profile that is affected in its lower part by this movement as it is shown in Figure 2.5.

The mean velocity profile of the turbulent zone $\bar{W}_u$ is given in the following equation [55]:

$$\bar{W}_u = \frac{\tilde{W}_k}{\chi_k} \log \frac{z}{z_{0,k}} \tag{2.12}$$

In which $\chi_k \approx 0.4$ corresponds to the Von-Kármán constant, $\tilde{W}_k$ is the friction velocity, $z_{0,k}$ is a reference altitude for the friction velocity so that:

$$z_{0,k} = \kappa_c \frac{\bar{W}_u^2}{g} \approx 0.008 \tilde{W}_k^2 \tag{2.13}$$

κ_c corresponds to the Charnock's constant [55]. This value might slightly be increased for coastal areas with very young waves, however, the selected value has given good results in approximations for a very rough sea with old waves.

According to Carter [56], Pierson and Moskowitz have proposed a method for energy spectrum for a fully developed sea, which was characterized based on wind static measure-

ments performed at a altitude of $z_0 = 19.5$ m. The spectrum equation as a function of the angular frequency ω:

$$S_{PM}(\omega) = 8.1 \times 10^{-3} \frac{g^2}{\omega^5} \exp\left(-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^4\right) \tag{2.14}$$

The peak angular frequency ω_p can be expressed in terms of the measurements at an altitude of 19.5 meters:

$$\omega_p = 2\pi f_p \approx 0.8772 \frac{g}{W_{19.5}} \tag{2.15}$$

The amplitude, A, of the wave in the sea surface is derived.

$$A = \sqrt{2\int_0^\infty S_{PM}(\omega) dw} \approx 7.4 \times 10^{-2} \frac{W_{19.5}}{g} \tag{2.16}$$

It is assumed that the ocean behavior is that of a fully developed sea, i. e. steady wind over long period with waves considered as a single sinusoidal wave.

Then, the wave surface height $Z(x,t)$ is

$$Z(x,t) = A\cos\left(kx + \omega_p t\right) \tag{2.17}$$

in which:

$$k = \frac{\omega_p^2}{g} \tag{2.18}$$

Since the wind velocity is dependent on the shape of the wave, an additional reference frame is needed. In such surface-fitted Ω, the curvilinear coordinates (m_Ω, n_Ω) are defined as

$$m_\Omega = x + Ae^{kz}\sin\left(kx - \omega_p t\right) \tag{2.19a}$$

$$n_\Omega = -z - Ae^{kz}\cos\left(kx - \omega_p t\right) \tag{2.19b}$$

The partial derivatives of (m_Ω, n_Ω) are necessary to add them up to the horizontal and wave gradient logarithmic influences

$$\frac{\partial m_\Omega}{\partial x} = 1 + Ake^{kz}\cos\left(kx - \omega_p t\right) \tag{2.20a}$$

$$\frac{\partial n_\Omega}{\partial z} = A\kappa e^{kz}\sin\left(kx - \omega_p t\right) \tag{2.20b}$$

(2.9) should transformed to be expressed with Ω frame coordinates so that the shear profile follows the contour of the wave and considering a logarithmic profile, i.e. $1 < A < 2$,

$$W_{\text{shear}} = \frac{\tilde{W}_k}{\chi}\log\left(\frac{n}{z_0}\right) \tag{2.21}$$

The 3D wind velocity and the wind spatial gradient can be derived from the previous relationships. The wind velocity is expressed in terms of the logarithmic shear effect in the vertical direction W_λ, and in terms of the pump effect due to the wave W_φ. First, considering (2.21) the wind velocities with the added up shear effect are:

$$W_{x_\lambda} = W(u)\frac{\partial n_\Omega}{\partial x} \tag{2.22a}$$

$$W_{z_\lambda} = W(u)\frac{\partial n_\Omega}{\partial z} \tag{2.22b}$$

The velocity perturbations due to the wave pumping are defined as:

$$W_{x_\varphi} = -A\sqrt{g\kappa}\cos\left(\kappa x - \omega_p t\right)e^{-\kappa n} \tag{2.23a}$$

$$W_{z_\varphi} = -A\sqrt{g\kappa}\cos\left(\kappa x - \omega_p t\right)e^{-\kappa n} \tag{2.23b}$$

Resulting in the following 3D wind equations for the boundary layer expressed in the wind frame:

$$W_x = W_{x_\lambda} + W_{x_\varphi} \tag{2.24a}$$

$$W_y = 0 \tag{2.24b}$$

$$W_z = W_{x_\lambda} + W_{x_\varphi} \tag{2.24c}$$

The spatial gradient models shown in (2.8) and the particular case for the oceanic gradient model shown in (2.21) may allow the extrapolation of the measurements to other altitudes. Typically, this is performed based on an empirical power law considering an exponential distribution between two wind measurements at a certain altitude. However, the exponent is subject to the region stability and the roughness, which is addressed in the shear model. In the presence of temporary gradients, i.e. discrete or continuous gusts, the distribution gets wider, however in the long term, it still may be able to approach to that measurement.

2.2.1 Continuous gust estimation

Two methods are proposed for continuous gust identification in both short and long term. Due to the nature of the continuous gust phenomena, there is high variability in the conditions, and not always are suitable for soaring. Therefore, the objective of the continuous gust estimation is to provide with information to determine if the gust is suitable for soaring or to avoid the area. The considered methods should be suitable for real-time implementation. Also, the computational cost have to be relatively low for an on-board implementation.

2.2.1.1 Long term prediction

The first method utilizes a standard Gaussian Process (GP) regression [11] to provide a suitable estimation in the long term. The purpose of this estimation is to determine if there is a potential gust.

Considering a set of vertical wind observations of size $\bar{M}$, $\hat{\mathbf{W}}_{\mathbf{z}} = \hat{W}_{z,i}|_{i=1}^{\bar{M}}$. The wind speed prediction $\bar{W}_p(x)$ at any location x can be expressed as:

$$\bar{W}_p(x) = \sum_{i=1}^{\bar{M}} k_i \hat{W}_{z,i} \tag{2.25}$$

in which k_i is the i-th coefficient of the linear combination of wind measurements $\hat{\mathbf{W}}_{\mathbf{z}}$. Based on the work presented by Park et. al. [57], an optimal coefficient is determined by minimizing the prediction error.

$$\min_{\mathbf{k}} E\left[(\bar{W}_p(x) - W_p(x))^2\right] = \min_{\mathbf{k}} (k^T \left[\mathbf{Q}(\mathbf{X},\mathbf{X}) + \sigma_n^2 \mathbf{I}\right] k - 2k^T \mathbf{q}(\mathbf{X},x) + q(x,x)) \tag{2.26}$$

which can be determined by calculating the covariance matrix $\mathbf{Q}(\mathbf{X},\mathbf{X})$ and the covariance vector $\mathbf{q}(\mathbf{X},x)$ between every two observations at locations $\mathbf{X}$ and x; finally $q(x,x)$

represents the covariance value . This leads to express standard GP regression of the linear predictor as:

$$\bar{p}(X) = \mathbf{k}^T \hat{\mathbf{W}}_\mathbf{z} = \mathbf{q}(x,\mathbf{X}) \left[\mathbf{Q}(\mathbf{X},\mathbf{X}) + \sigma_n^2 \mathbf{I}\right]^{-1} \hat{\mathbf{W}}_\mathbf{z} \tag{2.27}$$

and the covariance value $Q(\bar{p}(x)))$ can be expressed as:

$$Q(\bar{p}(x)) = q(x,x) - \mathbf{q}(x,\mathbf{X}) \left[\mathbf{Q}(\mathbf{X},\mathbf{X}) + \sigma_n^2 \mathbf{I}\right]^{-1} \mathbf{q}(x,\mathbf{X}) \tag{2.28}$$

where σ_n^2 is the measurement noise covariance.

2.2.1.2 Short term prediction

The second approach is used to perform long-term predictions with a non homogeneous regression prediction model. Lerch and Thoraninsdottir [58] have performed a comparison between three non-homogeneous regression model, which allows to produce predictions in day time window. Since the intention of the intended testing flight campaigns is to store data into a single database, a big amount of data can be utilized to perform the predictions with the selected regression model.

At this stage, the truncated normal model was selected as a form of wind estimation. Being W the wind speed and $X_1,\ldots,X_j$ the ensemble member forecasts, the predicted distribution of W can be approximated by a truncated normal distribution:

$$W|X_1,\ldots,X_j \sim \mathbf{N}_{[0,\infty)}(\mu,\sigma^2) \tag{2.29}$$

where the mean μ is an affine function of the ensemble forecast and the variance σ^2 is an affine function of the ensemble variance. If these exchange members are exchangeable [58], the distribution function of the Truncated Normal (TN) distribution $F(z)$ is given by:

$$F(z) = \Phi\left(\frac{\mu}{\sigma}\right)^{-1} \Phi\left(\frac{z-\mu}{\sigma}\right) \tag{2.30}$$

for $z > 0$, where Φ is the cumulative standard normal distribution.

This is indeed a simple non-homogeneous method. However, results indicate that the training period to produce accurate predictions in one-day ahead forecasts is of the equivalent 30 days of continuous measurements [58]. Hence, the goal is to be able to forecasts gusts at least after a month of experimental campaign, which intend to have a very bast amount of data for the prediction models. In addition the long term prediction model goes for measurements that are accumulated in the period of 6 months [58], from which analysis is considered a goal for the end of the experimental campaign. These two

methods will allow to understand the gust formation in a region and also to characterize the region stability which will allow to have a better wind forecast for the soaring applications.

After the understanding of the estimation models, the next step is to have a set of equations of motion that will be utilized in the simulation stage and also to understand certain aspects that are critical to the trajectory generation methods.

2.3 Equations of motion

The understanding of both spatial and temporary gradients are the base for the estimation of the wind field and the identification of features of interest for possible soaring. First, a set of equations of motion that consider the spatial gradient need to be defined, for this, the research work presented by Bowler [46] compared two approaches of equations of motion, one by Sachs [59] and the other by Zhao [43], both in the context of dynamic soaring. The Sachs model has fewer trigonometric terms and it does not require the wind rate to be solved, however it presents a disadvantage when it comes to set the initial conditions in a trajectory optimization problems. In contrast, Zhao's approach considers the use of airspeed, flight path angle and heading angle as state variables, which allows a more intuitive solution. As in [46], the set of equations of motion that will be utilized is the one presented by Zhao.

Consider the aircraft depicted in Fig. 2.6 located at point r with coordinates (x,y,z) in the inertial frame; the acceleration of a point mass can be derived considering acting lift L, drag D, thrust T, and weight mg forces. The wind field velocity components (W_x,W_y,W_z) are defined in the inertial frame. The angle γ_a corresponds to the air,-relative climb and ψ_a is the heading angle of this air-relative motion vector.

First, the vehicle velocities are defined in the inertial frame:

$$\dot{\mathbf{r}} = \mathbf{v}_a + \mathbf{w} \tag{2.31}$$

the components of $\dot{\mathbf{r}}$ are defined as:

$$\dot{x} = V_a \cos\gamma \cos\psi + W_x(x,y,z,t) \tag{2.32a}$$

$$\dot{y} = V_a \cos\gamma \cos\psi + W_y(x,y,z,t) \tag{2.32b}$$

$$\dot{z} = -V_a \sin\gamma + W_z(x,y,z,t) \tag{2.32c}$$

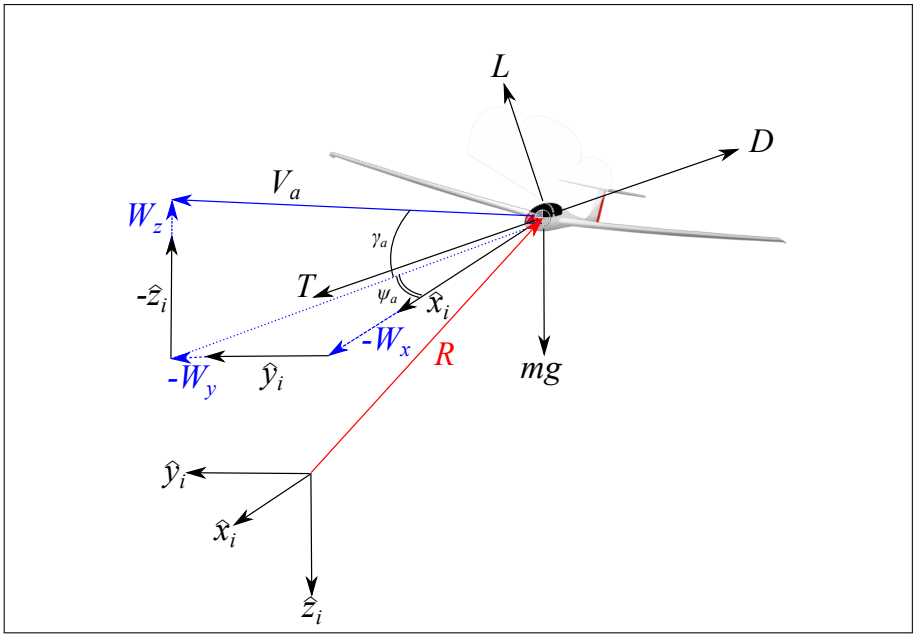


Figure 2.6 Relationship between the inertial and the wind frame, illustrating the most significant angles and the forces acting on the platform.

The differentiation with respect of time of the velocities in (2.32a) - (2.32c) are the body axis accelerations accelerations:

$$\ddot{x} = \dot{V}_a \cos\gamma\cos\psi - \dot{\gamma}V_a \sin\gamma\cos\psi - \dot{\psi}V_a \cos\gamma\sin\psi + \dot{W}_x \tag{2.33a}$$

$$\ddot{y} = \dot{V}_a \cos\gamma\sin\psi - \dot{\gamma}V_a \sin\gamma\cos\psi - \dot{\psi}V_a \cos\gamma\cos\psi + \dot{W}_y \tag{2.33b}$$

$$\ddot{z} = -\dot{V}_a \sin\gamma - \dot{\gamma}\cos\gamma + \dot{W}_z \tag{2.33c}$$

The lift L and drag D in Figure 2.6 are defined as functions of the true airspeed and the aerodynamic coefficients:

$$L = \frac{1}{2}\rho V_a^2 S C_L \tag{2.34}$$

where ρ is the air density, S is the reference area and C_L is the lift coefficient.

The drag polar can be defined as an n-degree polynomial, however, the approach proposed in [46] and [43] is selected due to their previous use in dynamic soaring calculations:

$$D = \frac{1}{2}\rho V_a^2 S(C_{D_0} + C_{D_1}C_L + C_{D_2}C_L^2 + C_{D_3}C_L^3 + C_{D_4}C_L^4) \tag{2.35}$$

in which $C_{D_n}, n = 1 \dots 3$, correspond to the drag coefficients, which can be approximated with a wind tunnel characterization and also could be estimated with a CFD analysis.

With the lift and drag definitions, one can apply the second Newton law to define the sum of forces in the three axis of the inertial frame:

$$m\ddot{x} = (T-D)\cos\gamma\cos\psi - L(\sin\gamma\cos\psi\cos\phi + \sin\psi\sin\phi) \tag{2.36a}$$
$$m\ddot{y} = (T-D)\cos\gamma\sin\psi - L(\sin\gamma\cos\psi\cos\phi - \cos\psi\sin\phi) \tag{2.36b}$$
$$m\ddot{z} = -(T-D)\sin\gamma - L(\cos\gamma\sin\phi) + mg \tag{2.36c}$$

where ϕ is the roll angle.

Now, the wind velocity rate of change in equations (2.33a) - (2.33c) correspond to the spatial and temporal wind gradient. The spatial gradient can be defined in its matrix form $\mathbf{J_w}$ as:

$$\mathbf{J_w} = \begin{bmatrix} \frac{\partial W_{x_\lambda}}{\partial x} + \frac{\partial W_{x_\varphi}}{\partial x} & 0 & \frac{\partial W_{x_\lambda}}{\partial z} + \frac{\partial W_{x_\varphi}}{\partial z} \\ 0 & 0 & 0 \\ \frac{\partial W_{z_\lambda}}{\partial x} + \frac{W_{z_\varphi}}{\partial x} & 0 & \frac{\partial W_{z_\lambda}}{\partial z} + \frac{W_{z_\varphi}}{\partial z} \end{bmatrix} \tag{2.37}$$

The temporal gradient, which correspond to the stochastic portion defined in (2.3), is defined as

$$\mathbf{J}(t)_\mathbf{w} = \begin{bmatrix} \frac{\partial W_{x_\lambda}}{\partial t} + \frac{\partial W_{x_\varphi}}{\partial t} \\ 0 \\ \frac{\partial W_{z_\lambda}}{\partial t} + \frac{\partial W_{z_\varphi}}{\partial t} \end{bmatrix} \tag{2.38}$$

The equations of motion, with the aircraft position at point R, can be then derived in terms of the gradients

$$L = \frac{m}{\cos\phi}\left(V_a\frac{d\gamma_a}{dt} + g\cos\gamma_a - \begin{bmatrix}\cos\psi_a\sin\gamma_a\\ \sin\psi_a\sin\gamma_a\\ \cos\gamma_a\end{bmatrix}^{\mathrm{T}}(\mathbf{J_w}\dot{R} + \mathbf{J}(t)_\mathbf{w})\right) \tag{2.39a}$$

$$\dot{\gamma} = \frac{1}{V_a}\left(\frac{L}{m}\cos\phi - gcos\gamma + \begin{bmatrix}\cos\psi\sin\gamma\\ \sin\psi\sin\gamma\\ \cos\gamma\end{bmatrix}^{\mathrm{T}}(\mathbf{J_w}\dot{R} + \mathbf{J}(t)_\mathbf{w})\right) \tag{2.39b}$$

$$D = \frac{1}{2}\rho V_a^2 S C_D(\alpha) + \frac{L^2}{\frac{1}{2}\rho V_a^2 S\pi\mathrm{Re}} \tag{2.39c}$$

$$\dot{V}_a = \frac{-D}{m} - g\sin\gamma - \begin{bmatrix}\cos\psi\cos\gamma\\ \sin\psi\cos\gamma\\ -\sin\gamma\end{bmatrix}^{\mathrm{T}}(\mathbf{J_w}\dot{R} + \mathbf{J}(t)_\mathbf{w}) \tag{2.39d}$$

$$\dot{\psi} = \frac{1}{V_a\cos\gamma}\left(\begin{bmatrix}\sin\psi\\ -\cos\psi\\ 0\end{bmatrix}^{\mathrm{T}}(\mathbf{J_w}\dot{R} + \mathbf{J}(t)_\mathbf{w})\right) \tag{2.39e}$$

2.4 Wind vector estimation

An initial estimation of the local wind can be provided as in [18] from the set of equations of motion (2.39), in which the wind velocity is computed directly from the sensor information, this is referred as the *direct computation method*. This comes from the relationship of the wind velocity observed in (2.32a) - (2.32c). If the wind terms are solved, it can be obtained:

$$W_x = \dot{y} - V_a\cos\gamma\cos\psi \tag{2.40a}$$

$$W_y = \dot{y} - V_a\cos\gamma\cos\psi \tag{2.40b}$$

$$W_z = \dot{z} + V_a\sin\gamma \tag{2.40c}$$

The airspeed components can be renamed as: $u = V_a\cos\gamma\cos\psi$, $v = V_a\cos\gamma\cos\psi$ and $w = V_a\sin\gamma$. Considering the information provided by an IMU, which corresponds to the body axis accelerations $(\ddot{x}_b,\ddot{y}_b,\ddot{z}_b)$, the angular rates (p,q,r), and the attitude angles pitch θ, roll ϕ, and yaw ψ, and considering a and accelerometer bias $\mathbf{b}$ and with measurement

noise $\mathbf{n}$ the accelerations considering the wind components expressed in the body frame and the airspeed components in the body frame would be:

$$\ddot{x}_b = \dot{W}^b_{x,I} - \dot{u}^b_I + qw - rv + g\sin\theta + \mathbf{b}_x + \mathbf{n}_x \tag{2.41a}$$

$$\ddot{y}_b = \dot{W}^b_{y,I} - \dot{v}^b_I + ru - pw - g\cos\theta\sin\phi + \mathbf{b}_y + \mathbf{n}_y \tag{2.41b}$$

$$\ddot{z}_b = \dot{W}^b_{z,I} + \dot{w}^b_I + pv - qu - g\cos\theta\cos\phi + \mathbf{b}_z + \mathbf{n}_z \tag{2.41c}$$

Since the calculation of the rate of change of the velocity, with respect the air mass, and expressed in the body frame, cannot be determined with the on-board sensors it can be estimated with a second order numerical differentiation. Therefore, the wind speed rate of change at the previous step $k-1$ results in

$$\begin{bmatrix} \dot{w}^b_x \\ \dot{w}^b_y \\ \dot{w}^b_z \end{bmatrix}_{k-1} = \begin{bmatrix} \ddot{x}_b - \mathbf{b}_x \\ \ddot{y}_b - \mathbf{b}_y \\ \ddot{z}_b - \mathbf{b}_z \end{bmatrix}_{k-1} + \begin{bmatrix} -g\sin\theta \\ g\cos\theta\sin\phi \\ g\cos\theta\cos\phi \end{bmatrix}_{k-1} - \begin{bmatrix} qw - rv \\ pw - ru \\ qu - pv \end{bmatrix}_{k-1} - \frac{1}{2\Delta t}\begin{bmatrix} u_k - u_{k-2} \\ v_k - v_{k-2} \\ w_k - w_{k-2} \end{bmatrix} \tag{2.42}$$

The estimation of the wind vector using the information of the IMU and the groundspeed components from the GNSS, allow a direct computation of the vector. The main disadvantage of this estimation method is that the airspeed components expressed in the body frame are necessary to perform the computation. Most of the COTS autopilots provide an approximation of these values. By looking at the main source code of open license low cost autopilots, one can infer that the estimation is poor, as it only considers two axis and not the full rotations. This approximation was introduced by Premerlani [30], in which a body-to-earth DCM is defined with a simple derivation, the wind speed measurements based on the known speed triangle. The source code of the selected autopilot has such implementation as seen in Code 2.1. The open source project allows the possibility of implementing the aforementioned method in the selected platform, which is discussed in chapter 5.

Code 2.1 PX4 Wind Calculation Code.

```
if (_tas_data_ready && (_imu_sample_delayed.time_us -
    _airspeed_sample_delayed.time_us < (uint64_t)5e5)) {
    _state.wind_vel(0) = _state.vel(0) - _airspeed_sample_delayed.
        true_airspeed * cosf(euler_yaw);
```

```
    _state.wind_vel(1) = _state.vel(1) - _airspeed_sample_delayed.
        true_airspeed * sinf(euler_yaw);
  } else {
    // If we don't have an airspeed measurement, then assume the
        wind is zero
    _state.wind_vel(0) = 0.0f;
    _state.wind_vel(1) = 0.0f;
  }
```

An alternative method is proposed by Wenz [26, 28, 27] as part of the MarineUAS project, and Johansen [25], in which an optimal estimator is used for the estimation of wind velocity as a part of an algorithm that aims to estimate the angle of attack α, the sideslip angle β aerodynamic coefficients of the aircraft while flying. The most popular approach is the utilization of an EKF, which drives to some challenges, such as the problem of distinguishing the changes of the aerodynamic coefficients, for instance in an icing episode, and the changes on the wind velocity. The method presented by Wenz evolved from an EKF that needed persistence of attitude excitation for convergence, to a MHE, the architecture of the system adopted in this research, that contains MHE is shown in Figure 2.7, illustrating the interaction for different functionalities.

The initial hypothesis is that the values obtained with the direct computation method, computed with the integration of (2.42), are sufficiently accurate to provide with the necessary information to identify the features that are suitable for energy extraction.

2.5 Wind feature identification

The behavior of the wind is a complex phenomena, which is influenced by different factors that may produce sudden changes that imposes restriction on the modeling. These factors include the terrain (surface stability and surface roughness), the time of the day, time of the year, among others. However, the potential identification of features, such as gusts and the characterization of the shear, will allow the determination of portions of the mission that can be utilized for dynamic soaring. This section describes the algorithms that were proposed and utilized for the identification and characterization of these features, which allow the creation of a live wind map. One of the objectives of the determination of these algorithms is to understand the influence of the ocean in the determination of the adequate conditions for AEH.

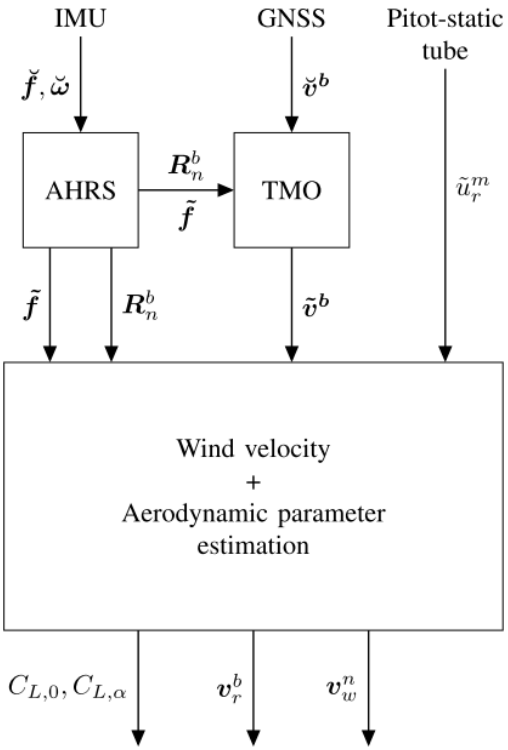


Figure 2.7 The architecture of the moving horizon estimator as presented in [27], with the Attitude and Heading Reference System (AHRS) with inputs of body axis accelerations, f and rotational rates ω; the AHRS determines the accelerations, velocities and computes the position and the rotational matrix from the NED frame to the body frame R_n^b. A Translational Motion Observer (TMO) function is utilized to reduce the noise levels of the groundspeed and the altitude, due to the instability of the vertical channel of the AHRS while computing the vertical velocity. The Pitot tubes inputs the indicated airspeed $\tilde{u}_r^m$ that is used to calculate the true airspeed v_r^b, the aerodynamic coefficients $C_{L,0}$, $C_{L,\alpha}$ and the local wind velocity v_w^n.

2.5.1 Weibull Distribution

Several authors proposed probability density functions for analysis in the study of the wind distribution in the atmosphere. The most common approaches have been the gamma function, the beta function, logistical function, lognormal function, Rayleigh function and Weibull function [60]. Both Weibull and Rayleigh functions are widely used in meteorology [61, 60, 31, 62], specially in wind energy applications. It has been known that the goodness of the estimation depends on the method of determining the parameters

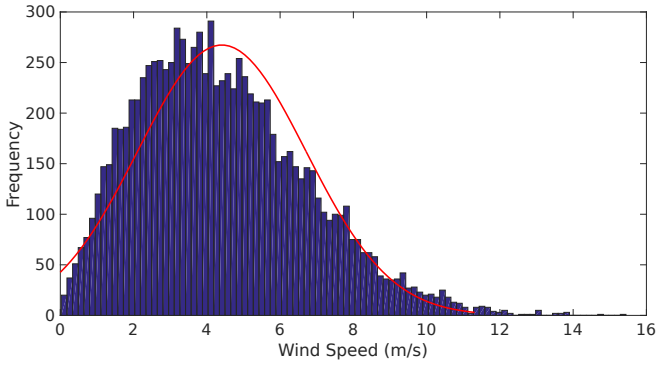


Figure 2.8 Weibull distribution showing the dataset of a National Oceanic and Atmospheric Administration NOAA measurement station at an altitude of 30 m.

of the selected distribution, and there are many studies in which the efficiency of the estimation of the parameters [61]. Both the Weibull and Rayleigh distribution have two parameters that can be adjusted to fit the data set of the wind speed, which may vary due to numerous factors. The Weibull shape parameter, κ, adjust the dispersion of the distribution in such a way that when κ is larger, there is less dispersion with a higher peak value. The scaling parameter, ν, controls the abscissa scale of the distribution. Considering a dataset of wind estimations $\mathbf{W} = (W_1 ... W_n)$, the general form of the Weibull Probability Density Function (PDF) is:

$$f(W) = \frac{\kappa}{\nu}\frac{W}{\nu}^{\kappa-1} e^{-(\frac{W}{\nu})^{\kappa}} \tag{2.43}$$

The Cumulative Density Function (CDF) of the Weibull distribution is:

$$F(W) = 1 - e^{-\frac{W}{\nu}^{\kappa}} \tag{2.44}$$

From this function the most probable wind speed W_{mp} at a particular location can be expressed in terms of the Weibull parameters:

$$W_{\text{mp}} = \nu(1 - \frac{1}{\kappa})^{\frac{1}{\kappa}} \tag{2.45}$$

A typical Weibull distribution is shown in Figure 2.8:

The next step of the feature identification is the determination of the Weibull parameters to be able to predict the wind speed at different altitudes and to be able to identify if the phenomena can be utilized for energy harvesting. The identification of the Weibull parameters can be accomplished by different means, however, the selected approach consist

in the implementation of a GA considering the iterative measurements of the wind speed. Since the wind measurements may not be all the time at the same altitude, one can group the measurements in altitude groups due to the minor changes of the wind speed among small altitudes in typical shear conditions.

2.5.2 Genetic algorithm

GA was proposed originally by Holland [63], and it is an optimization technique inspired in the Darwinian evolutionary theory, in particular to the *survival of the fittest* principle. Given an objective function, also known as fitness function, an initial set of possible solutions are randomly generated. Each possible solution is named chromosome and the total chromosomes form a population of a generation. The chromosomes of this generation are forced to evolve into better chromosomes, i.e, close to an optimal value of the fitness function. The evolution of this chromosomes is done with three operations: the reproduction, the cross-over and the mutation. The computational flow of data is illustrated in Figure 2.9

The **reproduction** or **selection** operators choose the best answers of the initial population and are reproduced based on the results after evaluating the fitness function. The selection of this best solution is achieved either by roulette proportionate selection or to a tournament selection, but also the best solution may be passed directly to the next generation. Furthermore, during the selection process, a small number of values with less fit are selected for the optimization search to keep the diversity in the chromosomes, which will prevent convergence in local solutions with a low fit.

In the **crossover** phase consist in the recombination of portions of the set of solutions with a relatively high fit. The principle of this is that this recombination would create an even better solution. There are different recombination methods, such as the cut and splice crossover or uniform crossover. The selection of this method is utilized to be similar to the chromosome represented the generation.

Finally, the **mutation** aims to keep the diversity in the solution to prevent conversions to local minimum values. The mutation generation ensures the diversity among solutions, meaning that the mutation is the only operator that can solely achieve a semi-optimal solution. The most common mutation method is the bit mutation; however, complex mutation methods can be introduced, in which the values are chosen based on a probability density function.

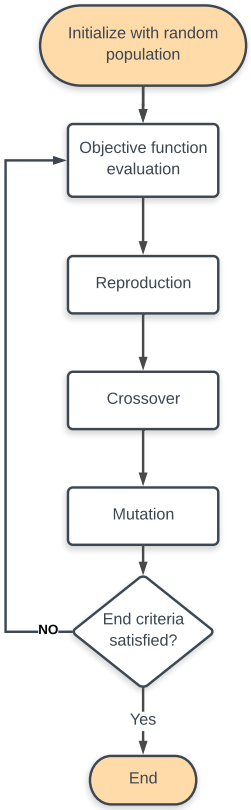


Figure 2.9 GA computational flow diagram.

2.5.2.1 Fitness function for wind estimation

The Weibull shape parameter κ and the scaling parameter ν can be calculated based on the moments of the wind distribution; considering the random variable W, that corresponds to a wind estimate, the mean $E([W]) = \mu$, the variance $E\left[(W-\mu)^2\right] = \sigma^2$ and the standard deviation can be utilized to determine a fitness function as seen in [31]:

$$\mu = \nu\Gamma\left(1+\frac{1}{\kappa}\right) \tag{2.46a}$$

$$\sigma^2 = \nu\left[\Gamma\left(1+\frac{2}{\kappa}\right) - \Gamma^2\left(1+\frac{1}{\kappa}\right)\right] \tag{2.46b}$$

A modified index of dispersion $D(W)$ as the variance-to-squared-mean ratio is defined as:

$$D(W) = \frac{\sigma^2}{\mu^2} \tag{2.47}$$

Combining (2.46) with (2.47) leads to:

$$\frac{\sigma^2}{\mu^2} = \frac{\Gamma(1+2/\kappa)+\Gamma^2(1+1/\kappa)}{\Gamma^2(1+1/\kappa)} \tag{2.48}$$

From here, a residual error ε can be defined as the difference between the wind estimate set $\mathbf{W}$ moments and the right part of (2.48):

$$\varepsilon = \frac{\sigma^2}{\mu^2} - \frac{\Gamma(1+2/\kappa)+\Gamma^2(1+1/\kappa)}{\Gamma^2(1+1/\kappa)} \tag{2.49}$$

Therefore, the objective of the GA is to find a value of κ that, with the given set of measurements $\mathbf{W}$, the residual error converges to a predefined tolerance value. As in [60], this tolerance value was set to $\varepsilon \leq 1 \times 10^{-3}$. The value of the scale parameter *nu* can be found using

2.5.2.2 Error tests for the fitness function

Once a κ value is found, there are two statistical tests that are conducted to the distribution to assess the level of fitness of the estimated data with the distribution. The first test is the Root Mean Square Error (J), that is defined as:

$$\text{RMSE} = \left(\frac{1}{n} \sum_{i=1}^{n} (w_i - w_i^c)^2 \right)^{\frac{1}{2}} \tag{2.50}$$

where, n is the number of wind estimates and w_i is is the value at time i and w_i^c is the correlated value evaluated with the most probable speed, see (2.45).

The second test is the Kolmogorov-Smirnov (KS) test which calculates the maximum error in a CDF:

$$\text{KS} = \max |P(w_{\text{ks}}) - Q(w_{\text{ks}})| \tag{2.51}$$

in which P and Q are the CDF's of the theoretical data and observed estimates in which values do not exceed w_{ks}. After the evaluation of the two tests with (2.50) and (2.51), if both the J and the KS test are kept below a threshold, then the convergence value of κ is accepted, meaning that with a lower residual error, the fit will be better. Note that the

tests could be evaluated with every additional measurement, which could be suitable for low frequency estimations, e.g. with a meteorological station, however, for an unmanned system, the evaluation has to be done a posteriori and at a lower frequency than the estimations, because they may increase the overall computation costs and may impact on the latency of the estimates. Therefore, the tests are performed after every 30 estimates. This value was found during the experimentation phase, ensuring the confidence of the tests.

2.6 Wind estimation and feature identification algorithms

A generic wind estimation and feature identification algorithm is defined to perform in a cyclic approach the wind estimation and the feature identification. Algorithms work in real time from the methods presented previously. The algorithms described in this section describe the processes of wind field estimation with the direct computation method. Then, the same estimation with the use of an EKF, and finally the feature identification process, considering the identification of a shear and the identification of both discrete or continuous gusts.

The purpose of defining the algorithms is to allow the implementation on the functionalities on the system that will extend the flight duration. Computation-wise, the proposed algorithms allow easy implementation. The algorithms cover the wind field estimation (see Algorithm 1) and the features identification process, including the detection of continuous gusts, discrete gusts and the characterization of the wind shear.

Note that the inclusion of a GA to identify wind features certainly adds computational complexity to the algorithm. Since the dispersion index cancels the estimation of the scaling parameter, ν, the shaping parameter, κ, is the one that needs to be optimized in the fitness function (2.49).

2.6.1 Wind estimation with direct computation method

The first part of the wind-related functionalities is the estimation of the wind field, i.e. wind velocity and wind rate of change (line 11 and 12 in Algorithm 1). The computation of the wind velocity is a simple integration of the rate of change that can be obtained by taking consecutive measurements of parameters from the IMU, the pitot probe and the GNSS. In this algorithm, the estimation procedure uses different methods that get the information necessary to compute the wind information. The information is taken in a cyclic manner, ensuring that the state of the elements is correct, meaning, the information is available and the covariance values associated with these parameters are low. Once the

values are received, they are processed to obtain a wind estimation that will be published to be stored in the temporary and permanent database.

Algorithm 1 Wind field estimation with the direct computation method

1: **procedure** WINDESTIMATIONDCM($\hat{a}$,$\hat{r}$,$\hat{u}$,$\hat{\mathbf{b}}$, $\uparrow$ **n**,Attitude, RotationRates, t)
2: **while** INS and GNSS and Pitot is OK **do**
3: $[a_x, a_y, a_z]$ = GetAccelerations();
4: $[\dot{x}, \dot{y}, \dot{z}]$ = GetGroundVelocities();
5: $[u, v, w]$ = GetAirspeedComponents();
6: $[\theta, \rho, \psi]$ = GetAttitudeAngles(); ▷ These angles can be obtained as in (2.40).
7: $[p,q,r]$ = GetRotationRates();
8: $[u, v, w]_{k-2}$ = GetPrevRotationRates(); ▷ Determine at $k-2$ as in (2.42)
9: $[w_x, w_y, w_z]$ = CalculateWind();
10: $[\dot{w}_x, \dot{w}_y, \dot{w}_z]_{k-1}$ = CalculateWindRates(); ▷ The rates are calculated at step $k-1$.
11: PublishWind();
12: PublishWindRates();
13: **end while**
14: **end procedure**

Algorithm 1 depends on the measurements computed by the autopilot, or any external source that calculate the functions (line 11 and 12 in Algorithm 1). The communication of the different variables has to allow the real time implementation. The actual communication means is discussed in Chapter 5. The solution of the navigation is calculated in the autopilot with the use of a Kalman Filter. Therefore, the error at each measurement is contained within the autopilot, and since the relationship of the variables is direct, the same error is propagated to the wind estimate. This means that the error will not grow and if the autopilot provides reliable estimates, then the error in the wind velocity vector estimation is contained. Nevertheless, there are checks of the covariance, which can be extracted on the autopilot as part of the looping conditions of the measurement.

2.6.2 Wind feature identification algorithm

The wind feature identification algorithm is a prediction tool that aims to detect and identify features that are necessary for the energy extraction process. The algorithm performs a statistical test to a group of wind velocity estimates to determine if a feature is present. First, the module requests a wind estimation and stores it into a temporary database. If there are sufficient estimates (at least 30) from the current flight, the system starts a feature detection process by ordering the wind database with respect the UAS altitude. Since the altitude reading varies with time, even in small amounts, the estimates

are grouped according to a reference altitude and they are selected if they are sufficiently close to the reference value. Normally the references altitudes are integer numbers and the groups are conformed by those readings between a tolerance of ± 1 m.

Then, the module should find the corresponding Weibull parameters for each altitude using GA from the grouped wind estimates. If the system finds the Weibull parameters, a most probable wind speed at the corresponding reference altitude is generated. The process is repeated until the local maximum altitude is reached. At first, the system performs an analysis to determine the presence of a shear, which is a very common feature [31]. The wind prediction module tries to characterize the coefficient A_σ in (2.9), that minimizes the error between the most probable wind speeds for the reference altitudes. If the estimates are distributed according to the Weibull distribution and there is a value of A_σ that produces an acceptable error into the system (during the testing, the A_σ coefficient was selected when the average error among the different altitudes was $\varepsilon \leq 5\,\mathrm{m/s}$), an alert is triggered and the system recognizes the presence of a shear.

Afterwards, the system performs a statistical analysis to determine anomalies (significant jumps) in consecutive wind estimates. These were performed by looking for sudden increases into the running standard deviation of the wind estimates. If there is a sudden increase an initial alert is generated because potentially a gust is identified. If the system is not capable to determine accurately the Weibull parameters of the system, most probable wind speeds cannot be fitted into a shear, and/or the running standard deviation presents drastic changes, i.e. there are continuous increases in the running standard deviation, the system assumes the presence of a continuous gust which triggers a short term Gaussian Regression process to characterize the feature.

Algorithm 2 describes the insights of the wind feature detection and identification algorithm. Algorithm 3 defines the procedure of gust detection, Algorithm 4 describes the process of grouping altitudes for statistical tests, and finally Algorithm 5 shows the steps of the Weibull parameter estimation with the use of a GA. The goal is to avoid a big computational impact with the implemented GA algorithm (see Figure 2.9). In this case, the GA implementation is part of the function that calculates the most probable wind speed at a certain altitude described in Algorithm 5, the parameters were carefully selected, based on Liu's approach [31], which prove a low computational impact, since only the shaping parameter κ is estimated. The other parameters necessary to calculate the most probable wind speed are relationships that are not dependent on the GA, which is a very good choice for the real time implementation of the algorithm.

Algorithm 2 Wind feature detection and identification algorithm

```
1: procedure RequestWindEstimation
2:     WindEst=RequestWindEstimate()                    ▷ Request from Algorithm 1.
3:     WDb = StoreinDB(WindEst)
4: end procedure

5: procedure DetectFeature(WDb)                ▷ If database has at least 30 elements
6:     if WDb.Size≥ 30 then
7:         Start = True;                                ▷ Start detection of features.
8:     else
9:         Alert=InsufficienElements
10:    end if
11:    if Start=True then
12:        WDb = OrderAltitudes(WDb);                   ▷ Order WDb based on altitude.
13:        for i = 1 ←AltMax do        ▷ Check for altitudes 1 m to maximum altitude .
14:            NearAlts=FindNearAltitudes(WDb,i,thres);
15:            AdNearAlts=RequestDb(i);                 ▷ Additional WDb elements.
16:            NearAlts=[NearAlts:AdNearAlts];                ▷ Group elements.
17:            WindVelMP=FindMPWVel(NearAlts)   ▷ most probable wind speed at i.
18:            MPS(i)=Store(WindVelMP);       ▷ Store the most probable wind speeds.
19:        end for
20:        A_σ=CalcCoeff(MPS)                  ▷ Calculate A_σ coefficient from (2.9).
21:        if A.Exist = True then
22:            Alert=ShearDetected;
23:        end if
24:        i++;
25:        if Exist(A)=False then
26:            GustDetected = GustDetection()
27:        end if
28:        if GustDetected = True then
29:        end if
30:    end if
31: end procedure
```

There are two functionalities in Algorithm 2 that need to be defined, one is the gust

detection algorithm *GustDetection()*, and the algorithm that groups the altitudes based on an altitude reference *FindNearAltitudes()*, that are shown in Algorithm 3 and Algorithm 4 respectively.

Algorithm 3 Gust Detection algorithm

```
1:  procedure GustDetection
2:      DetectJumps(WDb.Velocity,Std(WDb.Velocity))
3:      if JumpDetected=True then
4:          JumpCounter++;
5:      end if
6:      if JumpCounter≥threshold then
7:          Alert=ContGustDetected
8:      else
9:          Alert=DiscGustDetected;                          ▷ Gust Detected.
10:     end if
11:     if CommHandler.Request.Alert.DiscGustDetected=True then
12:         Gust=DetectJumps.Jumpsize
13:     else if CommHandler.Request.Alert.DiscGustDetected then
14:         ContGust=PerformGaussianRegressionWDb
15:     end if
16: end procedure
```

The algorithm that is used to group the altitudes based on a reference is shown in algorithm 4:

Algorithm 4 Grouping near altitudes algorithm

```
1:  procedure FindNearAltitudes(WDb,alt,thres) ▷ Find altitudes in WDb close to alt.
2:      Counter = 0;
3:      for i = 1 ←WDb.Size do
4:          if alt-thres≤WDb(i).Altitude≤alt+thres then
5:              NearAlts(Counter++)=WDb(i);                  ▷ Store whole WDb.
6:          end if
7:      end forreturn NearAlts;
8:  end procedure
```

The determination of the most probable wind speed at a given altitude is described in algorithm 5 in which ,.

Algorithm 5 Weibull parameter calculation algorithm

procedure FINDMPWVEL(NearAlts) ▷ Find altitudes.
    κ =CalcKappa(NearAlts) ▷ Calculate shaping parameter using GA.
    $v = \frac{1}{\text{Mean(NearAlts)}}\Gamma(1+\frac{1}{\kappa})$ ▷ Calculate scaling parameter from (2.46a).
end procedure

procedure CALCKAPPA(Altitudes) ▷ GA Implementation.
    PopulationSize=50;
    FunctionTolerance=1×10^{-3};
    MaxGenerations=100;
    CrossOverFraction=0.8;
    StdAlt =Std(NearAlts); ▷ Calculate standard deviation.
    MeanAlt=Mean(NearAlts); ▷ Calculate mean.
    PopKappa==rand(PopulationSize); ▷ Initialize with random population.
    while ε >FunctionTolerance **do**
        for $j = 1 \leftarrow$PopKappa.Size **do**
            Results(j)=ObjFunc(PopKappa(j),StdAlt,MeanAlt); ▷ Evaluate Objetive Function.
        end for
        Parents=Selection(Results,PopKappa); ▷ Selection of elements for newGeneration.
        Reproduction(Parents,PopKappa,MaxGenerations;) ▷ Creation of new population.
        Crossover(CrossOverFraction); ▷ Scattered crossover function.
        Migration(); ▷ Gaussian Mutation function.
    end while
end procedure

The aforementioned algorithms are sufficient to provide the necessary functionalities for the detection and identification of features of interest for the dynamic soaring. The detection process can be executed in real time with the presented algorithms, however, they are based on the simplest implementation and do not consider entirely the effects of the interaction of the atmospheric/ocean boundary layer. Nevertheless, this effects may

not impact significantly in the detection process, but they are relevant for the positioning of auxiliary waypoints that will define the soaring trajectories in oceanic operations. The determination of the wave amplitude and frequency on fully developed seas, based on the wind measurements, will allow the determination of the places where the platform may sink optimally for a higher energy gain. The process of trajectory generation and tracking is discussed in the following chapters.

2.7 Chapter summary and conclusions

This chapter presented the different models that were utilized to find a method that extends the flight duration of a UAS. These models were subject of a thorough analysis and were evolved from simple models widely used in aeronautic applications (CS-25/FAR-25 Airworthiness Standard and some Military specifications), to more complex model that consider the interaction of the ocean with the atmosphere. This evolution is key to understand the importance of having accurate wind measurements and how the different phenomena can be utilized correctly to get the maximum energy gain. Also, the impact of the surface accidents, such as mountains, buildings, etc, is discussed, because this factor will determine the ability to extend the wind predictions to wider areas.

In addition, this chapter presented the dynamic model that will be utilized along this work. This model gives big importance to the wind estimation, allowing a full consideration of the impact of the wind estimations into the model. The most important part of this chapter is the estimation methods, from which the *direct computation method* was selected to be implemented due to its simplicity and the proven results in literature. However, other models are discussed, which were developed as part of other research activities of the MarineUAS model that can be implemented to improve the estimation framework.

Furthermore, the methods and algorithms for identification and characterization of features algorithms are presented, including the implementation of the different statistical tests, based on the Weibull distribution. The identification of the Weibull parameters given a dataset was accomplished with the use of a GA and the insights of its implementation was also discussed. The chosen algorithms provide a solid base for the accurate estimation, but also to a real-time implementation which is key to the process of extracting energy.

The main contribution of this chapter is the utilization of the Weibull distribution in real-time applications, because it was utilized commonly for very long datasets for wind turbine placement. In addition, the models that consider the interaction of the wind and the ocean are not widely used, specially in the prediction process, which is also an advantage in the selected approach because not only wind predictions can be performed but also the

wave behavior could be inferred from the wind measurements which would be a great advance for autonomous soaring applications.

The following chapter will discuss the next stage on the development of a framework for flight duration extension: the trajectory generation in which the different steps for generating a wind efficient trajectory are discussed.

3 Energy-efficient trajectory generation

Don't quack like a duck, soar like an eagle.

KEN BLANCHARD

This chapter describes the *trajectory generation* methods that were employed for the energy-efficient trajectory. Firstly, an overview of the different forms of dynamic soaring is presented, with an analysis of the albatross flight, including an energy analysis followed by the details on the implementation of a generator of dynamic soaring trajectories in real time focused in oceanic operations using curve parametrization. The goal of the trajectory generation functionality is to create on a waypoint-to-waypoint basis trajectories that will aim to maximize the efficiency.

3.1 Dynamic soaring

The *dynamic soaring* technique consists in the exploitation of spatial gradients in a wind field, allowing savings in the energy that is utilized to generate lift. This technique can be observed in nature with the albatross (*Diomedea Exulans*). The phenomena was firstly studied in the late XIX and early XX centuries, with the main purpose to replicate its extraordinary ability to soar very long distances in the Southern Ocean without wind flapping [19, 59, 17, 23]. The understanding of the dynamic soaring and the possible large-scale replication has its basis on the observation of the albatross (see Figure 3.1). This birds are biologically designed to fly long distances with a remarkable wing span that goes up to 3.7 m on males and a weight that goes from 7 kg to 10 kg [55].

According to [23], two theories aim to explain the long-distance soaring of the albatross. Although both theories suggest that the albatross exploit spatial gradients, the source of this gradient differs from one to the other. The first one suggests that the albatross utilizes the wind shear, whereas the other indicates that the albatross uses the updrafts formed with

Figure 3.1 The wandering albatross in flight. For a complete illustration of the dynamic soaring an explanatory video can be found in [64] .

the interaction in the air/ocean boundary layer. The wind in the air/ocean boundary layer has a wind shear component and a vertical motion, so the contributions of each effect are hard to be studied. Also, the gusts, lee eddies and rolls (temporary gradients) affect the study of each effect since these phenomena have often wind shear and updrafts. Typically, the study of the dynamics of the albatross has to be simplified [59, 22]; the most common assumption is that the winds are purely horizontal and the absence of waves, therefore, the wind-slope soaring component is excluded. Moreover, the typical wind models consider at most a logarithmic or an exponential gradient model, which does not recognize the wave effects on the gradient and makes it difficult to understand the flight dynamics of the albatross fully. Richardson [23] proposes a model based on the observation of the albatross to estimate the change of energy on the albatross flight and also to create a more accurate model of the wind-shear soaring. This model is the combination of the observations made by Rayleigh [19] that introduced the concept of a bird soaring on the spatial gradient. Also, it is based on the observations of Pennycuick [21], that observe in detail the soaring process of the albatross in the lee-side of the waves, and also in the dynamic model introduced by Lissaman [22] that creates one of the firsts dynamic models applicable to both birds and vehicles but with the limitations mentioned above.

3.1.1 The Rayleigh cycle

The flight pattern that describes the trajectories of the albatross when performing energy extraction is often referred as the Rayleigh cycle [23]. This cycle consists in the utilization of two wind layers and it is often modeled based on the premise that a minimum wind speed is required to perform this maneuvers. The aerodynamic forces studied in Chapter 2, are dependent on the airspeed, however there are studies [22] that offer a comparative of the calculations of the energy while using airspeed or ground speed. There is an increase of airspeed when dynamic soaring is performed, and this increase occurs with the wind-

shear layer cross. However, there is a ground speed increase, which is uncorrelated to the airspeed, that occurs with a heading turn from headwind to downwind. With these statements there is a slight complication in the interpretation of the energy conservation principle. The most popular dynamic equations that offer a solution to this problem were stated by Sachs [59]. The proposed model has been used widely [55, 46] while trying to study and imitate the soaring trajectories. The results show that the albatross needs to get sufficient energy to compensate the drag losses, and this energy comes from both wind-shear soaring and wave-slope soaring.

From the observations performed int he southern hemisphere, concretely in Cape Town Southafrica, Richardson [23] studied albatross soaring in a wind field with gusts that go from $5\,\mathrm{m/s}$ to $10\,\mathrm{m/s}$. The soaring pattern, that lasts in average $10\,\mathrm{s}$ was described as follows:

> Each swoop began with a fast flight parallel to and just above the windward side of a wave. This was followed by a turn into the wind and climb of around 10 m-15 m followed by a downwind descent towards another wave and a turn parallel to the wave. [23]

Richardson developed an schematics of this pattern based on his observations (see Figure 3.2), which is a simplification of the real pattern. For simplification, the waves are shown uniformly, illustrating a fully developed sea, in which the input of energy to the waves from the local wind is in balance with the transfer of energy among the different wave components.

Considering the shear model presented in (2.9), it can be observed that with a logarithmic growth ($1 < A < 2$), the majority of the increase in wind speed occurs in the boundary layer. There are measurements that suggest that the critical wind speed increase is located within 2 m over the sea surface. As it can be seen in Figure 2.1, the structure of this area has additional lee eddy effects and gusts. These lee eddies are utilized by the albatross to perform the dynamic soaring, by entering and leaving the boundary layer. The Lee eddy zone is illustrated in Figure 3.3

Several authors [21, 59, 55] refer the lee eddy dynamic soaring as gust soaring. However, this is not purely gust soaring since the energy gain occurs by transitioning between the lee-eddy zone to over the boundary layer. In the actual gust soaring the gusts are aligned with the lift vector togenerate an energy gain. Another form of dynamic soaring, often referred as hill or slope soaring occurs due to the interaction of the wind and the terrain. The updrafts are produced as the wind circulates over hills and mountain ridges [45]; the actual soaring maneuvers are skewed helical close to the leeward side of the ground. The

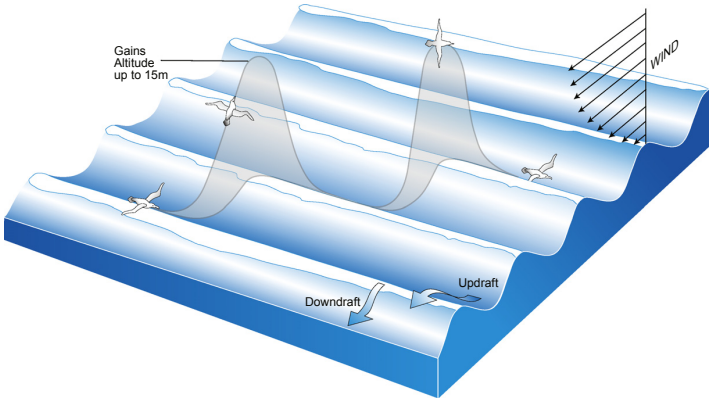


Figure 3.2 Trajectory of the swooping flying pattern of the albatrosses. The waves on the schematics are moving downwind. The phases of the flight are performed in different wind direction: climb is performed upwind and the descent is performed downwind, which is parallell to the wave move. Originally presented in [23].

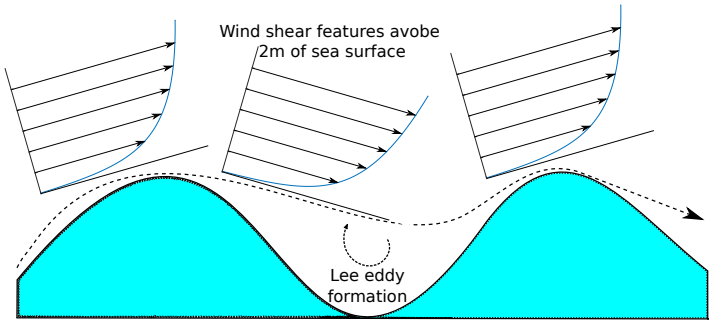


Figure 3.3 Formation of lee eddies in the boundary layer that are utilized by the albatross for dynamic soaring. Once the albatross crosses the boundary layer, indicated by the dotted line, the airspeed increases significantly; this energy gain is used to climb over 10 m.

wind over the top of the windward side is strong compared to the downward side that is decreasing with the slope. This concept is illustrated in Figure 3.4

3.1.2 Swooping maneuver characterization

The swooping maneuvers executed in the Rayleigh cycles are not homogeneous. Depending on the conditions, the albatrosses adjust the maneuver to change the airspeed. These maneuvers should be characterized by simplying the Rayleigh cycles. They can be represented as arc portions, which results in a zigzag flight. The modifications of the climb

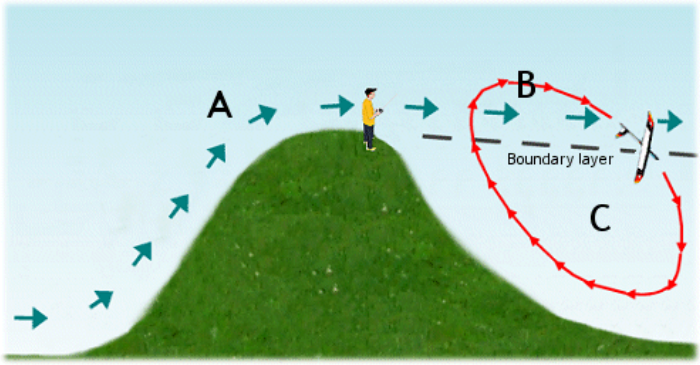


Figure 3.4 Dynamic soaring concept performed in the continental platform, often referred as slope soaring. Typically it is performed on the leeward side of a ridge. Originally presented in [65].

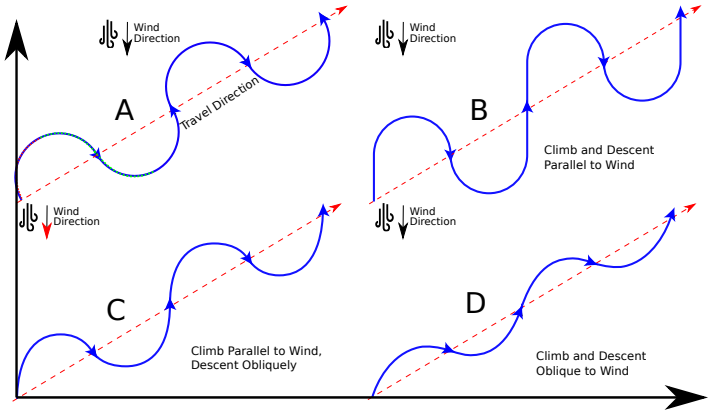


Figure 3.5 Examples of swooping maneuvers represented as united arcs based on the observations of Richardson [23]. (A) Semicircular (perfect) pattern which results in a medium energy gain. (B) Modified pattern so that the climb and descent phases are performed parallel to the wind shear direction for maximum energy gain. (C) Modified pattern for climb parallel to wind but descent parallel to wave crests, which is similar to the albatross flying. (D) Modified pattern for oblique climb and cross wind descent, which can be utilized for reduced energy gains in large wind shear.

and descent patterns across the shear allows energy gains and losses which are useful in low or high wind shear events.

Considering the energy conservation law, the soaring losses have to equal the gains while crossing the shear layer, resulting in:

$$\frac{1}{2}mV_{a_2}^2 = \frac{1}{2}mV_{a_1}^2 \tag{3.1}$$

where V_{a_2} is the airspeed after crossing the wind shear and V_{a_1} is is the airspeed before flying trough the wind shear.

$$\frac{1}{2}m(V_{a_2}^2 - V_{a_1}^2) = 0 \tag{3.2a}$$

$$\frac{1}{2}m(V_{a_2} - V_{a_1})(V_{a_2} + V_{a_1}) = 0 \tag{3.2b}$$

For simplification, the term $(V_{a_2} + V_{a_1})$ can be equal to the double of average airspeed in a circular flight $(V_{a_2} + V_{a_1}) = 2\mu(V_a)$; the term $(V_{a_2} - V_{a_1})$ is the actual airspeed increase ΔV_a.

Considering the rate of change of speed and the drag dissipation, so that:

$$\frac{dV_a}{dt} = \frac{gV_\zeta}{V_a} \tag{3.3}$$

where V_ζ is the sinking speed through the air.

It has been proved that in constant flight and while performing a semi-circular maneuver (see Figure 3.5-A), the decrease of airspeed is linear with time, therefore in half loop $\frac{t}{2}$, the airspeed will be decreased by half. The main component that contributes to the soaring is the vertical one; therefore, considering (3.3) the change of airspeed in the vertical component can be expressed as a function of the period for each cycle τ and the gravity g:

$$\Delta V_a|_z = \frac{g\tau}{2\left(\frac{\mu(V_a)}{V_\zeta}\right)} \tag{3.4}$$

Considering a quadratic drag as in (2.35), the value of the coefficient $\frac{V}{V_z}$ was approximated by Pennywick [21]. He proposed a maximum value of this coefficient of 21.2 for soaring, with an average cruising speed of 16 m/s with the idea of determining the minimum $\Delta V_a|_z$ necessary for soaring. The results of this analysis were as follows:

> The minimum $\Delta V_a|_z$ for Rayleigh cycle was found to be 3.55 m/s with an associated average airspeed $\mu(V_a)$ =16 m/s, average glide ratio $\frac{\mu(V_a)}{Vz} = 13.8$ and an average bank angle of 47.5° in the circle. [23, 24]

This analysis was extended by Richardson in [24], in which the results are extrapolated for any R/C glider, stating firstly the Rayleigh cycle period for a maximum gliding airspeed. This period is also associated to an optimum loop diameter and the results are proved to be accurate even for fast gliding speeds. Note that the cycle is intending to be the neutral dynamic soaring exemplified in Figure 3.5-A. The aforementioned period and loop diameters are given by:

$$\tau|_{\max V_a} = \frac{2\pi\mu^2(V_a)}{gV_a} \tag{3.5}$$

$$d|_{\max V_a} = 2\frac{\mu^2(V_a)}{g} \tag{3.6}$$

In addition, the loading factor η, which refers to the ratio of the lift and the platform weight, in the soaring maneuver can be approximated by the following relationship that includes the centripetal acceleration and gravity.

$$\eta = \frac{1}{\cos\phi} \approx \frac{2\pi V_a}{g\tau} \tag{3.7}$$

The roll angle ϕ of a given period can be calculated at a desired airspeed as a function of the Rayleigh cycle period:

$$\phi = \arccos\left(\sqrt{\frac{1}{\left(\frac{2\pi V_a}{g\tau}\right)^2 + 1}}\right) \tag{3.8}$$

Finally, the minimum wind speed for the a gliding speed in a neutral dynamic soaring can be calculated based on the load factor:

$$W_{\text{ds}} = \frac{\pi V_a}{\max \frac{V}{V_z}} \tag{3.9}$$

The aforementioned parameters (3.5) - (3.8) were considered for very high airspeeds, which are adequate for a specific type of high performance gliders and in a circular flight. However, the cases considered in this research effort are more resemblant to the actual albatross. Despite these considerations, the equations can be used for semi-circular Rayleigh cycles, which will derive into a simplification of the control strategy, as only varying the roll angle direction and a coordinated turn will approximate the desired maneuver, based on a specific period and diameter.

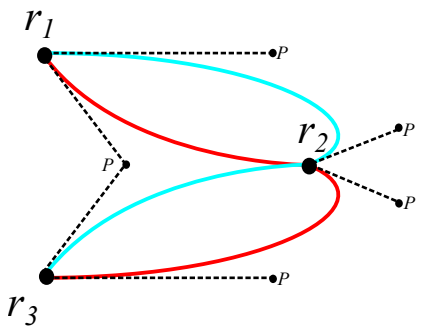


Figure 3.6 Proposed parametric curves, showing a pair of curves merging three waypoints with reciprocate control points. The blue curve shows a curve parametrized with a shaping clockwise direction, and the red one shows a curve with a shaping on the counterclockwise direction.

3.2 Curve selection and parametrization

There are variations of the Rayleigh cycle that needs to be addressed, particularly in heavy winds which are common to geographic areas that are of interest to the present research. Therefore, an additional method is proposed toproduce a parametric curve that will allow the variation of a curve on a waypoint to waypoint basis and also in real time. There are different ways to represent the curves in the Rayleigh Cycle. The selected method is to parametrize a curve that allows the incorporation of kinematic constraints into the curve design toproduce a smooth and flyable curve.

The main objective is to compute parametric curves toperform dynamic soaring. An example of such curve can be observed in Figure 3.6.

It is important to highlight that the desired characteristics of the curve should be defined allowing the modification of parameters to dynamically change between swooping patterns. The selected method was based on the parametrization proposed by Hobby [66], in which the curve has specific parameters of interest and it is computed in linear time, allowing a first order and closely second order geometric continuity. The calculation of such parameters allow the definition of the Bézier control points, which leads to the construction the spline's cubic Bernstein polynomial.

One advantage of the proposed method is that the curves are invariant under scaling, rotation and reflection transformations, and the effect of change of a control point (node) curve decreases exponentially as it gets close to the other node.

Let $\{\mathbf{R}\}$ be a set of waypoints in $\mathbb{R}_3$: $\{\mathbf{R}\} = \{R_1, R_2, \ldots, R_n\}$. A curvy flight path has to be constructed in such way that it passes through every R_i. These are the three main

vaules that should be parametrized:

- Directional slope vectors $\tilde{M}_i$ for every R_i
- Entry and exit velocity functions $S_{\text{in}}, S_{\text{out}}$ for every pair of waypoints R_i, R_{i+1}
- Tension values at every waypoint τ_i

The $\tilde{M}_i$ vectors determine the slope at the entry and exit point of each waypoint so that the direction of the curve at every waypoint can be manipulated freely. Of course this imposes several restrictions on the smoothness level. Therefore, the tension and velocity factors impose geometric restrictions to the direction.

The tension, τ_i, is closely related to the smoothness, which can be measured fundamentally by two means. Either by determining the maximum curvature $k(t)$ of the time, or by determining the maximum rate of change of the curvature over the arc length which is the preferred method:

$$\max_{0\leq t\leq 1} |\frac{dk}{ds}| \tag{3.10}$$

Hobby [66] proposes a pair of velocity functions, S_{in} and S_{out}, to obtain the minimum value of (3.10). This values have to be updated on a waypoint-to-waypoint basis because if these velocity functions are chosen to be the same for all the waypoints in $R(t)$ the smoothness measurements can present local minimums with possible discontinuities.

The proposed velocity functions are:

$$S_{\text{in}}(\xi,\eta) = \frac{2+a}{1+(1-C)\cos\xi + C\cos\eta} \tag{3.11a}$$

$$S_{\text{out}}(\xi,\eta) = \frac{2-a}{1+(1-C)\cos\eta + C\cos\xi} \tag{3.11b}$$

where

$$\xi = \arg \hat{M}_i - \arg\,(R_{i+1} - R_i) \tag{3.12}$$

$$\eta = \arg\,(R_{i+1} - R_i) - \arg \hat{M}_{i+1} \tag{3.13}$$

$$a = A(\sin\xi - B\sin\eta)(\sin\eta - B\sin\xi)(\cos\xi - \cos\eta) \tag{3.14}$$

The chosen initial values for A, B and C constants, which are commonly used for spline generation in computation applications, are [66]: $A = \sqrt{2}$, $B, = 1/16$ and $C = \frac{1}{2}(3-\sqrt{5})$.

As stated before, four Bézier control points are defined in terms of the curve parameters. The initial and final control point correspond to the consecutive respective waypoints of the path R_i and R_{i+1}:

$$P_0 = R_i \tag{3.15}$$

$$P_1 = R_i + \frac{S_{\mathbf{in}}(\xi_i, \eta_{i+1})}{3\tau_i} ||R_{i+1} - R_i|| \hat{m}_i \tag{3.16}$$

$$P_2 = R_{i+1} \tag{3.17}$$

$$P_3 = R_{i+1} \frac{S_{\mathbf{out}}(\xi_i, \eta_{i+1})}{3\tau_{i+1}} ||R_{i+1} - R_i|| \hat{m}_{i+1} \tag{3.18}$$

The proposed polynomial for a cubic interpolation of the control points is defined as:

$$R(t) = K_3 t^3 + K_2 t^2 + K_1 t + K_0; \tag{3.19}$$

where the K_n parameters are defined as functions of the control points P_n:

$$K_0 = P_0 \tag{3.20a}$$

$$K_1 = 3 * (P_1 - P_0) \tag{3.20b}$$

$$K_2 = 3 * (P_0 - 2P_1 + P_2) \tag{3.20c}$$

$$K_3 = -P_0 + 3(P_1 - P_2) + P_3 \tag{3.20d}$$

An example of a generated curve of four waypoints can be observed in Figure 3.7.

3.2.1 Curve generation algorithm

The calculated curves have to be generated on a waypoint-to-waypoint basis allowing the reshaping of the trajectory in accordance with the wind conditions to maximize the energy gain (see Figure 3.7). The generation algorithm employed in the overall trajectory generation and tracking solution considers both the bank angle method, i.e. determining the bank angle for dynamic soaring and perform semicircular trajectories within a given period and diameter for a desired soaring speed, and the usage of hobby splines for complexer cases. Furthermore, the placement of the set of mission waypoints for energy maximization will be studied in the following chapters. In the case of semi-circular Rayleigh cycle Section 3.1.2, the process is straight forward: if the winds allow the possibility of performing stable

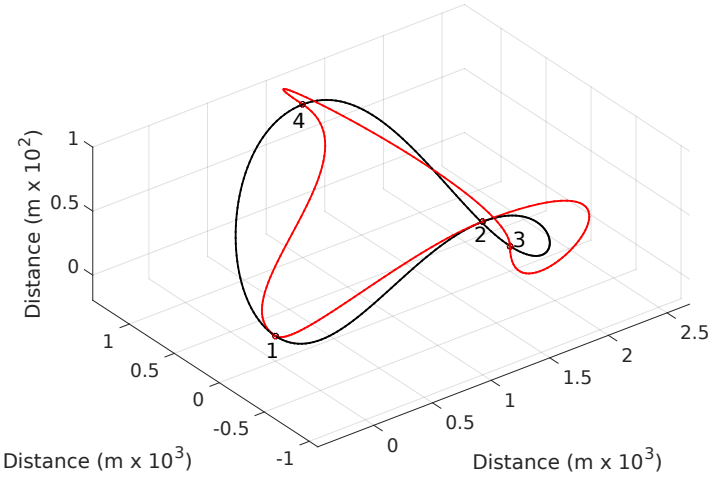


Figure 3.7 Example of parametrized curve showing the effects of modifying parameters. The curve is a closed one, showing lower tension and velocity values in black and stiffer parameters in red.

dynamic soaring, 10knot $\leq W \leq$ 30knot, then the period is calculated together with the optimal radius for the soaring speed that does not exceed the maximum load factor.

If the winds are higher than the ones mentioned above, then between two consecutive waypoints, one or two additional waypoints will be introduced based on the wind speeds and the distance. This additional waypoints will be the transition between the climbing and descent phases of the Rayleigh cycle and will be positioned based on the desired period which can be estimated initially by determining a desired increment on airspeed based on (3.4). However, in the case of a fully developed ocean, the period of the Rayleigh cycle will match the wave period determined in (2.15). The curve velocity functions as in (3.11) are determined so that the maximum curvature does not exceed the semi-circular trajectory which will have the maximum airspeed gain; the tensions τ_i at each waypoint is chosen to be close to 1 for closed Rayleigh cycles and close to 0 for open Rayleigh cycles. The unitary vectors are chosen to follow the desired flight direction between Rayleigh Cycles. A Rayleigh cycle with a parametrized curve can be observed in Figure 3.8.

3.3 Chapter summary and conclusions

This chapter presented a biomimetic approach to replicate the flight of the albatross toextract energy. Firstly an analysis of the albatross flight is presented. These observations provide with different types of trajectory that will result into a different behavior of

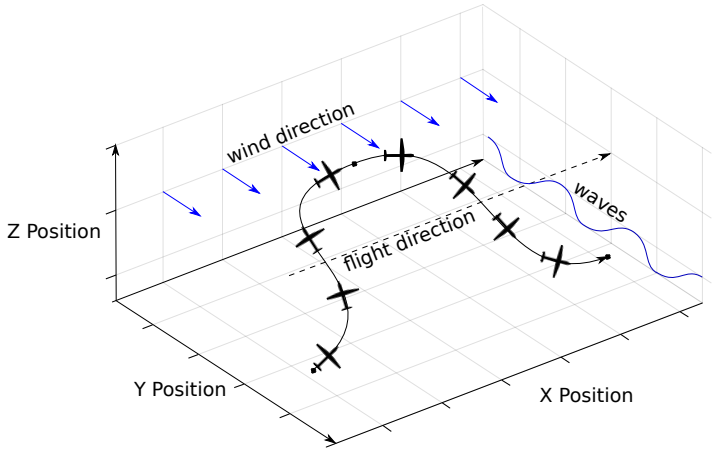


Figure 3.8 Example of a Rayleigh cycle following the method of parameterizing the curves. The desired trajectory is marked in black. The additional waypoint is observed in the middle of the curve and the overall vehicle trajectory follows the dotted flight direction, which is accomplished by selecting the unitary vectors parallel to the desired direction. In this case the tension values were selected to be $\tau = 0.04$ for every waypoint .

the energy gain process. In addition, the necessary conditions for dynamic soaring are discussed toprovide with sufficient information for the trajectory generation.

These trajectories, which are referenced as Raleigh Cycles, are complex curves which are difficult to replicate. Hence, a method that generates a parametric curve that may replicate this trajectory is presented. In other research activities, the dynamic soaring trajectories were considered circles or portions of a circle, which is not always efficient, specially in oceanic conditions. Therefore, the parametric curve offers versatility to produce complex, but smooth curves which really imitate the albatross flight. The parametrization method, offer numerous advantages, because the determination of the control points depends on the desired characteristics of the curve, which could be constrained by the kinematic characteristics of the platform.

Moreover, the need to add auxiliary waypoints into the construction of a Rayleigh cycle was identified and address with an algorithm that permits their determination considering the wind information extracted in the previous chapters, that considers the interaction between the ocean and the atmosphere. Therefore, the waypoints are placed in such way that a portion or a full Rayleigh cycle is constructed but it prioritizes the preexisting waypoints, which is key in the context of coastal and maritime surveillance missions.

The trajectory generation functionality is the key part of the energy extraction process

of the dynamic soaring, and the generation method is a key contribution of this work. The smoothness, which is a factor that depends on the curvature, is such that it is ensured that the tracking of these trajectories will not represent a problem in terms of flyability.

The next chapter discusses the trajectory tracking and the waypoint sequencing reconfiguration, which are the two steps that are performed tomaximize the flight duration without compromising the mission integrity, which most of the time depend on the precise fly-over of the mission waypoints.

4 Wind efficient flight and trajectory tracking

If you have everything under control, you're not moving fast enough.

Mario Andretti

This chapter presents the algorithm employed to perform changes in an initial waypoint sequence to maximize the wind energy gain in a given mission.

In the context of a surveillance mission, the typical approach is to generate a set of waypoints, considering a partition technique to cover a specific area. The waypoints can be positioned based on different approaches to optimize the path in terms of distance, probability of finding a feature of interest, etc. An additional step could be performed to set up the sequence of waypoint in the most efficient manner considering the wind.

The implementation of a re-sequencing algorithm will permit energy gains in straight flight, and to implement the soaring cycles in better conditions by allowing simpler Rayleigh cycles for higher energy gains. Also, it will prevent the execution of Rayleigh cycles in portions of the flight path that will add unnecessary complexity to the mission by determining if the energy gain will be significant.

Initially, a brief analysis on the selected method for complex area coverage is presented, from which the flight plan is generated. Following by the waypoint re-sequencing algorithm. Finally, a methodology for trajectory tracking is proposed and analyzed to complement both the trajectory generation presented in Chapter 3, and the waypoint re-sequencing function.

4.1 Area partitioning technique

The trajectory tracking technique was based on the approach presented by Balampanis et al. in [67], in which a framework for coverage considering complex areas is presented. The

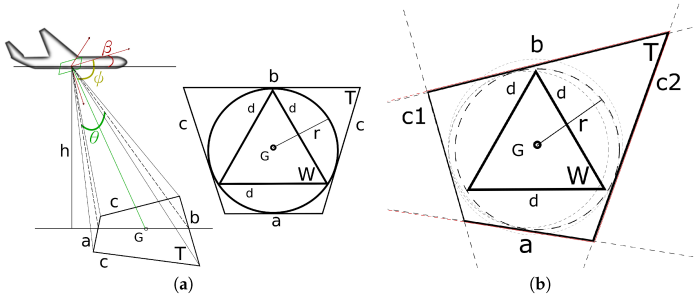


Figure 4.1 The trapezoid represents the field of view of a mounted sensor in the platform, the triangle is utilized as the partition unit of the area, which ensures that at slight changes of attitude as in (a), which change the trapezoid shape, the information will not be lost. This work as presented by Balampanis et. al. [67] as part of the MarineUAS project..

complex areas could include holes (restrictions) and could be in any shape. Besides, the selected methodology allows the partitioning considering multiple vehicles, which could also be utilized to extend the wind feature identification methods presented in chapter 2. In this research, the Constrained Delauney Triangulation (CDT) [68] is utilized as the area partitioning technique, and the Lloyd optimization is used to further improvement of the resulting triangles by improving the angle between cells. The cells are chosen to be triangular because with such form it is assured that each cell can be contained into a sensor footprint, as in any surveillance mission, the typical pattern of a sensor footprint is trapezoidal as it is observed in Figure 4.1.

The CDT introduces edge constraints to create a mesh of triangles. The circumcenter of each triangle is considered a waypoint to provide with full coverage of the area. A cost is given to each triangle to sequence the flight plan, and this cost can be based on different criteria (task, area or agent-based). In the research presented by Balampanis et al. in [67], the waypoint planning problem is a graph search problem of some nodes organized in the CDT. The advantage of this approach is that the region can include complex area restrictions, resulting in even more complex areas and the coverage of the area is still sufficient. For the purpose of this research project, which includes the surveillance of complex maritime/coastal areas, the studied regions considered air traffic restrictions in different locations. For instance, regions such as the great Seattle area (see Figure 4.2), were studied due to the amount of restrictions (urban areas, islands, military operations areas, etc.). These regions were partitioned using CDT. The size of the sensor footprint is variable, and depends on the UAS attitude and altitude; this has a noticeable impact on the cell size. For standardization purposes and also following the current regulations on

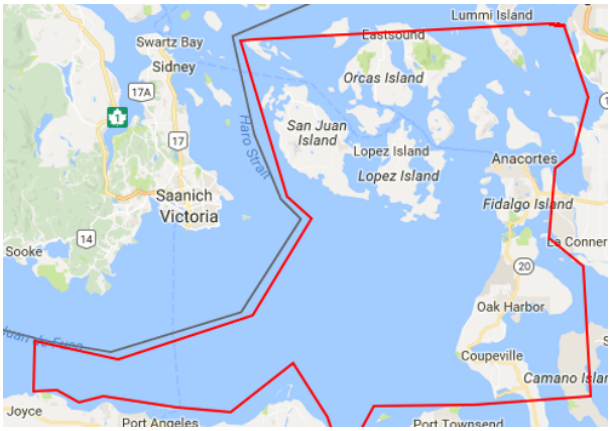


Figure 4.2 A test case area, north of Seattle. The red polygon defines the mission area whereas several other restrictions apply.

Remote-Piloted Aircraft System (RPAS)'s, the cells are typically calculated at an altitude between 100 m and 300 m.

4.1.1 Waypoint sequencing algorithm

After the triangulation, the areas are subdivided based on the number of UAS's. The initial weighting criteria for each cell is the proximity to the border which may result into a spiral path. Furthermore, additional weighting can be provided, such as the one presented by Balampanis et al. in [69], in which the weighting technique is improved for a variation of a pure pursuit tracking method. This was achieved by weighting the resultant angle between any two waypoints which, will then maximize or minimize smoother curves. However, to adjust this method to the requirements of this research, a reconfiguration technique is proposed in which, once the wind features are identified, the sequence is reconfigured with two main objectives:

- To maximize the number of legs in which the platform flies with tail-wind, meaning that there is a significant positive component between the predominant wind direction and the leg direction.
- To identify the potential legs in which a Rayleigh cycle can be performed for energy maximization.

Regarding the first objective, there might be a slight complication with the initial weighting problem: if the main weighting contributor is the comparison between the track and the wind direction, then there might be legs with sharper turns. The second objective

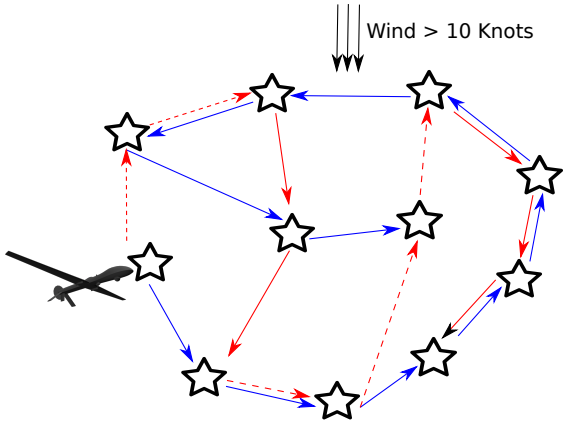


Figure 4.3 An example of leg reconfigurations where the blue lines represent the sequence generated with the CDT algorithm with a normal weighting, and the red lines sequence represent the sequence reconfigured. The solid red lines represent normal leg trajectories, and the dotted lines represent trajectories that will be flown with a Rayleigh cycles since there is head wind.

involves an intricate process; in the Rayleigh cycle description in subsection 3.1.1, there are different cycles that can be performed according to the scenario. There are cycles that may result into a dramatic speed increase, which may produce additional difficulties in the control strategy or in the structural integrity of the vehicle. On the other hand, there are cycles that may reduce the speed even with a significant energy gain. Therefore, an algorithm that considers these restrictions allowing firstly a smart waypoint sequencing, and a selective trajectory generation is needed. The generated curves of the Rayleigh cycles may introduce additional complexity because the energy gain depends very much on how accurate is the tracking therefore, any strategy for creating complex trajectory has to prevent any overload in the control strategy. A graphical example of the re-sequencing algorithm can be observed in Figure 4.3.

The wind-based waypoint sequencing reconfiguration occurs after two events are completed: the first is that there is a valid waypoint sequence, and the second is that there is a sufficient number of wind estimates or sufficient wind information. The availability of wind information imposes the first restriction, that is that the sequencing needs to be performed after the vehicle is airborne, since the main information comes from the wind velocity estimates, however, there are still ways to determine wind direction during planning.

The determination of when the waypoint list can be reconfigured into a wind-based sequence t_0, occurs when the predominant wind speed and direction is identified at the

mission altitude. This can occur by two means, either with the extrapolation of meteorological data from a near meteorological station. This information can be extracted from the National Oceanic and Atmospheric Administration (NOAA), the European Climate Assessment and Dataset (ECAD), which is very useful in case of oceanic missions, and also from the Meteorological Terminal Aviation Routine Weather Report (METAR) reports available. These lasts are valid if the flight is close to a station that emits these reports. With the initial wind direction, the weighting of the cells in the CDT can be adjusted, giving more weight to the neighboring cells that produce a leg that will have a positive component. The weight can be increased or decreased proportionally to the angular difference between the leg and the predominant wind direction. The predominant wind direction can be adjusted on-board, allowing dynamic changes of the sequence.

The process of sequence reconfiguration is illustrated in Algorithm 6.

Algorithm 6 Wind Feature Estimation

```
1: procedure ReconfigureSequence(W, R)   ▷ Depends on the wind and waypoint sequence.
2:     if W is valid then                                   ▷ Wind information is valid.
3:         if R is valid then                               ▷ Valid waypoint sequence
4:             NewSeq = R
5:             for i = 0 to Size(R) do
6:                 N=GetNeighbor(R(i));                   ▷ Get closest waypoints to the current
7:                 for j = 0 to Size(N) do
8:                     A.ang=CalculateAngles;             ▷ Calculate leg angles of neighbors.
9:                     for k = 0 to Size(A) do A.weight(k) = A.ang(k)/W.dir   ▷ Weight the angles according to the wind direction
10:                        if A.weight(k)>Thres then
11:                            M=CheckAngles(A(k));       ▷ Check angles of neighbor waypoints
12:                            M=Max(M)                   ▷ get the maximum angle of the neighbor waypoints
13:                            NewSeq(i+1=R(Index(M)));   ▷ adjust waypoint sequence
14:                            WptReconfFlag(i)=True;
15:                        end if
16:                    end for
17:                end for
18:            end for
19:        end if
20:    end if
21: end procedure
```

The implementation of the reconfiguration process makes a comparison of different

angles between the existing waypoint and the neighbor waypoints. If the angle is close to the wind direction this is stored into a weighting variable that can be utilized in the weighting algorithm presented by Balampanis [67]. However, if other weighting inputs are not to be considered, the next step is to determine the neighboring waypoints with the highest weights, if this are greater than a threshold are taken to the next step which is to determine which one of them will produce the widest angle with respect to the previous one, which is the one that will be then selected as the reconfiguration waypoint.

The determination of where to execute a Rayleigh cycle should be performed when the UAS is airborne. This does not affect the sequence itself but allows a different type of legs between waypoints. Due to the complexity of resultant legs, the process cannot be performed at every leg, so the criteria to select on which legs the Rayleigh cycle will be implemented is based on the wind direction and the shear intensity. The intensity of the shear is calculated with the methods presented in section 2.5, in which the intensity can be inferred once the shear model parameters are identified. The trajectory will be generated when the shear intensity is calculated. Depending on the distance of the actual leg, there will be at least a full Rayleigh cycle. For simplification, the method looks for how many suitable legs for Rayleigh cycles are ahead. For instance, Figure 4.3 presents the distances of the legs that are suitable for Rayleigh cycles, i.e. the dotted dash lines in red, are of different distances. Moreover, the next leg has to be taken into account since its direction will determine the slope unitary vectors, $\tilde{M}_i$ for the resultant curves.

4.1.1.1 Intermediate waypoints and control points

Due to the complexity of the Rayleigh cycle, at least one intermediate or auxiliary waypoint W_a should be added because the altitude at the finalization of the cycle has to be the same as the previous point, otherwise the footprint will be different and the coverage will be affected. This intermediate waypoint or waypoints could be placed closed to the middle point of the leg if it is one, or close to a third of a distance of the leg if there are three, etc. This waypoints will be positioned at the left or right to the straight leg depending on the turning direction of the next leg. If there is no turning in the next waypoint, then the direction will be alternating between the left and right for consecutive legs with Rayleigh cycles. If the distance between waypoints is big enough to perform more than one Rayleigh cycle, then more waypoints will be positioned, allowing the alternation of direction between Rayleigh cycles. This concept can be observed in Figure 4.4. The determination of the altitude of the intermediate waypoint will depend on the identified shear. Basically, the system has to ensure that the aircraft descents up to a point where the gradient is lower than a threshold which will vary on a waypoint to waypoint basis.

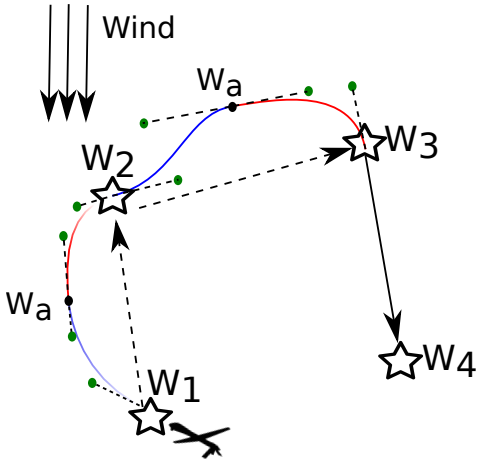


Figure 4.4 The first four waypoints of Figure 4.3 with the computed trajectories. The green points are the control points of the Bezier curves; the control points have to be designed to meet the desired curve velocity and smoothness levels, which may vary according to the weather conditions and also to the desired type of Rayleigh cycles. In this case, the descent phase is illustrated in blue, and the e, and the climbing phase is illustrated in red, note that between these sub-legs there is an intermediate waypoint that marks the transition point.

This change will be proportional to the leg distance, and also to the wind velocity. For high winds, it is recommended a lower descent due to a possible dramatic increase on the airspeed which may produce a structural failure due to the load factor.

The determination of the control points is performed utilizing the methods presented in section 3.2. As it is mentioned before, the tension unitary vectors $\tilde{M}_i$ depend on the direction of the next leg or sub-leg. The tension has be kept low, in values that does not exceed $tau = 0.1$ since a higher value will produce more closed turns that will be hard to track. There are three values that can be modified in the velocity function along the tension, the constants A, B and C, which have to be modified only if the resultant curvature will affect the flight quality. The selected values are the ones proposed by Hobby [66]: $A = \sqrt{2}$, $B, = 1/16$ and $C = \frac{1}{2}(3 - \sqrt{5})$. The modification of the tension will affect the curvature. An example of a computed curve in three dimensions can be observed in Figure 4.5.

To summarize, the system requirements for a wind efficient flight plan are as follows:

- A valid waypoint sequence is necessary. This sequence shall ensure a level of area-coverage based on the footprint size.

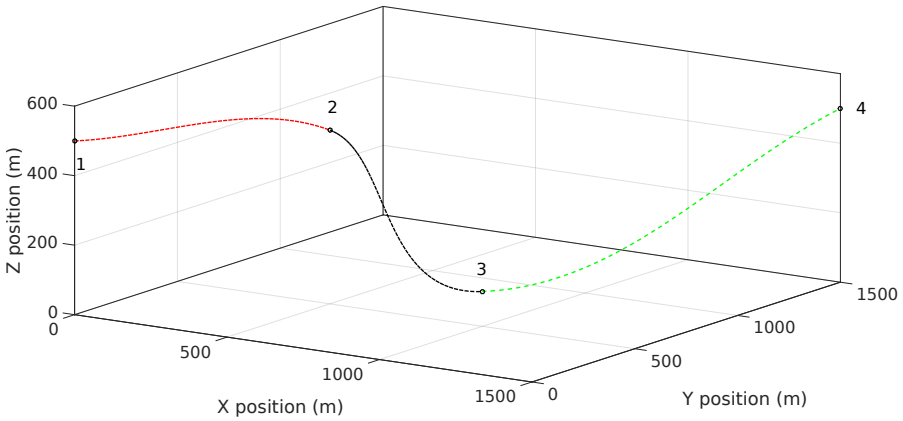


Figure 4.5 Spline generated by the trajectory generation function in which four waypoints are joined following a Rayleigh cycle.

- The planner shall allow the wind information to be considered as a weighting element for the sequence.
- The sequence can be reconfigured only when sufficient wind information is available, at least the predominant wind direction in the area.
- The wind information may come from different sources, but all of them shall be proved to be valid. Weather reports from meteorological stations are acceptable means.
- The selection of legs in which a Rayleigh cycle could be perform shall be independent of weather the sequence was reconfigured with .

4.2 Guidance and Control Strategies

The traditional path tracking is designed based on a internal control loop (control system), which aims to stabilize the vehicle dynamics [70]. Then, with a sort of time-scale separation criteria an external control loop (guidance system) is designed, that transforms the tracking errors into commands for the inner loop. The first problem of this approach relies on the time-scale separation principle which forces the design of the internal control loop with a large bandwidth to accurately track the expected commands from the guidance system. At this point is necessary to differ the trajectory tracking from the path following approaches. The trajectory tracking considers a path which is parametrized by time, so that the guidance system tracks a point while moving through the path. In contrast, on the

path following the vehicle has to stay close to the path without tracking an explicit point [71].

4.2.1 Trajectory tracking guidance method

The selected guidance and tracking methodology that is suitable for the problem presented in this thesis, is the tracking of a parametrized 3-Dimensional spline. Concretely, the method presented by Zhu et al. in [44] and the one presented by Miao et al. in [48]. The first one proposes a two stage method in which the curve is framed with different techniques that are discussed in section 4.3, and the other is the control strategy, in which the dynamic error equations are formulated. The second one provides course and flight path angle control laws with a Lyapunov function based on vector field theory and with error equations based on a moving frame (Serret-Frenet frame).

The Zhu approach considers an evolved Frenet-Serret frame (Type 1 Bishop frame). The Frenet-Serret formulas are fundamental for the development of the curve theory. An independent speed controller (forward channel) is designed to perform the velocity tracking. This is separated from the actual trajectory tracking which might be suitable for the dynamic soaring case, as only the speed needs to be limited to prevent exceeding the maximum turning rates. This is already considered in the trajectory design. However, the observations of dynamic soaring trajectories suggest that with severe wind conditions the speeds are dramatically increased [23]. In the case of lateral and longitudinal errors, this approach permits that the shape of the curve does not influence on the actual tracking performance. Nevertheless, the approach needs to be further developed since there are no direct stability proof. This is solved with the Miao approach, in which an adaptive control strategy is used. In this approach the control strategy is developed in depth and the stability of the system is considered for the design of the control laws by constructing a Lyapunov function that considers vector fields to generate the course inputs to inner-loop attitude control laws [72], which ensures asymptotic stability in the presence of constant wind disturbances. This presents different advantages since the adaptive control method does not depend on the dynamic model of the UAS, which is an advantage in terms of actual implementation, as it is not computational expensive.

One of the key objectives is to consider the trajectory problem independently from the inner control loop, which simplifies the complexity by using differential geometry and also is suitable for the FDES dedicated computer. Therefore, the combination of the framing scheme and the vector field adaptive control method is a very promising solution for the implementation in the FDES guidance and control functionalities and are chosen to analyze and implement them in this thesis.

4.3 Moving Frame Analysis

The formulation of a moving frame is key for the implementation of guidance and control strategies. Following the design philosophy adopted in the previous work, the aim is to present a method for curve framing that is suitable for the generated curves from the FDES trajectory generation function. The concept of a moving frame relies on the fact that the errors are solved in a frame that travels along the curve instead of solving them in the inertial frame, avoiding complex coordinate transformation between frames. This travel is not relative to the time but to the curvature. First, the following frames are considered: $\{\mathscr{I}\}$ is the inertial frame, $\{\mathscr{B}\}$ the body frame, $\{\mathscr{W}\}$ the wind frame, and $\{\mathscr{F}\}$ the curve frame, as illustrated in Figure 4.5. In differential geometry, the Frenet-Serret frame is the one that has been widely used to characterize the curve in the E^3 euclidean space. The relative parallel vector of the $\{\mathscr{F}\}$ frame are defined by the tangent vector T, and the normal vectors B and N. In terms of the arc length s the torsion, τ, and the curvature, κ, are defined as:

$$\frac{d\tau}{ds} = T \tag{4.1}$$

$$\kappa = \left| \frac{dT}{ds} \right| \tag{4.2}$$

$$\tau = -B' \cdot N \tag{4.3}$$

$$\begin{bmatrix} \frac{dT}{ds} \\ \frac{dN}{ds} \\ \frac{dB}{ds} \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} T \\ N \\ B \end{bmatrix} \tag{4.4}$$

The evolution of the bishop frame is the type 1 Bishop frame [44, 73] which is defined in terms of the tangential vector, T, and two auxiliary normal vectors, B and N, so that:

$$\begin{bmatrix} \frac{dT}{ds} \\ \frac{dN}{ds} \\ \frac{dB}{ds} \end{bmatrix} = \begin{bmatrix} 0 & k_1(s) & k_2(s) \\ -k_1(s)0 & 0 & \\ -k_2(s) & 0 & \end{bmatrix} \begin{bmatrix} T \\ N_1 \\ N_2 \end{bmatrix} \tag{4.5}$$

where k_1 and k_2 are the two curvature parameters of the bishop frame. Therefore, a relationship between the Frenet-Serret frame and the type 1 Bishop frame can be established, which is expressed in terms of the rotation angle, θ_F, that dictates the rotation of

the Bishop normal vectors N_1 and N_2 so that:

$$\begin{bmatrix} T \\ N \\ B \end{bmatrix} = \begin{bmatrix} T \\ \cos\theta_F(s)N_1 + \sin\theta_F(s)N_2 \\ -\sin\theta_F(s)N_1 + \cos\theta_F(s)N_2 \end{bmatrix} \tag{4.6}$$

and

$$\theta_f(s) = \arctan\frac{k_2(s)}{k_1(s)} \tag{4.7}$$

$$\tau(s) = \frac{d\theta(s)}{ds} \tag{4.8}$$

$$\kappa(s) = \sqrt{k_1^2 + k_2^2} \tag{4.9}$$

Note that the type 1 bishop frame is isomorphic to the Frenet-Serret frame since the normal frames rotate with an angle, θ_F, with respect the bishop frame. Let us consider the curve in (3.19), the first derivative over time is:

$$\dot{R}(t) = 3K_3t^2 + 2K_2t + K_1; \tag{4.10}$$

Being the magnitude $|\dot{R}(t)| = \text{norm}(\dot{R}(t))$ from the definition of the Frenet-Serret frame:

$$T(t) = \frac{\dot{R}(t)}{|\dot{R}(t)|} \tag{4.11}$$

Then, the elements of (4.5) will be determined:

$$\frac{dT}{ds} = \frac{dT(t)}{dt}\frac{1}{|\dot{R}(t)|} \tag{4.12}$$

.

This is only sufficient to determine the Normal $N(t)$ and Binormal $B(t)$ axis of the Frenet-Serret frame with respect the time so that:

$$N(t) = \frac{dT}{ds}\left|\frac{dT}{ds}\right|^{-1} \tag{4.13}$$

$$B(t) = T(t) \times N(T) \tag{4.14}$$

The curvature and torsion of the Frenet-Serret frame can also be determined:

$$\kappa = \frac{dT}{ds}\frac{1}{\left|\dot{R}(t)\right|} \tag{4.15}$$

$$\tau = -\frac{1}{\left|\dot{R}(t)\right|}\frac{dB}{dt}\cdot N \tag{4.16}$$

The curve presented in Figure 4.5 is framed utilizing the Frenet-Serret moving frame, the result is shown in Figure 4.6.

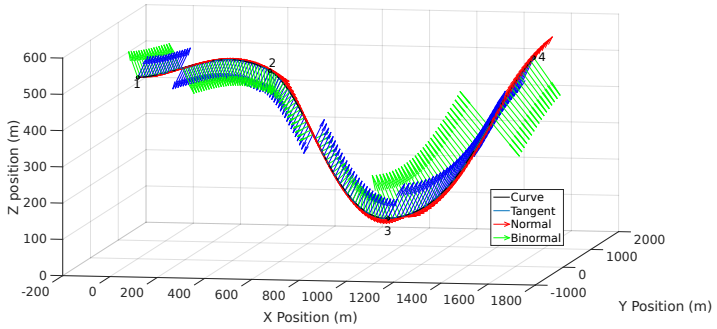


Figure 4.6 The curve presented in Figure 4.5 showing the tangential T, normal N and Binormal B vectors after framing it with the Frenet-Serret frame among the different waypoint segments.

To move to the Bishop frame, the angle θ_F should be determined by using the following equation:

$$\theta_F(t) = \int \tau(t)\left|\dot{R}(t)\right| dt + \theta_0 \tag{4.17}$$

Then, the two curvature parameters of the bishop frame can be determined from (4.7):

$$k_1 = \frac{\kappa}{\sqrt{1+\tan^2\theta_f}} \tag{4.18}$$

$$k_2 = \tan\theta_F k_1 \tag{4.19}$$

The generated curve, framed on the type 1 bishop frame, is shown in Figure 4.7.

The next step is to define and calculate the error between the body frame B and the curve frame F. The error definition, the construction of the Lyapunov function, and the definition of the control laws are based on the work presented in [48].

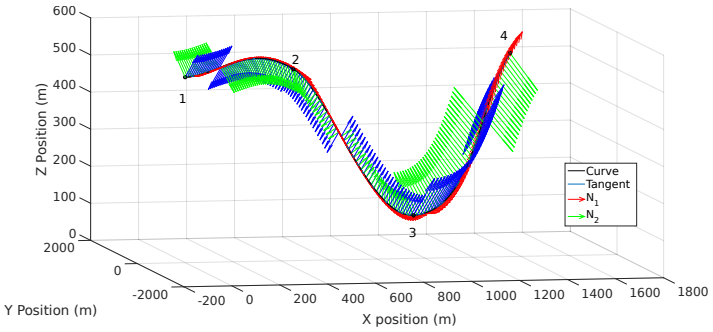


Figure 4.7 The curve presented in Figure 4.5 showing the tangential T, and the normal N_1 and N_2 vectors after framing it with the type 1 Bishop frame among the different waypoint segments.

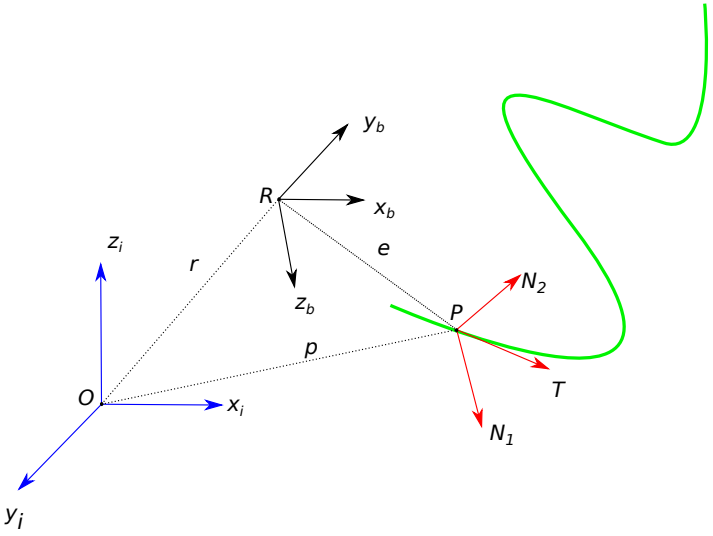


Figure 4.8 Represented error among the curve, the blue frame represents the inertial frame i with origin point at O. The black frame represents the body b frame of the platform with the center of gravity located at R. The red frame represents the frame located at the tracking point P. The **e** distance represents the positional error between the tracking point and the current aircraft position at R.

Consider the frames depicted in Figure 4.8. The goal is to determine the error equations that describe the positional error, e, between the vehicle position at time, t, and a tracking point, P, that will be changing with the aircraft move and also the angular errors between the body and the frame. In the case of Figure 4.8, the curve frame F is a type 1 Bishop frame, showing the N_1, N_2 and tangential T vectors.

Consider a rotation matrix $\mathbf{C}_I^F$ that changes a coordinate set from the I to F frame. Such matrix can be parametrized locally by the angles χ_f and γ_f that represent the attitude at the tracking point (track and flight path angle respectively). This matrix is:

$$\mathbf{C}_I^F(\chi_f,\gamma_f) = \begin{bmatrix} \cos\gamma_f\cos\chi_f & \cos\gamma_f\sin\chi_f & -\sin\gamma_f \\ -\sin\chi_f & \cos\chi_f & 0 \\ \sin\gamma_f\cos\chi_f & \sin\gamma_f\sin\chi_f & \cos\gamma_f \end{bmatrix} \tag{4.20}$$

The rotational rate at point P, ω_f, expressed with respect the inertial frame is defined as

$$\omega_F = \begin{bmatrix} 0 \\ \dot{\gamma}_f \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\gamma_f & 0 & -\sin\gamma_f \\ 0 & 1 & 0 \\ \sin\gamma_f & 0 & \cos\gamma_f \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\chi}_f \end{bmatrix} \tag{4.21}$$

Given a parametrized curve $P(t)$, in the 3-dimensional space, with the equation as in (3.19), the arc length, s, can be defined as a function of the coordinates of a tracking point P, (x_f, y_f, z_f):

$$ds = \sqrt{\left(\frac{dx_f}{dt}\right)^2 + \left(\frac{dy_f}{dt}\right)^2 + \left(\frac{dz_f}{dt}\right)^2}\, dt \tag{4.22}$$

Then, the values of $\dot{\gamma}_f$ and $\dot{\chi}_f$ can be expressed as a function of the arc length $\dot{s}$ of the parametric curve, and

$$\dot{\chi}_f = \frac{\chi_f}{ds}\frac{ds}{dt} = \frac{d\arctan\frac{\dot{x}_f}{\dot{y}_f}}{ds}\dot{s} = \frac{\dot{x}_f\ddot{y}_f - \ddot{x}_f\dot{y}_f}{\left(\dot{x}_f^2+\dot{y}_f^2\right)^{\frac{3}{2}}} \tag{4.23a}$$

$$\dot{\gamma}_f = \frac{\gamma_f}{ds}\frac{ds}{dt} = \frac{d\arctan\frac{\dot{z}_f}{\sqrt{\dot{y}_f^2+\dot{x}_f^2}}}{ds}\dot{s} = \frac{\ddot{z}_f\left(\dot{x}_f^2+\dot{y}_f^2\right) - \dot{z}_f\left(\ddot{y}_f\dot{y}_f+\ddot{x}_f\dot{x}_f\right)}{\sqrt{\dot{y}_f^2+\dot{x}_f^2}\left(\dot{x}_f^2+\dot{y}_f^2+\dot{z}^2\right)}\dot{s} \tag{4.23b}$$

The velocity at the tracking point P can also be expressed in the curve frame F as

$$\left.\frac{dp}{dt}\right|_F = \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} \tag{4.24}$$

The velocity at the point R, expressed in the inertial frame is

$$\left.\frac{dr}{dt}\right|_F = \frac{dp}{dt} + \mathbf{C}_F^I\left(\frac{d\mathbf{e}}{dt}\right)_F + \mathbf{C}_F^I(\omega_F\times\mathbf{e}) \tag{4.25}$$

If (4.25) is multiplied by the rotation matrix defined in (4.20), (4.25) becomes

$$\mathbf{C}_I^F \left(\frac{dr}{dt}\right)_I = \left(\frac{dp}{dt}\right)_F + \frac{d\mathbf{e}}{dt} \tag{4.26}$$

The velocity components at point R are $\frac{dr}{dt} = [\dot{x} \quad \dot{y} \quad \dot{z}]^T$. The error components are defined as $\mathbf{e} = [e_w \quad e_l \quad e_o]^T$, being the forward, lateral and longitudinal errors respectively, and the term $\omega_F \times \mathbf{e}$ can be expanded to

$$\omega_F \times \mathbf{e} = \begin{bmatrix} -\dot{\chi}_f \sin\gamma_f \\ \dot{\gamma}_f \\ \dot{\chi}_f \cos\gamma_f \end{bmatrix} \times \begin{bmatrix} e_w \\ e_l \\ e_o \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_f e_o - \dot{\chi}_f \cos\gamma_f e_l \\ \dot{\chi}_f \cos\gamma_f e_w + \dot{\chi}_f \sin\gamma_f e_o \\ -\dot{\chi}_f \sin\gamma_f e_l - \dot{\gamma}_f e_w \end{bmatrix} \tag{4.27}$$

.

From (4.26), and using the previous definitions, it can be rewritten as

$$\mathbf{C}_I^F \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{s} + \dot{e}_w + \dot{\gamma}_f e_o - \dot{\chi}_f \cos\gamma_f e_l \\ \dot{e}_l + \dot{\chi}_f \cos\gamma_f e_w + \dot{\chi}_f \sin\gamma_f e_o \\ \dot{e}_o - \dot{\chi}_f \sin\gamma_f e_l - \dot{\gamma}_f e_w \end{bmatrix} \tag{4.28}$$

Solving for the error rate of change components yields

$$\dot{e}_w = V_g \cos(\gamma - \gamma_f)\cos(\chi - \chi_f) - \dot{s}\dot{\gamma}_f e_o + \dot{\chi}_f \cos\gamma_f e_l \tag{4.29a}$$

$$\dot{e}_l = V_g \cos(\gamma - \gamma_f)\sin(\chi - \chi_f) - \dot{\chi}_f \cos\gamma_f e_w - \dot{\chi}_f \sin\gamma_f e_o \tag{4.29b}$$

$$\dot{e}_o = V_g cos(\gamma_f - \gamma) + \dot{\chi}_f \sin\gamma_f e_l + \dot{\gamma}_f e_w \tag{4.29c}$$

Note that the course and groundspeed are used because ground-reference measurements produces a better response in the presence of stochastic wind disturbances, such as gusts [72].

The COTS autopilot can implement course and flight path-hold loops, the course and flight path angle dynamics can be expressed as:

$$\dot{\chi} = \varsigma(\chi_c - \chi) \tag{4.30a}$$

$$\dot{\gamma} = \vartheta(\gamma_c - \gamma) \tag{4.30b}$$

where γ_c and χ_c are the commanded flight path and course angles and ς and ϑ are constants that characterize the speed of response of such angles. The angular error is

defined as:

$$\dot{\chi}_e = \dot{\chi}_f - \varsigma(\chi_c - \chi) \tag{4.31a}$$

$$\dot{\gamma}_e = \dot{\gamma}_f - \vartheta(\gamma_c - \gamma) \tag{4.31b}$$

The tracking problem can be formulated from (4.30) and (4.31), in which the goal is to minimize each one of these equations until the values reach a point close to zero.

Considering the development on the vector field based curve tracking presented by Nelson and Miao [72, 48], the flight path angle errors can be expressed as a function of the positional error (4.30):

$$\chi_l(\mathbf{e}_l) = -\chi_\infty \frac{2}{\pi} \arctan(\kappa \mathbf{e}_l) \tag{4.32a}$$

$$\gamma_o(\mathbf{e}_o) = -\gamma_\infty \frac{2}{\pi} \arctan(\mu \mathbf{e}_o) \tag{4.32b}$$

where $-\chi_\infty \in \left(0, \frac{\pi}{2}\right]$ and $-\gamma_\infty \in \left(0, \frac{\pi}{2}\right]$ are the maximum absolute desired value of the respective error.

From here, the Lyapunov function $\mathscr{L}$ is proposed with three objectives, one is to minimize the distance error as close as possible to zero, then is to minimize in the same way both the course and the flight path errors.

$$\mathscr{L} = \frac{1}{2\nu}\left(e_w^2 + e_l^2 + e_o^2\right) + \frac{1}{2}(\chi - \chi_l(\mathbf{e}_l))^2 + \frac{1}{2}(\gamma - \gamma_o(\mathbf{e}_o))^2 \tag{4.33}$$

where ν is a weighting factor, that reduces the contribution of the positional error. The three terms are defined for the minimization of the different errors with the aforementioned dynamics.

The time derivative of the Lyapunov function in (4.33) is:

$$\dot{\mathscr{L}} = \frac{1}{\nu}(\dot{e}_w e_w + \dot{e}_l e_l + \dot{e}_o e_o) + (\chi - \chi_l(\mathbf{e}_l))(\dot{\chi} - \dot{\chi}_l(\mathbf{e}_l)) + (\gamma - \gamma_l(\mathbf{e}_o))(\dot{\gamma} - \dot{\gamma}_l(\mathbf{e}_o)) \tag{4.34}$$

Substituting (4.29) into (4.34) yields:

$$\begin{aligned}\dot{\mathscr{L}} = \frac{1}{\nu}\big[&\left(V_g\cos\left(\gamma-\gamma_f\right)\cos\left(\chi-\chi_f\right)-\dot{s}\dot{\gamma}_f e_o+\dot{\chi}_f\cos\gamma_f e_l\right)e_w\\ &+\left(V_g\cos\left(\gamma-\gamma_f\right)\sin\left(\chi-\chi_f\right)-\dot{\chi}_f\cos\gamma_f e_w-\dot{\chi}_f\sin\gamma_f e_o\right)e_l\\ &+\left(V_g cos\left(\gamma_f-\gamma\right)+\dot{\chi}_f\sin\gamma_f e_l+\dot{\gamma}_f e_w\right)e_o\big]\\ &+\left(\chi-\chi_l(\mathbf{e}_l)\right)\left(\dot{\chi}_f-\varsigma\left(\chi_c-\chi\right)-\dot{\chi}_l(\mathbf{e}_l)\right)\\ &+\left(\gamma-\gamma_o(\mathbf{e}_o)\right)\left(\dot{\gamma}_f-\vartheta\left(\gamma_c-\gamma\right)-\dot{\gamma}_o(\mathbf{e}_o)\right)\end{aligned} \tag{4.35}$$

Rearranging the equation (4.35) and solving for γ_c and χ_c, the control laws for flight path angle and track are determined:

$$\begin{aligned}\chi_c(\mathbf{e}_l) =&K_\chi\left(\chi-\chi_l(\mathbf{e}_l)\right)+\chi+\frac{1}{\alpha}\chi_l(\mathbf{e}_l)V_g\cos\gamma\sin\chi\\ &-\frac{1}{\alpha}\frac{e_l V_g\cos\gamma(\sin\chi-\sin\chi_l(\mathbf{e}_l))}{\nu\left(\chi-\chi_l(\mathbf{e}_l)\right)}\end{aligned} \tag{4.36}$$

$$\begin{aligned}\gamma_c =&-K_\gamma(\gamma-\gamma_l(\mathbf{e}_o))+\gamma+\frac{1}{\beta}\dot{\gamma}_o(\mathbf{e}_o)V_g\sin\gamma\\ &-\frac{1}{\beta}\frac{e_o V_g\left(\sin\gamma-\sin\gamma_o(\mathbf{e}_o)\right)}{\nu\left(\gamma-\gamma_o(\mathbf{e}_o)\right)}\end{aligned} \tag{4.37}$$

where K_χ, K_γ and ν, α and β are the coefficient that make the state variables tend to zero. To proof the stability of the control law, substituting the (4.36) and (4.37) into (4.34), the equation can be rewritten as:

$$\dot{\mathscr{L}} = \frac{1}{\nu}e_l V_g\cos\gamma\sin\left(\chi_l(\mathbf{e}_l)\right)-\frac{1}{\nu}e_o V_g\sin\left(\gamma_o(\mathbf{e}_o)\right)-K_\chi\left(\chi-\chi_l(\mathbf{e}_l)\right)^2-K_\gamma(\gamma-\gamma_l(\mathbf{e}_o))^2 \tag{4.38}$$

It is easily observed that when the errors tend to zero, $\chi_d=\chi_f$ and $\gamma_o=\gamma_f$, then $\dot{\mathscr{L}}=0$ and if any of these conditions is not truth then $\dot{\mathscr{L}}<0$. When time tends to infinity, all the errors, including the values of $\gamma_o(\mathbf{e}_o)$ and $\chi_l(\mathbf{e}_l)$, tend to zero.

The most important advantage of this adaptive control method is that it is completely independent of the dynamic model of the UAS, which is very helpful for uncommon platform configurations, such as flying wing, or delta wing UAS's. In addition, the error equations are relatively easy to compute and the implementation will not affect the overall

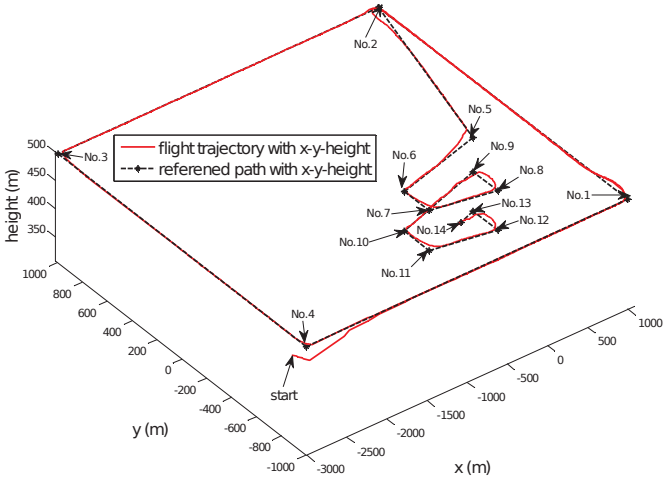


Figure 4.9 As presented in [48], the experiment shows convergence in the path marked in black for a total of 14 waypoints and an offtrack starting position.

computational time of the other functionalities presented. In [48] the algorithm was able to follow accurately a 3D path (see Figure 4.9), however they claim that the system is able to follow a 5th order spline without any experimental proof of it. It is assumed that the main reason of the selected flight path of experiment is the easy implementation of the course and flight path angle of the desired trajectory for the autopilot control. In Chapter 6, the software of the selected autopilot for this research is analyzed since the actual control loop of the groundspeed and course may require additional implementations.

The adaptive method has been proved to work in straight trajectories as in [72], the stability conditions for both the error dynamics and the control laws are analyzed extensively. Furthermore, in [48], the method is proved at a experimental stage for straight 3D paths with great convergence above. There has been attempts to prove the vector field theory approach in more complex situations at an experimental phase. Nelson [72] simplifies the problem by approximate the turns to arcs, however Jung [74] considers a more complex curve for the 2-Dimensional approach. However the results achieved show that with low wind disturbances and with a considerable level of smoothness along the path, the control shall be able to track the splines effectively.

4.4 Chapter summary and conclusions

This chapter discussed two processes. The sequence reconfiguration and the trajectory generation, which are the primary methods for the realization of the flight duration increase.

In the case of the sequence reconfiguration, an algorithm was proposed. This algorithm reconfigures the waypoint sequence, which was previously generated with an area partition method, by modifying the weight of the cells of a partitioned area according to the wind information; this information can be provided to the system either with the use of an external sensor, such as the anemometer, or while flying to perform an online reconfiguration based on the determination of the predominant wind speed. This reconfiguration intends to maximize the legs in which the aircraft flies with a tailwind. This provokes an automatic reduction of the thrust and an energy saving.

The sequencing reconfiguration was incorporated late into the design of the system for flight duration increase. Initially, the goal was to create a full flight plan that uses the trajectory generation algorithm, so that the tracking is performed at every leg of the flight plan. After an analysis of the complexity the tracking algorithm it was decided that this function could be triggered only at specific locations of the flight plan, which are previously analyzed, and reduces the complexity of the tracking functionality.

The trajectory tracking process focuses on the ability of the system to follow the Rayleigh cycles, which is accomplished after a previous selection of legs that are candidates for dynamic soaring. First, a moving frame is added to the curve, based on the Type-1 Bishop frame, that determines the attitude that the aircraft has to follow while flying in a tracking point along the curve. Then, after the framing process, an adaptive control system, based on the vector field theory is implemented. An error calculation is performed in which the positional and angle errors are determined from the current aircraft position with respect the frame at a given tracking point. This error leads to the construction of a Lyapunov function, that determines the control laws that will minimize this function and ultimately follow the desired trajectory.

The main contribution discussed in this chapter is the inclusion of a waypoint sequencing algorithm to loose up the complexity of tracking continuous Rayleigh cycles that may produce positional errors on a given mission. This allows the utilization of a two-stage control strategy that uses the legacy autopilot control for waypoint navigation, which typically uses a classic outer guidance loop and an inner control loop, and also the implementation of the vector-field adaptive control discussed above.

The next chapter discusses the integration of the wind-based waypoint sequencing, and the Rayleigh cycle tracking algorithms with the other functionalities that were discussed

in the previous chapters. Also, the different testing benches are discussed.

5 System integration

Foolproof systems don't take into account the ingenuity of fools.

GENE BROWN

The previous chapters explained the different functions that are necessary to extend the flight duration for oceanic and coastal missions. The main functionalities that are considered for the duration extension are:

- **Wind field identification function**: identifies the wind velocity vector and the wind velocity rate of change at the given location.
- **Wind feature characterization function**: based on the available wind information, i.e. wind estimates and weather information, some features of interest are identified and characterized to determine if the wind field is suitable for AEH.
- **Trajectory generation function**: aims to generate suitable trajectories on a waypoint-to-waypoint basis to perform Rayleigh cycles.
- **Waypoint sequence reconfiguration function**: given a waypoint sequence, some legs are selected in order to perform the Rayleigh cycles, and the sequence is reconfigured for maximum energy gain.
- **Trajectory tracking function**: in the legs that are selected for Rayleigh cycles, the generated trajectory has to be tracked with a robust control scheme for a feasible energy gain.

Based on these functions, an architecture needs to be defined to implement them into a UAS. This architecture has to fulfill a set of requirements for robustness, safety and reliability to be able to provide with real-time flight duration enhancement. This architecture shall meet stringent computational cost, due to the complexity and has to allow upgradeability and maintainability. With this philosophy, this chapter proposes an architecture of a **Flight Duration Enhancement System** (FDES) byfollowing a design methodology of systems engineering that will ensure the correct functionality of the system.

Furthermore, hardware and software requirements should be defined. The high level requirements are derived from the aforementioned functions, which will cascade into lower level requirements to define a system architecture. These requirements are cascaded into lower level requirements which contain specific functions that will be implemented into the hardware and software and further to a system architecture. software selection and hardware architectures, the implementation framework, the selection of components and the selection of data buses have their basis on system requirements that follow best practices of the systems engineering, which is considered the best possible approach for defining the architecture. These are the requirements:

1. The FDES shall allow the computation of its function on board and in real time.
   1.1. The FDES shall provide with the instantaneous wind vector.
      1.1.1. The FDES shall extract the wind information from different sources, including on-board sensors and external sources
   1.2. The FDES shall identify the wind features in the field in which the UAS is flying
   1.3. The FDES shall provide a means for energy gain on the waypoint legs while executing its mission
      1.3.1. The FDES shall compute efficient trajectories that ensure the energy gain
      1.3.2. The FDES shall provide a means for tracking the wind efficient trajectories
2. The FDES shall allow the implementation of other functionalities relative to the function of the UAS. The FDES functionalities are not part of the mission but a means to fulfill the mission with the maximum efficiency
   2.1. The FDES hardware selection shall allow the implementation of other mission related functionalities
      2.1.1. The FDES shall utilize the standard on-board sensors for the calculations among the different functions
      2.1.2. The FDES shall fit on the selected platforms, with a weight that does not exceed 500 g for its components
3. The FDES shall avoid data coupling between the different functionalities
   3.1. The FDES shall use a distributed communication system between functionalities
   3.2. The FDES shall allow the inclusion of different modules to the communications framework
4. The FDES hardware architecture shall maximize the use of COTS components
5. The FDES hardware communication scheme shall utilize standardize protocols
   5.1. The FDES shall allow the interoperability with commercial and open source frameworks (ROS)
6. The FDES shall provide means of validation of the estimated parameters (wind)
7. The validation source shall have an independent source of uncertainty
8. An Air Data Computer (ADC) module shall validate the wind estimations and provide information on the angles of attack and sideslip to validate the results

Different alternatives were explored to define a high level architecture, which will be derived from the requirements to the selection of components and the software implementation. After different attempts, the final architecture was considered to have all the main functionalities of the FDES implemented in a separated module, except from the trajectory tracking which is proposed to reside partially in such module and partially in the tracking module or autopilot. The reason for this is to avoid latency in safety-critical parameters and also to use part of the control strategy implemented in the autopilot. Such architecture is depicted in Figure 5.1. It can be observed that there is an additional ADC module (see Requirement 8.) that has direct interaction with both the FDES and the Autopilot. This module was proposed for validation and optionally to the enhancement of the results by providing another source of estimation of the flight parameters. The architecture depicted in Figure 5.1 is the base to define a software and a hardware architecture. The hardware architecture includes the selection of components, the definition of communication protocols and the means of such communications. The software architecture includes the definition of different functionalities, the interaction between them, and the algorithms to optimize the computational costs. The high-level architecture indicates two communication means, that are typically utilized in applications that require large data storage (Wi-Fi) and a typical radio system utilized in UAS's applications. Both communications have to be resolved in terms of hardware and software due to the additional information elements that have to interact among the functionalities. The following sections will describe the implementations of the high level architecture in terms of software and hardware, and how this selection was performed to fulfill the performance and safety levels required for the duration enhancement process.

5.1 Hardware integration

The Hardware components were chosen based on the aforementioned requirements, in which the main driver is the use of COTS components. Due to the increasing use of UAS's, the COTS components are key to keep the platform costs low. Nevertheless, an important aspect in the selection of the components are that they should be reliable for the given missions. In addition, some COTS components are open source or allow the development of functionalities with common Integrated Development Environment (IDE) software, that will allow the maintainability and the proper documentation of the architecture. The following section describes the important aspects of the hardware components that intervene in the different processes of the FDES. The details and characteristics of the hardware components can be found in Appendix A.

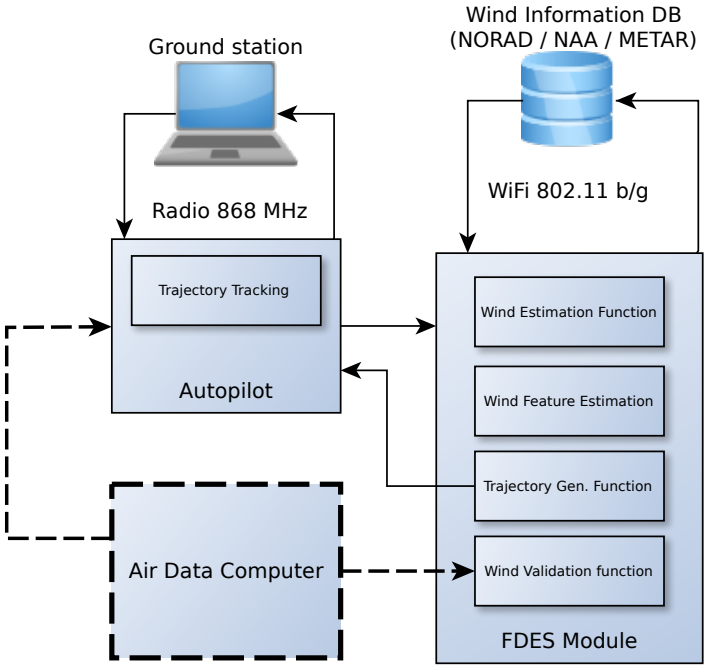


Figure 5.1 High level architecture of the Flight Duration Enhancement System (FDES) where three main modules are appreciated. The autopilot has direct communication with the FDES. Also, the FDES shall have access to a external database that can be utilized to increase the precision in the measurements, and can be updated with external sources. The Autopilot has direct communication with the ground station, and if the FDES provides with information to the user has to go trough the autopilot with any available protocol. Optionally an Air Data Computer (ADC) can be attached to the autopilot and to the FDES module to validate the wind results or to enhance the prediction by providing a more exact wind measurement.

5.1.1 Autopilot and FDES module

The autopilot selection was performed considering hardware and software reliability but also low costs. The selected autopilot is the Pixhawk PX4 [75] (see Figure A.1), which was developed by Lorenz Meier. Pixhawk is not directly an autopilot but a project that enables the implementation of a low cost and highly available autopilot solution into different platforms, such as DroneCode (linux-based) and PX4 (POSIX-based). This system provides the UAS with human-assisted and fully autonomous operations. The main components of the PX4 are enlisted in Table A.1. The PX4 have different communications

means to external sources, it offers the possibility of radio communications towards an S-bus interface, which is compatible with most of the RC devices. Also, it allows radio communications with most popular RC telemetry devices. Finally, the Pixhawk allows serial communications through its Universal asynchronous receiver-transmitter (UART) ports. This communication is fundamental for the FDES, since the intercommunication between modules has to be done through serial communications.

The ODROID-XU4, manufactured by Hardkernel®, was selected as the main computer for the FDES computer. Its versatility and hardware capabilities made them a very reliable choice for the different applications. The Odroid hardware can be observed in Figure A.2 and the relevant characteristics are enlisted in Table A.2. The flash storage allows the possibility of loading different operating systems. In the case of this project, a Linux distribution was utilized as the primary operating system, allowing the implementation of open source development environments, and also the compatibility with most of the systems. The hardware features allow the real time implementation, and the multiple UART buses, together with USB 3.0 enables the possibility of connecting multiple devices, including communication devices, sensors, etc.

5.1.2 Hardware communications

The proper selection of the hardware communications is a crucial part for the FDES, the requirements emphasize the real-time implementation of the different functionalities for extending the flight duration. There are different means for communication, which depend on the available information from each module. The autopilot processes the main flying parameters and allows different types of communication; the radio communications are utilized for both the RC and the telemetry. The UART ports allow serial communications and it is utilized as the link between the ODROID-XU4 (FDES module) and the autopilot. The ODROID-XU4 has a Wi-Fi link that allows the communication with other modules for a multiple-UAS approach and also for the ground station, that may run an additional routine for wind data processing. The ground station has as well a telemetry link, that is possible with the use of a COTS USB-to-UART device, which is common for RC applications. The Wi-Fi link between the ground station and the ODROID-XU4 is a long-range 2.4 GHz, that is widely used for wide-area applications. Finally, a 5.8 GHz Wi-Fi device is utilized for live video transmission. This channel is not utilized actively in the FDES, but it is considered a mission-related device. The RC transmitter-receiver system is a FrSky Taranis 9XD that also transmits in the 2.4 GHz band. The Taranis 9XD is composed by a 16-channel transmitter and, together with the FrSky X8R receiver. Apart from the common RC channels utilized for the aircraft control, and the switching between common

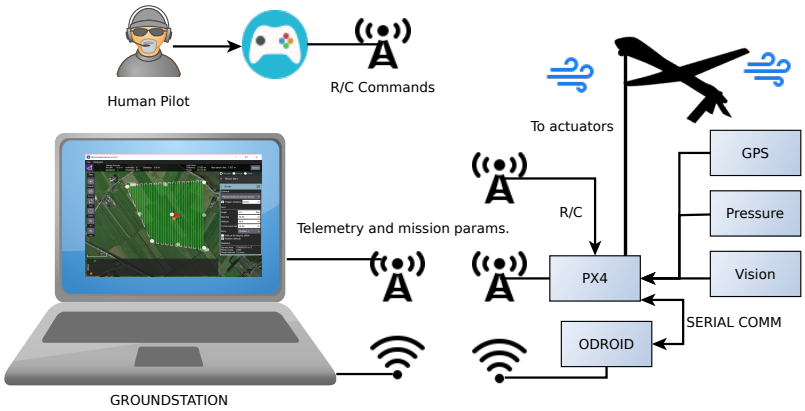


Figure 5.2 Communications framework and communications protocols employed in the FDES, the pilot commands through the RC, and the telemetry utilizes radio communications, the FDES module .

flight modes (AUTO, MANUAL, HOLD), the system allows the inclusion of additional radio modes, which may be related to the energy extraction process. The communication architecture can be observed in Figure 5.2.

5.2 Software integration

The hardware definition provided above provide the basis for the design of the software architecture. The selected architecture is based on a distributed communication framework that will allow upgrades and maintenance with a minimum impact on the system functionality. The communication framework is a topic subscribing and publishing scheme that has the different functionalities as separated software modules. The details on the software versions utilized can be found in Appendix A. Furthermore, the communication framework has a separated functionality that enables the communication with the MAVLINK [76] protocol for direct communication with the autopilot. The MAVLINK protocol was originally released by Lorenz Meier in 2009 with a open-source license. It is a communications protocol for unmanned aerial systems designed as a header-only message marshaling library. The protocol supports serial communications and also has interfaces with other emerging protocols that are becoming popular in the RC community, such as UAVCAN. The Mavlink message frame can be observed in Figure 5.3.

There are different tools that allow the incorporation of MAVLINK messages, into

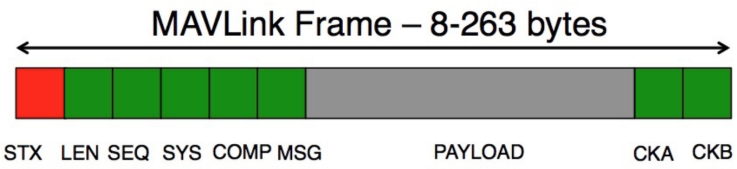


Figure 5.3 STX is the start of frame, following by the length of the payload, the packet sequence, which allows the detection of packet loss. The next are the identifiers of the system, the component and the message, that are defined by the protocol, followed by the payload of the message, which varies on size. The last portion of the frame is the checksum, that checks the entire packet, not only the payload, excluding the start of frame STX.

real-time applications. Although the system was intended to be used for ground station-to-aircraft communications, it can be used for real time applications.

Next, the communication framework architecture is defined. As it is observed in fig. 5.4, the main functionalities reside at the lowest level of the architecture. The selected programming language, C++11, allows the implementation of the functionalities at separated classes, which are to be used as part of the communication framework. Initially, the communication framework was designed to be implemented with sockets, however it was decided to utilize some ROS libraries that allow the easy implementation of a publisher and subscriber scheme, and permits the asynchronous communication between modules. The priorities of the message for the different functionalities have to be established in such way that those that are related to the trajectory tracking or other critical parameters are given the best priority or at least an enhanced

5.2.1 Communication framework implementation

The communication framework was originally designed as a topic publisher/subscriber software with sockets. However, due to the support and available information online it was upgraded to implement this framework with ROS. For this, the ROS version Kinetic Kane was selected since it is compatible with the Linux distribution of the Odroid and has long term support.

The main functionalities that are utilized for the publishing and subscribing scheme are those included in the `ros/ros.h` library that contains the headers necessary to use the most common utilities of the ROS middleware. The routines utilized are those related to the `Nodehandle` class, that is the means to create a handle to publish messages to a topic. The `advertise()` method of this class creates an object of the type `ros::Publisher`, which then can be utilized to call the method `publish()` that is asynchronous, and only

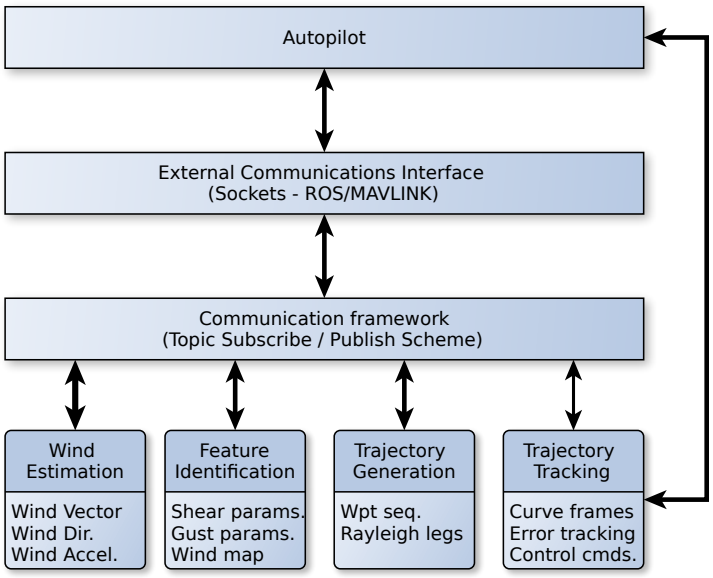


Figure 5.4 The Software Architecture defined, focused on the communication framework scheme. The low level functions publish and subscribe to topics to perform their different routines in an asynchronous manner.

processes the data if there are subscribers connected on the created topic. A simple example of the publishing is presented in Code 5.1.

Code 5.1 Simple publishing example.

```
ros::NodeHandle n;
ros::Publisher pub = n.advertise<std_msgs::String>("name", 5);
std_msgs::StringPtr str(new std_msgs::String);
pub.publish(str);
```

The subscriber method is easier to implement, as the routine `subscribe()`, of the needs to be called either by a a class, a function or a functor object, which leads to a change in the way the method is called.

Code 5.2 Simple subscribing function example.

```
void callback(const std_msgs::StringConstPtr& str)
{
```

```
...
}
...
ros::Subscriber sub = nh.subscribe("my_topic", 1, callback);
```

At each function of the FDES, there are a series of publishers and subscribers that transfer the data from each function to the other. The processes are executed asynchronously, and the trigger of the execution of the functionalities is the existence of the different topics that the routine is subscribing to, e.g. if the wind calculation requires information from the sensors, it waits if the message are available in MAVLINK and performs the calculation and publishes the necessary topic.

Once the communication framework is implemented, the next part is the definition on how the different functionalities were implemented.

5.2.2 Wind field estimation and feature identification

As discussed in Section 2.6, the wind field estimation (wind velocity and wind velocity rate of change) depends on different values that reside in the autopilot. The necessary messages, for the wind field estimation and the feature identification algorithm that the autopilot transmits to the FDES trough the communication framework are enumerated below:

1. **Gyro status** defined in the MAVLINK message `MAV_SYS_STATUS_SENSOR_3D_GYRO` message of the `MAV_SYS_STATUS_SENSOR` type enumeration.
2. **Acceleration status** defined in the MAVLINK message `MAV_SYS_STATUS_SENSOR_3D_ACCEL` of the `MAV_SYS_STATUS_SENSOR` type enumeration.
3. **Accelerations** defined in the `xacc`, `yacc`, `zacc` MAVLINK messages of the `SCALED IMU` type enumeration.
4. **Rotational rates** defined in the `xgyro`, `ygyro`, `zgyro` MAVLINK messages of the `SCALED IMU` type enumeration.
5. **Attitude angles** defined in the `roll`, `pitch`, `yaw` MAVLINK messages of the `ATTITUDE` type enumeration
6. **Groundspeed**, defined in the `vx`, `vy`, and `vz` messages of the `GLOBAL_POSITION_INT` type enumeration.

7. **Covariance matrix**, defined in the `covariance` message of the `GLOBAL_POSITION_INT_COV`, additionally, the velocity covariance in the `velvariance` message of the `CONTROL_SYSTEM_STATE` type enumeration.
8. **Airspeed**, defined in the `airspeed` message of the `CONTROL_SYSTEM_STATE` type enumeration.

Once the wind parameters are identified, the computational functions of the wind functionality of the FDES are implemented following the algorithms of Section 2.6. The result of these functions rewrite the messages of the autopilot that are related to the default wind calculation and publishes new messages in the MAVLINK format related to the identification of the wind.

5.2.3 Trajectory generation and sequence reconfiguration implementation

The trajectory generation and sequence reconfiguration function resides in the FDES module. The implementation requires of a valid waypoint lists and valid wind information. As it is mentioned before, the minimum information necessary to implement this functionality is the horizontal predominant wind direction, this information may be extracted from meteorological reports, if an Internet connection is available. Also, this information can be directly input by the user, or can be obtained with actual measurements of the UAS, since the running mean and standard deviation is stored in the system. If the information is used pre-flight, the system performs the necessary calculations and proposes a new sequence of waypoints from the beginning, and if the information is onboard, then the system proposes a reconfiguration of the remaining waypoints.

In addition, the system generates candidate legs for the implementation of Rayleigh cycles, and sends that leg information to the trajectory generation function that is performed in two stages, one that determines the auxiliary waypoints of that leg, i.e. the control points of the curve, including the generation of auxiliary waypoints to implement a full cycle or a portion of it. The following diagram illustrates the process described above.

5.2.3.1 Curve framing implementation

The curve framing was implemented as a separated module that receives the next Rayleigh leg from the current waypoint. The function performs the framing using the Type 1 Bishop frame, firstly by determining the number of tracking points among the curve which depend on the distance, typically a tracking point will be at approximately every meter of the leg. Once the curve is generated, the tracking points and the unitary vector associated to the frame is stored in a vector type variable. Once the next waypoint

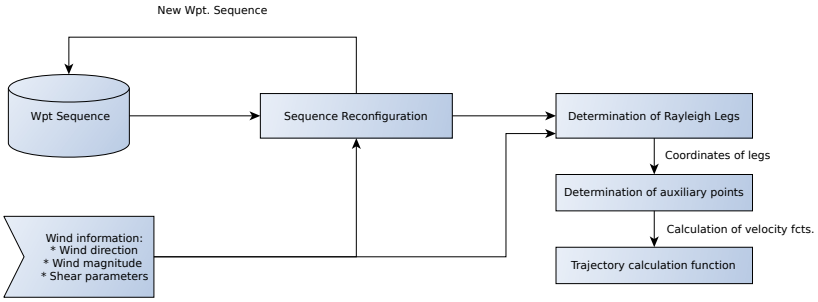


Figure 5.5 Functional diagram of the trajectory generation function.

is the beginning of a Rayleigh leg (`MAV_ROI_WPNEXT` of the `MAV_ROI` type enumeration), the curve framing implementation triggers the tracking function that is described in the following section.

5.2.4 Control Implementation

The control implementation is implemented as a selection function that will enable the tracking of the mission with the autopilot algorithms, which determine the position error and generate the commands using a PD controller. At this stage, the addition of this research is the tuning parameter selection, that is modified based on the wind. Observations on the extensive test campaign helped to determine that the tuning can be changed depending on the wind phenomena, in the case of heavy winds, shear or gusts, the system will lower the gains, and in the presence of calmed winds the system will tighten the gains to ensure less error. The preset values of the tuning are stored as constant values, and the system will choose a preset value based on the environmental conditions.

As it is mentioned in the previous section, if there is a candidate leg to execute a Rayleigh cycle, the auxiliary waypoints and the control points were generated and the curve is framed, then the system will start calculating the position error between the first tracking point. This is the beginning of the Rayleigh cycle. Once this error is minimum, which will depend on the acceptance criteria that is modified based on the environmental conditions, then the control laws will be determined and will be translated into pitch, roll and yaw commands to track the curve. At every step, the system is monitoring the distance to the tracking waypoint and it will determine, based on the heading and the position, if the tracking waypoint is reachable or not. If it is not reachable, the function will switch to the next waypoint, and the process will start again.

To prevent latency with the communications between the FDES the trajectory error determination and the tracking function were implemented as functions of the PX4 firmware, meaning the development framework of the PX4 was utilized [75]. The implemented function receives the tracking waypoint and the Bishop frame unitary vectors, computes the error equations presented in (4.30) and (4.31), and determines the control commands while monitoring for a change of the tracking waypoint which is computed in the FDES module.

5.2.5 Flying modes and control parameters

The PX4 architecture contains different flight modes that provide the user with options on the interaction level between the pilot and the FDES. To fully implement the FDES in a safely manner, two additional modules were added to the preexisting modes. The two added modes are:

- `RAYLEIGH`. This mode enables the reconfiguration of the flight path and the selection of legs for dynamic soaring. The tracking function is fully enabled allowing the dynamic change of the control gains.
- `WIND`. This mode enables the reconfiguration of the flight path only. The tracking function is disabled, but the dynamic change of the control parameters is allowed.

The control parameters that have to be modified for tuning are the *feed forward gain*, the *rate gain* and the trim offsets with the *integrator* gain for the roll, the pitch and optionally the yaw. Furthermore, the time constant of the outer loop, or guidance loop is also adjusted for both the pitch and the roll response. Regarding the waypoint tracking, which is not part of the Rayleigh cycle tracking, the controller is based on an L1 adaptive control technique [77], which has different tuning parameters that can be changed. The most critical parameters are the L1 damping factor the control and the L1 period, which is the distance that defines the tracking point ahead of the aircraft that is following. Finally, the last set of parameters that have to be tuned are those related to the Total Energy Control System (TECS), that aim to control the vertical response of the aircraft when it is climbing, descending or in level flight. The details of how the parameter values for high and low winds were determined can be found in Section 6.3.3.

5.3 Testing benches

The validation and verification is a key objective in the design of the FDES. Therefore, some testing benches need to be carefully designed to validate the different functionalities

at various levels. For this purpose, two main approaches were selected. The first one consists in the creation of a Hardware-In-The-Loop (HITL) architecture, that aims to prove the communications and the reliability of the hardware. It allows the testing of the different flight modes before the actual field experiments, but also to test the behavior of the systems under some potential failures, which is a significant step in the validation phase. The Software-In-The-Loop (SITL) architecture aims to test the different functions of the FDES at various levels and with different methodologies. It is key for testing modifications on the functionalities that are necessary to implement the control routines in the autopilot and the other functions in the FDES module. The software shall allow the testing with a dynamic model but also with available data from other flights, such as the estimation of the wind field, the identification of features, the control responses, etc. This section describes the architecture of both testing benches and discusses the selection of such architectures and the impact on the system validation.

5.3.1 Hardware-In-The-Loop setup

The HITL bench is composed by three main hardware components:

- The autopilot, that has updated with the latest version of the firmware, and has the telemetry module and the RC system attached to it.
- The FDES module, that has a USB-to-UART adapter for autopilot communications, has the ROS.
- The ground-station computer that runs a flying simulation software and the ground-station software that receives the data from the telemetry module.

For the flying simulation software, a commercial license of XPlane-10 was utilized as it is the main supported platform for the PX4 developer team. The communication is achieved with UDP ports that are utilized for the XPlane to transmit the flight parameters. The Simulator runs the dynamic model of the aircraft, that is achieved by compiling the JSBSim library [78], which is is an open source flight dynamics model (FDM). The simulator stablishes direct communication with the ground station, that runs the QGround-Control software [79] and embeds the simulator parameters (position, velocity, etc.) into MAVLINK messages with a unique message type enumeration. The Qgroudncontrol is connected trough a USB-To-UART interface to the autopilot that computes the mission parameters and sends the commands that will be embedded into MAVLINK command messages. The command parameters are sent to the simulator which provides a real-time response and offers the versatility of testing different flight modes. Since the autopilot has

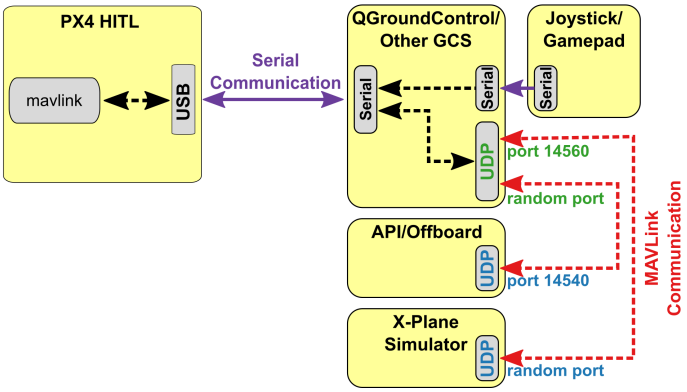


Figure 5.6 The HITL functional architecture, in which the ground station is connected to the flight controller hardware via USB, and acts as a gateway to forward data between the X-Plane simulator running on a development computer, PX4, and any offboard API.

the telemetry module and the RC system attached to it, the bench allows pilot commands from the RC transmitter and also the possibility of getting a whole set of telemetry parameters into another ground station for communication testing. The HITL architecture diagram is shown in Figure 5.6.

5.3.2 Software-In-The-Loop and simulation bench setup

The SITL bench is implemented in two parts. The first part consist on the generation and testing of the different models and functionalities that are utilized in a simple simulation form. This is accomplished with MATLAB®SIMULINK®which allows the creation of complex atmospheric scenarios that will allow the testing of the different functionalities of the FDES before the full implementation on the system. The Simulink Aerospace Toolbox allows the implementation of shear, gusts, and different atmospheric conditions, such as the humidity, dew point, temperature, etc. Once the complex scenario is created, the system allows the simulation using MATLAB®S-Functions, which enable the import of the libraries that were created for the FDES.

The second part aims to test different changes on the source code of the PX4 autopilot. As it is mentioned before, the trajectory generation function, the flight modes and the created MAVLINK messages require the modification of the source code and the creation of a firmware image particular for the implementation of the system. Architecture-wise the SITL is similar to the HITL architecture illustrated in Figure 5.6, however it does not require any hardware and can be fully implemented into one station. The purpose of the

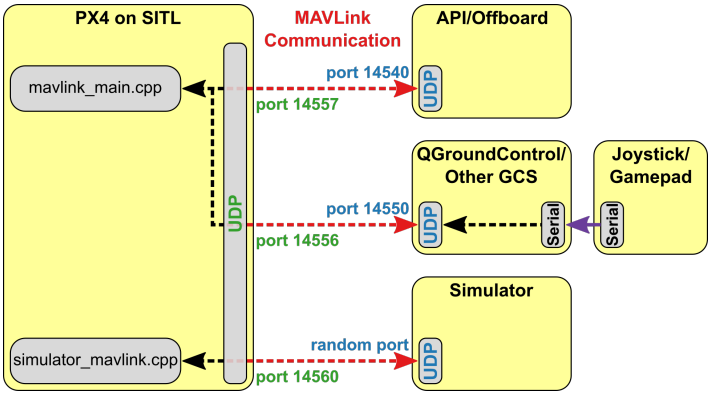


Figure 5.7 The SITL bench architecture shows the different modules and the communication means of the bench.

SITL test bench allow rapid prototyping and testing of different functionalities before compiling it into a final version.

Both the HITL and SITL test benches allow a full functional testing of the aforementioned architecture before testing the system on-board. The off-board testing was a very important part of the design of the FDES, which permits that different configurations and software versions at various levels are being explored.

5.4 Chapter summary and conclusions

This chapter discussed the integration of the different functionalities into a Flight Duration Enhancement System (FDES), that follows a systems engineering approach to implement the software and hardware architecture.

A set of system requirements are defined, from which the system architecture is derived, and then cascaded into the implementation of the different functionalities. In this chapter, all the design decisions, such as the selection of hardware components, the definition of the software and hardware architectures, etc. are justified with a based on the defined requirements. The design philosophy aimed to produce a system that can be scalable, and has high levels of maintainability and upgradeability without modifying the components or other pieces of software.

In terms of the hardware architecture the main design decision was the utilization of a FDES module with powerful characteristics in order to implement the different algorithms that were discussed in the previous chapters. Also, the selection of a COTS autopilot and sensors allow the implementation of the different functionalities as expected.

The software architecture had different design decisions that include the selection of a loosely coupled architecture in which the different functionalities are enabled and communicated with a topic-publish and subscribe scheme. In addition the selection of the means for interaction between the functionalities, the selection of the main communication protocol were discussed and justified according the system requirements.

Finally, the definition of the Software-In-The-Loop (SITL) and Hardware-In-The-Loop (HITL) architectures were defined. These functions are the foundation of the testing activities that will determine if the different hypothesis presented throughout this thesis are correct.

The main contribution of this chapter was the utilization of a systems-engineering approach in order to define and implement in a well-defined structure the different functions. The systems-engineering model is not a typical approach in research, however, its utilization permits the definition of appropriate test to validate and verify the different functionalities at various levels. Therefore, each functionality could be tested separately even at a prototype phase, and also during integration, allowing the utilization of real-telemetry data, or the testing of new missions.

The next chapter describes the different tests that were performed to each functionality at the stages described above. Conjointly, the description of the different testing platforms and sites is presented showing the results of the extensive experimental campaign that took place in order to validate the system.

6 Systems validation and verification, real experiments and results analysis

It works! It works!

Dr. Emmet Brown

This chapter describes the experimental process followed throughout the design and implementation of the FDES. Since the design approach followed the systems engineering methodology, there was an extensive testing campaign throughout the design and implementation process. These testing stages intend to validate each function at high an low level. The testing process started at the prototyping of the implementation of the different functionalities, and for this the MATLAB®SIMULINK®environment was utilized. Once the functions were properly tested in the simulation environment, the next stage was to implement it in the actual hardware followed by the SITL and HITL testing. After this stage, the system was tested in different locations and scenarios including the metropolitan Seville region, the coastal region of the South of Spain and some experiments that were performed in collaboration with other partners of the MarineUAS project. The chapter shows the validation and verification methods employed at each stage of the development, including a description of the experimental platforms and the insights of the different testing methods.

6.1 Wind field estimation and features identification testing

The first part of the development was the creation of algorithms for the estimation of the wind field (wind velocity and wind velocity rate of change). For this purpose, the initial stage was the implementation of the wind models in MATLAB®SIMULINK®environment, together with the implementation of the dynamic model presented in Chapter 2, with some

aerodynamic characteristics that are discussed in the next sections. The Aerospace toolbox contains wind-model blocks of shear, discrete and continuous gusts. In addition the AeroSim®blockset has been utilized to generate the discussed a fixed wing UAS model. Figure 6.1 illustrates the resultant model.

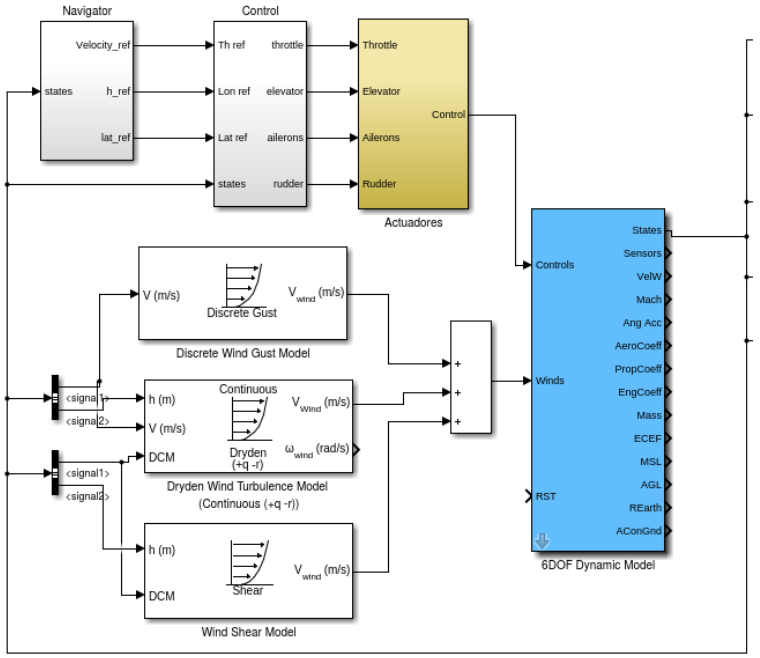


Figure 6.1 Simulink®model of the simulation environment for the wind field estimation and the estimation of the feature parameters. This was created with the Aerospace toolbox and the AeroSim blockset. The aerodynamic parameters were adjusted to match one of the platforms utilized in the experiments.

Table 6.1 shows the characteristics of the computer used in order to perform the simulations.

Table 6.1 Simulation Computer Relevant Characteristics.

CPU	Intel Core i7-5500U CPU 2.40GHz x 4
RAM	15.6 GiB
Graphics	Intel HD Graphics 5500 (Broadwell GT2)
OS Type	64-Bit
OS	Ubuntu 16.04 lts

Initially the intention was to test the system capabilities to identify the resultant wind field with inputs, and the response of the system with noise added to the process. The control system at this point was not optimized to work on the wind field, therefore, only a set of trimming parameters was identified, these are shown in Table 6.2.

Table 6.2 Selected Trimming Parameters.

Trim airspeed	25 m/s
Trim altitude	150 m
Trim bank angle	0^o
Fuel mass	2 kg
Flap setting	0

The scenario considers a planned helix flight ascending trajectory. Once the vehicle starts its flight, the trajectory is under the influence of different wind features which appear alone or multiple at a single time. Two scenarios are considered. The first one considers each feature separately (shear, discrete gust, continuous gust) and the second one considers all features at the same time. The purpose of this simulation is to prove the ability of the system in controlled conditions of detecting the features separately. Figure 6.2 depicts the wind effects on the trajectory.

While running the scenarios, the estimation methods were run in order to compare the data with the modeled one. The most common representation used at this stage is the wind vs. altitude graph, because it shows the response of the different wind features in a common baseline. The wind effects of the scenarios depicted in Figure 6.2 is shown in Figure 6.3.

The first tests goal was to prove that the system is able to detect the shear feature, discrete gusts and generating discrete gusts alerts. Once the features were identified, the next step was to generate a prediction model of the system that will generate the most probable wind speed at a certain altitude given the necessary conditions. The estimation process gets the wind velocity and the wind velocity rate of change and feeds the prediction process which identifies the present features, and characterizes the features based on the algorithms presented in Chapter 2. The features allow the prediction of a most probable wind speed at a given time, which is a measure on how good the feature identification and characterization function is. The results are presented in Figure 6.4.

Note that the results presented in Figure 6.4 do not consider any potential changes in the surface roughness and stability, which gives the possibility to extend the prediction

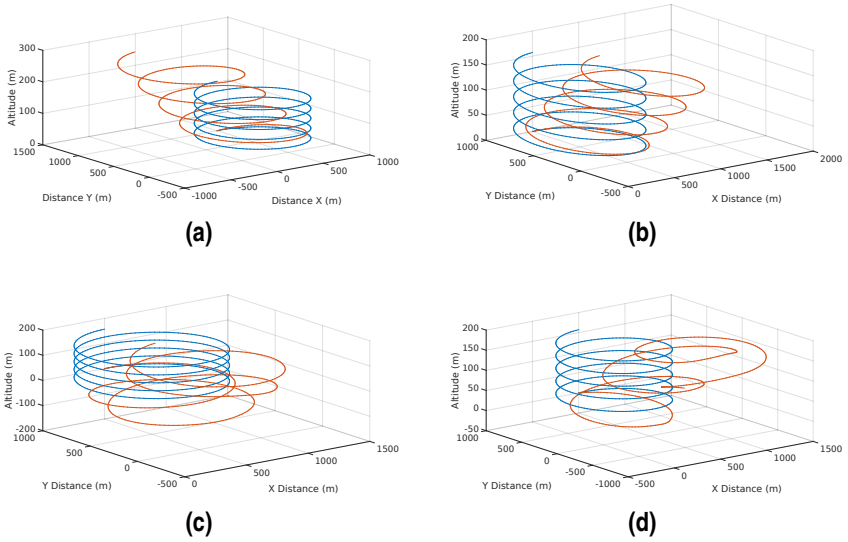


Figure 6.2 Effects of different simulated features on the vehicle trajectory: (a) the effect of a shear wind with increasing deviation as altitude rises; (b) the effects of a discrete gust with a constant deviation on the trajectory in a single direction; (c) a chaotic deviation due to the effects of a continuous gust, and finally (d) the total effects of the wind present at the same time.

path into the whole space. However, the incursion of noise and the presence of different features appearing at different times, provide the test with the means to validate the wind-related functionalities. In the results, it is observed that the sudden jump in the wind speed corresponds to the presence of a discrete gusts, then the predicted wind speed has some convergence time, and reacts quickly in the presence of the gust that occurs approximately after 250 seconds of simulation.

6.1.1 Wind feature identification validation simulations

To understand the identification process, other scenarios were considered in which the estimation of the wind speed, wind speed rate of change and wind direction was provided with the autopilot and compared to the methods presented in this thesis. This scenario utilizes off-board data obtained after different flights performed with a glider UAS (see Figure 6.5). Once the data is filtered, the portion of data that is usable is converted into a Simulink®`timeseries` object and fed to the identification algorithms. In addition, to validate the usage of the Weibull distribution, the uitilized data is fitted into a Weibull distribution using the MATLAB®statistic's and machine learning tools, then the Weibull

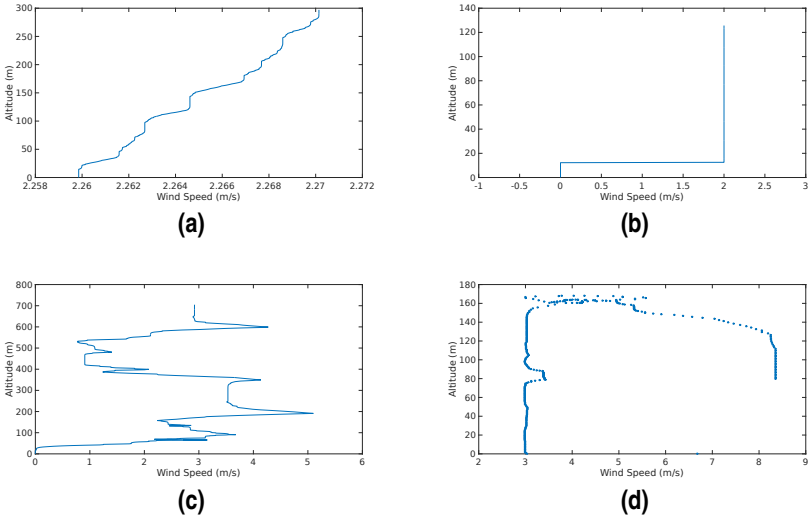


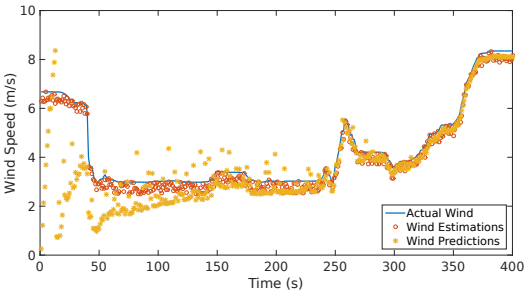
Figure 6.3 Wind Speed/Altitude maps of the different simulation scenarios: (a) the wind shear as an increase of wind speed with altitude; (b) a sudden increase in wind speed at a certain altitude (discrete gust); (c) a continuous gust with a chaotic effect and rapid increases and decreases of wind speed; and (d) the sum of the three effects.

parameters are compared to the ones calculated with the GA by utilizing a graphical function, namely toolbox`wblplot`, which enables a comparison between the obtained data (most probable wind speeds) and the PDF.

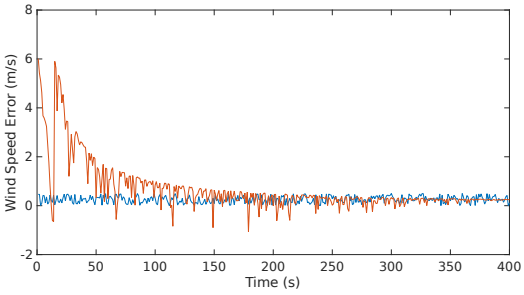
Two flights were considered on this analysis, the trajectories can be seen in Figure 6.6. The goal was to be able to analyze the telemetry data and see if the system is able to identify any potential feature. The experiments took places in the region of Brenes, Seville, Spain (37°31'08.6"N 5°51'31.5"W), and were executed in the summer season, with wind speeds close to 15 m/s for the first flight and low wind speeds with an average of 1.5 m/s on the second flight, as reported by the nearest meteorological station. Note that all the experiments performed in this region consider a surface stability is considered high since the summer season does not have much variability on the wind conditions and the roughness is considered to be of 0.5 as observed in Table 2.1, which implies a minimal impact on the shear identification.

As it mentioned in the previous chapters, the PX4 autopilot provides with estimates of wind speed and wind direction. The corresponding values of the flight depicted in Figure 6.6 can be observed in Figure 6.7.

In Figure 6.7b, it can be observed that there is a lot of variance in the direction, specially



(a)



(b)

Figure 6.4 Actual, estimated and predicted wind speed (a) and wind speed error (b) in the considered scenario. The predicted wind starts with high dispersion, however, it converges to the actual value within 100 seconds.

Figure 6.5 The glider utilized to perform the initial testing of the FDES wind-related functionalities.

in the wind direction calculation. These is attributed to the changes in attitude of the platform that is closely coupled with the calculation of the direction and does not provide with any method for rejecting such perturbations. On the other hand, the wind speed does not present significant changes, since the changes occurs at a very low wind speed. Next step was to feed the data-set of the wind speed into the Weibull distribution, for this,

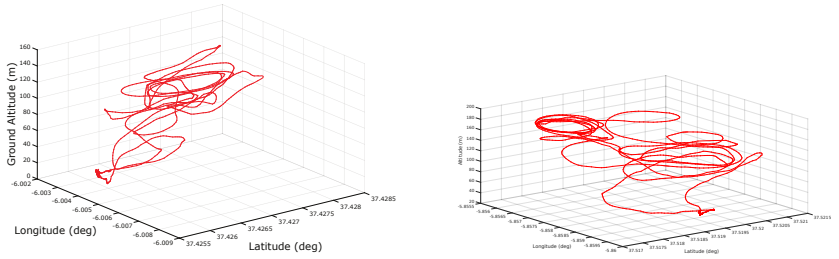


Figure 6.6 The trajectories of the experiments utilized for the feature identification process experiments that took place in Brenes, Spain. (a) and (b) show spiral flights and several altitude changes, which is adequate for the estimation process.

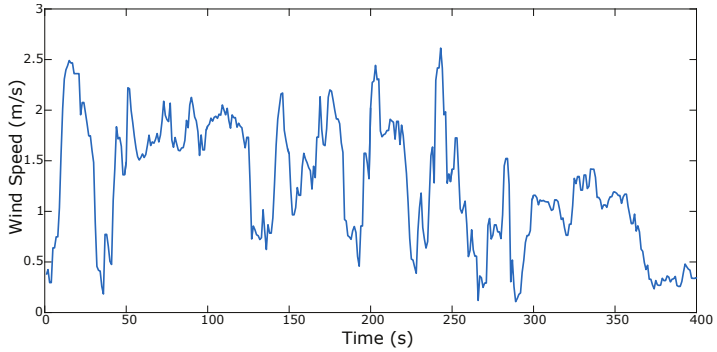
only the portions of the spiral flight is selected. The results of the fit are observed in the distribution probability plot depicted in Figure 6.8.

The telemetry data was then utilized to calculate the wind speed with the *direct computation method* presented in Section 2.4. The goal was to see if the obtained data fits into the Weibull distribution, as a validation means for data coherence. If the mean of this data was closed to the reported in the closest meteorological station, located at the Seville Airport (37°24'51.3"N 5°53'37.6"W). The results of the wind vector estimation method were calculated in the NED frame. Additionally, the RMS error was approximated following the error analysis from (2.50), which is derived from the Weibull distribution parameters. These results can be observed in Figure 6.9

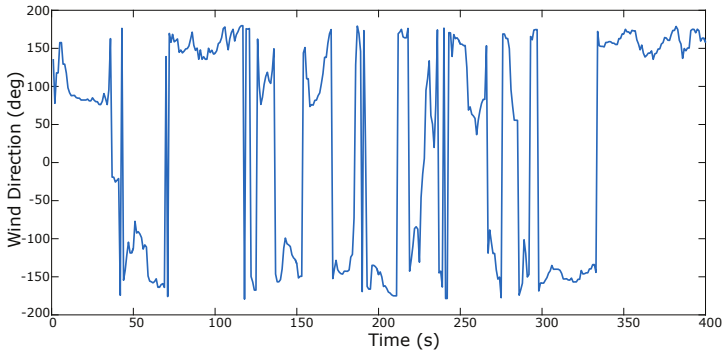
The next step was to determine the Weibull parameters of the selected data. The desired confidence interval of the parameter estimation was selected to be $\varepsilon = 0.95$ of (2.49), which is a common value for wind prediction [31] (see Table 6.3).

Table 6.3 Shape parameter κ and the scaling parameter ν for the Weibull distribution of the wind magnitude dataset obtained with the direct computation method of the flight shown in Figure 6.6a.

ν	12.4236 m/s
ν lower bound	12.3584 m/s
ν lower bound	12.4920 m/s
κ	2.5869 m/s
κ lower bound	2.3166 m/s
κ lower bound	2.6715 m/s



(a)



(b)

Figure 6.7 The results obtained with the autopilot calculation of the wind speed and direction on the experiment presented in Figure 6.6a, in which (a) shows the wind speed and (b) show the wind direction, in which abrupt changes are observed despite the assessment on the region stability and roughness.

The Weibull probability plot that illustrates the dataset can be seen in Figure 6.10, proving that the dataset matches a Weibull distribution, thus, the results are statistically coherent. As it is mentioned before the average wind speed is in the range of the wind speed reported in the Airport station.

Taking the whole dataset, the algorithm intended to find a characterization curve of the wind-shear. Initially the experiments intended to fit a simpler model of the shear, based on an empirical power law, which is described in (6.1).

$$\frac{W_1}{W_2} = \frac{h_1}{h_2}^{\varepsilon} \tag{6.1}$$

The goal was to find a coefficient ε, namely the Prandtl exponent, that provides with

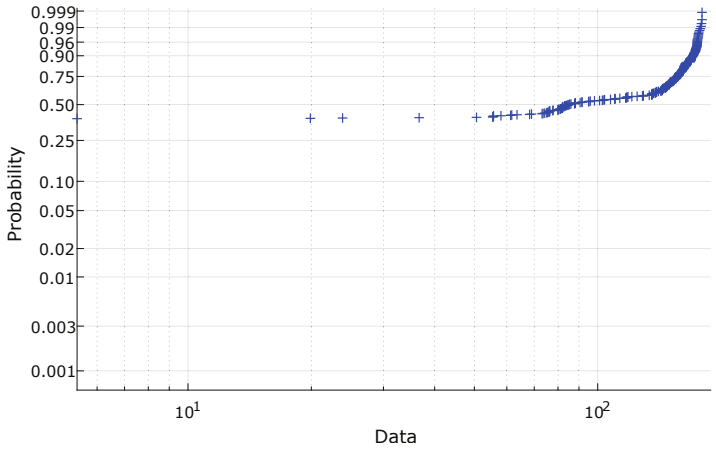


Figure 6.8 Weibull distribution plot of wind speed magnitude calculated by the PX4 in the helical portion of the flight shown in Figure 6.6a.

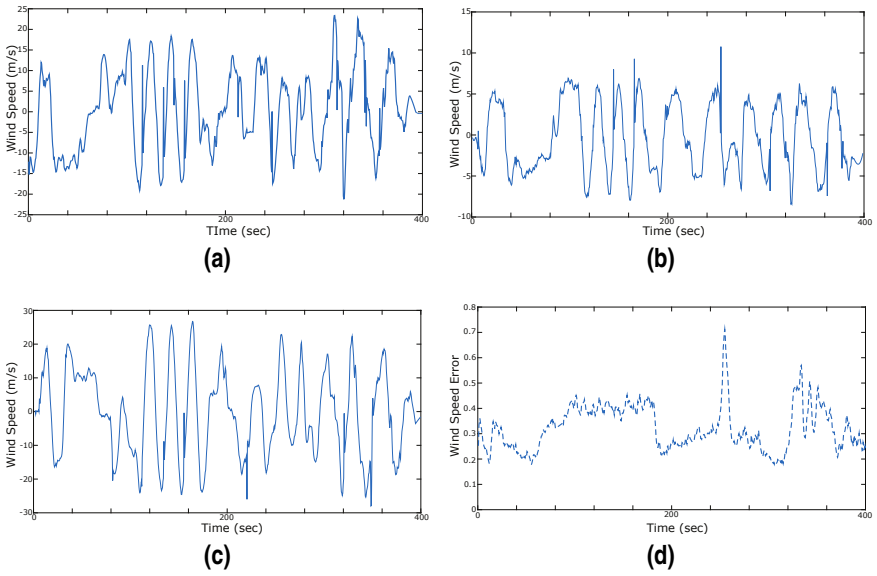


Figure 6.9 The estimation results with the direct computation method: (a) shows the north component, (b) corresponds to the east component, (c) is the down component, and (d) is the estimated RMSE error on the total magnitude.

the best possible fit of the dataset. The model evolved in later experiments, allowing the characterization of a more precise model, and simplifying the fitting process, since the goal became to find the coefficient A_σ in (2.9), which becomes into a simple problem.

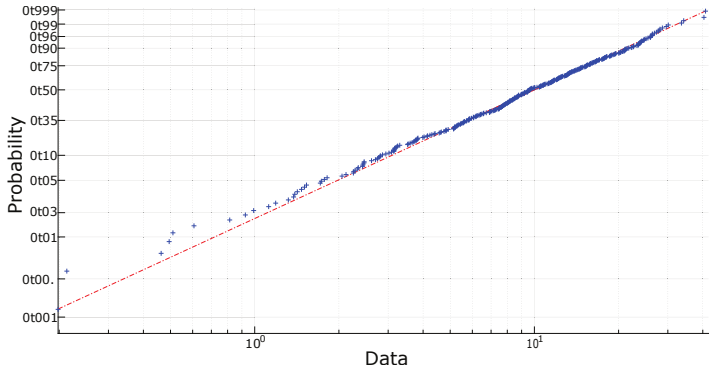


Figure 6.10 Weibull distribution plot of wind speed magnitude calculated off-board with the direct computation method in the helical portion of the flight shown in Figure 6.6a. The red dots indicate the theoretical Weibull distribution with the proposed parameters and the blue dots represent the dataset of the function.

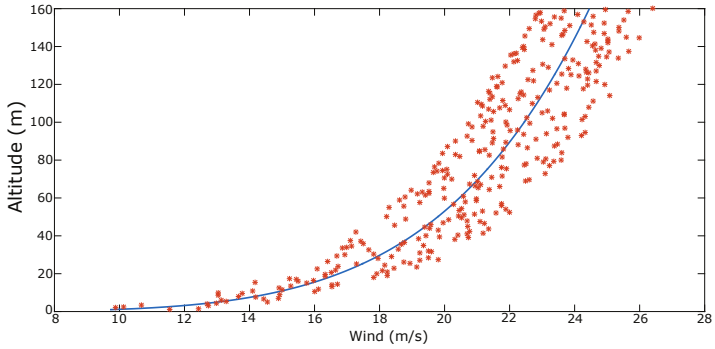


Figure 6.11 Determination of Prandtl exponent with the data estimates obtained with all the dataset of the flight Figure 6.6a.

The results obtained with the empirical power law can be observed in Figure 6.11. It is suggested that the value of $\varepsilon = 0.14$ is applicable to a wind shear, however it was observed that a value of $\varepsilon = 0.18$ was more accurate for this particular dataset.

The prediction process evolved from the determination to the Prandtl coefficient ε to the determination of the aforementioned A_σ coefficient, which allows a more computational efficient process. The Prandtl coefficient requires different comparisons among wind speeds to fit in the best possible curve, however, in the case of the estimation of A_σ, the process is simplified as only two reference values are required.

The results of the glider in the flight presented in Figure 6.6b followed the same estimation process, however, the use of the empirical power law was substituted to the use of equation (2.9). The data was fitted to a curve, in which the resultant coefficient is

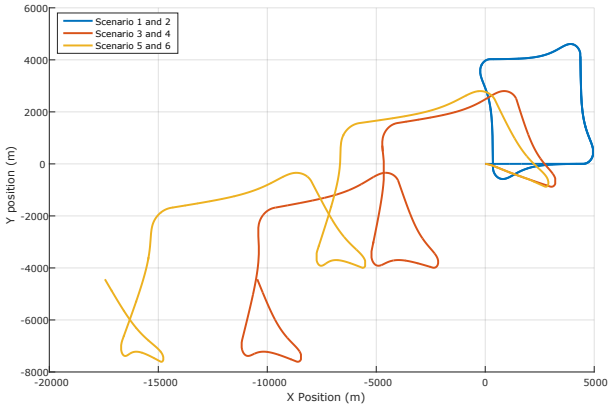


Figure 6.12 Trajectories of the simulated flights that show the effect of the different wind features when the aircraft is subject to them and the controller is unable to reject the disturbance and loses the intended trajectory track. In these simulations, the heading control was enabled and the waypoint control was disabled.

$A_\sigma = 1.43$.

With the evolution of the prediction process, the function was re-tested in a simulated scenario, which consisted on testing different cases that combine the wind features. The tested flight on a squared trajectory, with the implementation of the waypoint tracking algorithm of the PX4. The flights have 2 types of changes in altitude, one that has a sinusoidal wind vertical profile and the other a stepped profile with random altitude changes and holds. The MATLAB®Simulink®environment was utilized for this simulation and the simulation characteristics are shown in Table 6.4.

Table 6.4 Simulation 2 Scenarios Characteristics.

Scenario	Wind Features	ΔAltitude
1	Constant Wind	High (Sinusoidal)
2	Constant Wind	Low (Steps)
3	Wind Shear	High (Sinusoidal)
4	Wind Shear	Low (Steps)
5	Wind Shear + Gust	High (Sinusoidal)
6	Wind Shear + Gust	Low (Steps)

Each scenario was simulated for 1500 seconds, and a maximum shear speed was considered to be $W_{\text{shear}_{max}} = 15$ at $h = 300$m. The tested horizontal profiles are shown in Figure 6.12 and the horizontal profiles are depicted in Figure 6.13.

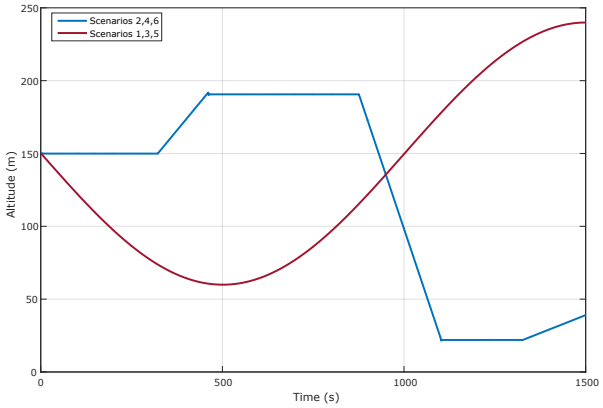


Figure 6.13 Vertical profiles of the simulated flights of simulation 2, showing the stepped vertical profile in blue and the sinusoidal profile in red. The altitude changes allow the estimation of the wind field in a broader altitude spectrum, which enhances the prediction process.

The simulation intends to validate the ability of the system to react to different features and to analyze the convergence time of the prediction. At every time, the prediction process intends to estimate the best fit to characterize the shear layer. The goal of the first two scenarios is to present very low winds to detect if the system rejects the noise and does not consider it a feature, even with the presence of an altitude change. The third and fourth scenarios have only a shear model activated with a different vertical profile. The last two cases consider the presence of a discrete gust that appears after a few minutes after the UAS is airborne. The results of the estimation include the average wind speed $\mu(W)$, the wind-shear coefficient A, and the magnitude of a discrete gust if it is present W_m as in Section 2.1.1.1, which is sufficient to assess the efficiency of the wind-estimation and identification functions.

Table 6.5 Scenarios outputs (single run).

Scenario	$\mu(W)$	A	W_m
1	0.1695m/s	N/A	N/A
2	0.1694m/s	N/A	N/A
3	11.2979m/s	1.5384	N/A
4	11.2972m/s	1.7982	N/A
5	15.5237m/s	1.4415,1.8129	4.96m/s
6	15.5176m/s	1.2712, 1.1321	4.94m/s

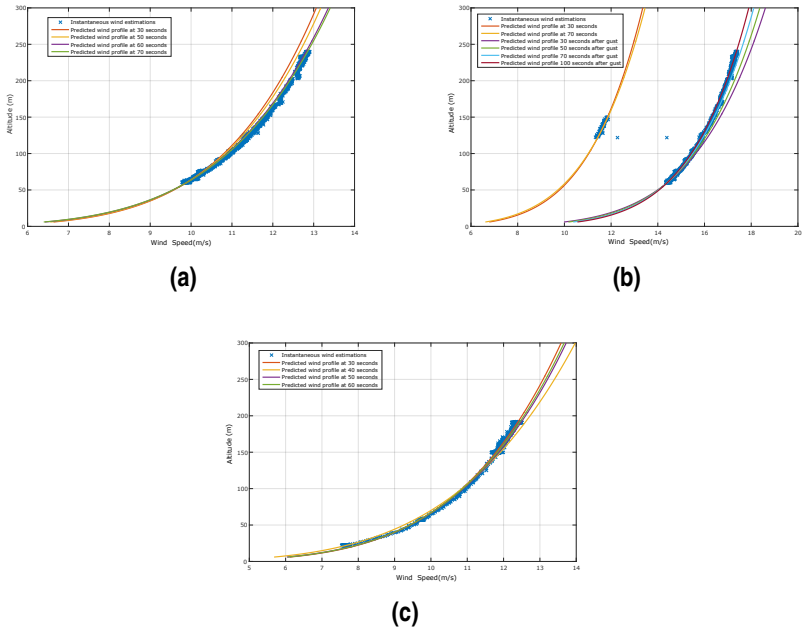


Figure 6.14 The evolution of the prediction process presented in the scenarios shown in Table 6.5, considering only the scenarios with the presence of a wind shear on sinusoidal and stepped vertical profiles. (a) and (c) correspond to the scenario 3 and 4 respectively, in which it can be observed that the convergence time is faster in a more varying vertical profile. (b) shows the scenario 5, in which a discrete gust provokes a sudden change on the estimation region.

As it is mentioned before, the prediction process is continuously evolving with the addition of new data into the data-set, and even the inclusion of external data allow the possibility of identifying the features more accurately. This process evolution in time is illustrated in Figure 6.14 in which the development of the prediction towards the final convergence value is illustrated. The goal was to determine the convergence time, to calculate the computational time of the identification cycle to assess the impact on the overall computational time and also to prove the ability to detect multiple features.

The results presented in Figure 6.14 proved that the method of looking for jumps in the standard deviation is adequate to detect external features, such as, discrete gusts. In the case of the computational cost, the evolution process has a GA algorithm which may impact significantly the overall process of flight duration enhancement. However, by utilizing the `clocks` function of c++ `time.h` library, the computational time can be determined by dividing the result into the constant `CLOCKS_PER_SECOND` which is part of the same

Table 6.6 SkyWalker Characteristics.

Wing Span	2122 mm
Wing Area	80 dm^2
Max Payload	2 kg
Center of Gravity	435 mm away from nose

(a) (b)

Figure 6.15 The SkyWalker X8 (a), from (SkyWalker Technology Co. Ltd, Wuhan, China) is a flying wing platform with a carbon fiber frame. The 4525DO sensor (TE Connectivity Ltd., Shaffhausen, Switzerland) with a resolution of 0.84 Pa [80].

library. The result shows an average execution time of $\mu(t_{\mathbf{GA}}) = 0.053$s with an standard deviation of $\sigma(t_{\mathbf{GA}}) = 0.012$s.

6.1.2 Wind estimation and feature identification SITL tests

After the wind field estimation and wind field identification function were validated at a prototype level, the next stage was to prove the estimation and identification process in a SITL scenario, this occurred after the creation of header files and implementation file. The implementation uses a Simulink®that interacts with the ground station software to replicate entirely a series of flights that were performed in the region of Brenes, Seville with another platform, the SkyWalker X8 (see Figure 6.15), which is a very popular commercial platform due to its payload capabilities. Its characteristics are enumerated in Table 6.6.

The relevant information of the experiments that were replicated in the SITL test can be observed in Table 6.7, and the trajectories of the corresponding flights can be observed in Figure 6.16.

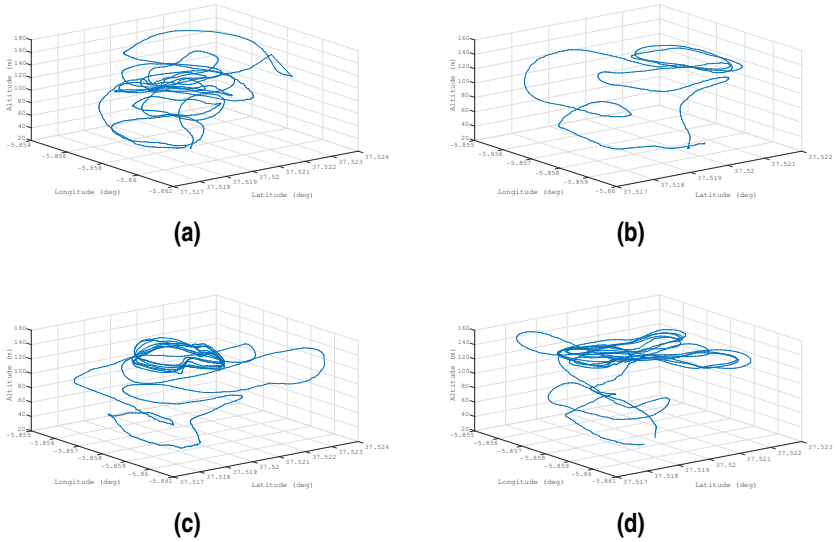


Figure 6.16 The trajectories of the flights performed by the SkyWalker X8 in the region of Brenes, Seville. The trajectories show helical flights with at altitude that goes from 100m to 150m above ground.

Table 6.7 Experiments information.

	Flight 1	Flight 2	Flight 3	Flight 4
Duration	521 s	315 s	631 s	749 s
Distance Traveled	5.1 km	3.7 km	6.3 km	7.4 km
Maximum Altitude	179 m	125 m	134 m	146 m

The estimation results obtained from the four flights have a more noisy input, which is attributed with the windy days with low stability in the region at the time of the flights. Nevertheless, the system was able to perform an estimate with these noisy measurements, excluding the outlying points that were caused due to the instability of the system. The wind vs. altitude maps of the flights can be observe in Figure 6.17

Note that in Figure 6.17b the results show two curves, which is similar to the simulation result showed in Figure 6.14b in which a jump in the standard deviation was interpreted as a discrete gusts, and then the curve is characterized again. The scenarios allow understanding how the noise in the estimation, derived from the airspeed measurement and the region stability affect the prediction process and the estimation. In the wind estimation data

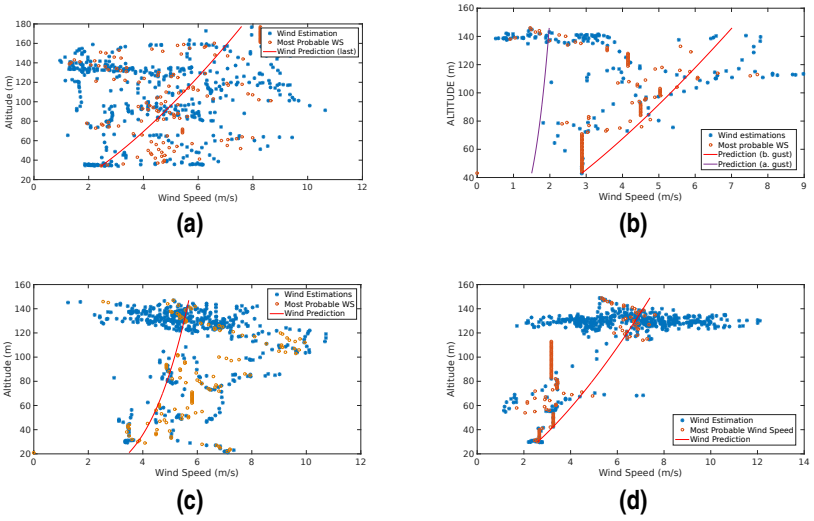


Figure 6.17 The estimation, most probable wind speeds and last shear prediction of the four flights presented in Figure 6.16. The blue points represent the wind speed estimations performed with the direct computation method, the orange dots represent the evolution of the most probable wind speed at the respective altitude and the lines represent the latest shear curve.

observed in Figure 6.17b, Figure 6.17c, and Figure 6.17d it can be seen a concentration of disperse wind estimations in the altitude in which the helical trajectory is performed. The concentrated data is distributed following the Weibull model allowing at a more accurate data prediction at those stages. Moreover, the actual variability of the data may not present a significant impact on the duration enhancement process, since the goal is to find just adequate conditions for the execution of the soaring maneuver and not to get a precise set of measurements.

6.2 Trajectory-related simulations

In the case of the trajectory generation and sequence reconfiguration, there were different experiments that were considered in to validate these functionalities. These experiments consider the use of the SITL testing bench described in the previous chapter, including also the software developed by Balampanis [67] for partitioning complex areas and the generation of an initial waypoint sequence, and also the use of telemetry data for after-processing to test the different capabilities.

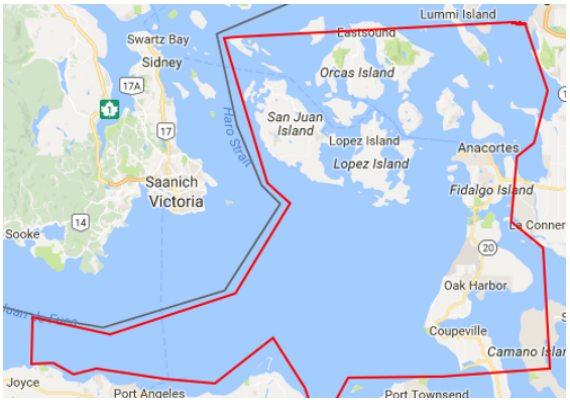


Figure 6.18 Test case area in the north of the Seattle Metropolitan Area in the state of Washington in the United States. The red-line polygon defines the mission area.

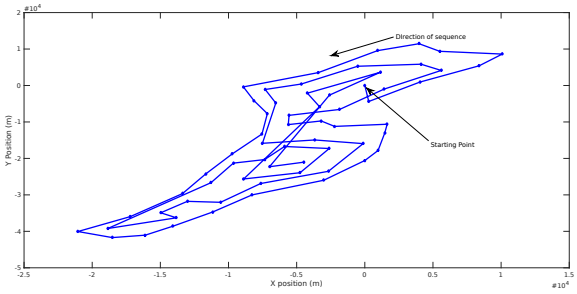


Figure 6.19 Initial area decomposition that indicates the starting point in the top part of the waypoints and the trajectory.

6.2.1 Sequence reconfiguration simulations

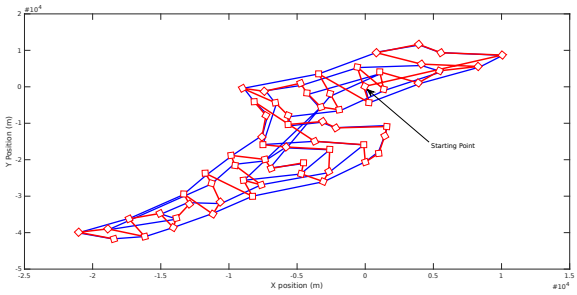
In the prototyping phase, the function was implemented in a similar way than the wind estimation and feature identification algorithms. The sequence reconfiguration consist in the redefinition of the order of the waypoints of a given surveillance mission to maximize the wind energy. This section describes the validation test performed to this functionality. Fort this test, the greater Seattle area was considered (see Figure 6.18).

Since the area is considerably big, it was partitioned and divided into task that have to be performed by different UAV's. For this case, the area was divided considering two mission agents and the resultant waypoint sequence of the first agent can be seen in Figure 6.19.

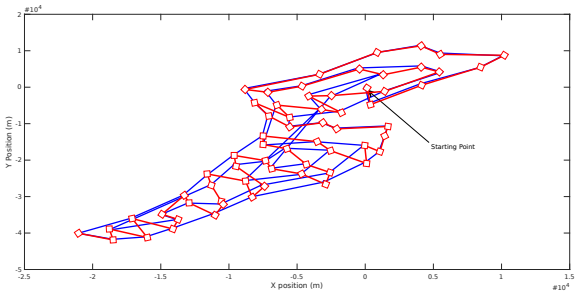
Once the SITL simulation is performed, the summary of the resulting mission is observed in Table 6.8, in which no wind information was fed into the reconfiguration function prior

Table 6.8 First Sequencing Simulation Results.

Flight Duration	3.2 h
Medium Altitude	100 m ASL
Total Energy Consumption	≈22385.66 W/h
Average Wind Speed	0 m/s
Average Airspeed	20.93 m/s



(a)



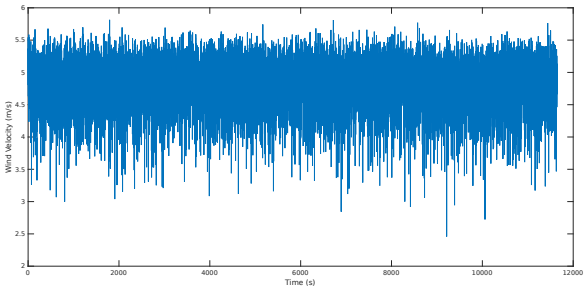
(b)

Figure 6.20 The waypoint sequences reconfigured considering a wind shear input (a), and a discrete gust input (b).

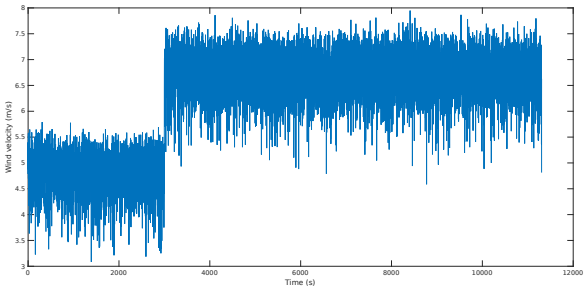
the analysis. In the SITL test, however, the wind is considered to be constant without any shear or gust effects.

The initial energy consumption was the one estimated with the method proposed by [81], in which the propulsion energy is estimated as a function of the trajectory. This estimation was utilized for this exercise only and the equations can be found in Appendix B.

Then, the presence of a gust and a shear is considered, so that the system reconfigures the system based on the pre-flight inputs of a shear and a gust. The resultant sequence, with the same starting point can be observed in Figure 6.20.



(a)



(b)

Figure 6.21 The estimates of wind velocity with the presence of a shear a(a) and the presence of a discrete gust input (b).

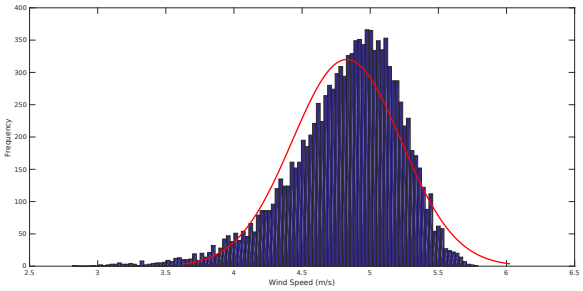
The results after running the SITL simulation can be observed in Table 6.9.

Table 6.9 First Sequencing Simulation Results.

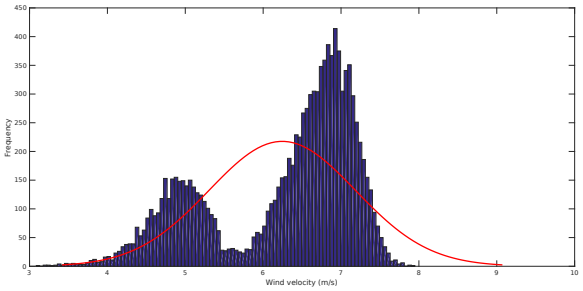
	Flight 2	Flight 3
Flight Duration	3.16 h	3.19 h
Medium Altitude	100 m ASL	100 m ASL
Total Energy Consumption	$\approx$ 19913.408 W/h	$\approx$ 20421.783 W/h
Average Wind Speed	5.01 m/s	6.24 m/s
Average Airspeed	16.74 m/s	15.23 m/s

The results show that after the sequence is reconfigured there is an improvement that goes up to 11% in the presence of low winds, and of 9% in energy consumption in the presence of discrete gusts. In addition, the horizontal wind estimations were also considered, to evaluate the performance of the measurement in different conditions. The estimation of the wind speed for both cases is shown in Figure 6.21.

In order to prove the validity of these results, an additional test was performed to



(a)



(b)

Figure 6.22 Weibull distribution plot of the wind estimations presented in Figure 6.21. The first dataset (a) shows a correct fit into a Weibull distribution . The second dataset (b) shows two spikes due to the presence of a discrete gust, note that the two data concentration spikes is the expected behavior that will lead to the automatic detection of the discrete gust feature.

determine if the wind velocity estimations fit a Weibull distribution. The atmospheric models had a Gaussian noise input, and the distribution of the data can be observed in Figure 6.22.

From the results observed in Table 6.9, it proves, at least in a controlled environment, that the sequence reconfiguration by itself may provide with significant energy savings in very long duration missions. What's more, during the execution of the reconfiguration and with the wind information that was fed into the system, the estimations performed on-board, prove that the system is able to recognize the features.

However, the approach in this research takes a step ahead and includes the definition of soaring trajectories that may increase this gain. The next paragraphs will discuss the next step of the flight path duration which is the trajectory generation and the trajectory tracking.

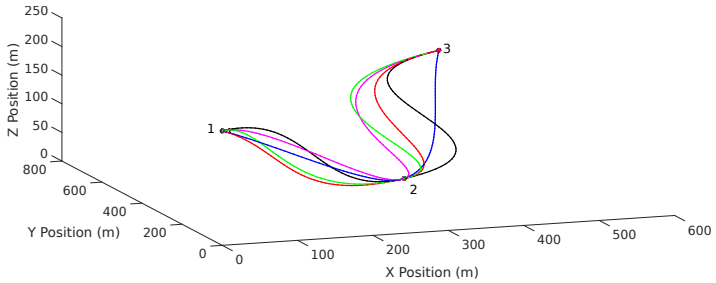


Figure 6.23 The blue curve hasn't any waypoint aligned with the curve. In the red curve the slope of the generated curve at its three waypoints will be fully aligned with the predominant wind speed vector and a tension of $\tau = 1$. The magenta and green curves have all its waypoints with wind-aligned slope unitary vector but with modified tensions for each (magenta with the lowest tension $\tau = 0.75$ and green with a higher tension $\tau = 2.5$). Finally the black one, considers waypoints 1 and 3 aligned with the slope with a high tension value $\tau = 5$.

6.2.2 Trajectory generation and tracking SITL tests

To validate the trajectory generation and tracking, a series of tests were performed, including the different methods presented in Chapter 3 and Chapter 4. Both functionalities are treated in this section because in some cases the simulations utilized the vector-field based tracking method described in Chapter 4 and in other simulations the native PX4 was utilized by creating extra waypoints along the curve.

The selected test consisted in the analysis of a Rayleigh cycle and the effect on the flight after changing the curve parameters. This was done considering the existence of constant wind, with the utilization of the simplified shear model presented in Chapter 2 (2.8), which does not include the atmospheric/ocean interaction. Nevertheless the goal is to prove the effect on different types of Rayleigh cycles while traveling between two waypoints. As it is mentioned in Chapter 4, the Rayleigh cycle is a process of energy gain and energy losses in which, to be considered efficient the gains have to be greater than the loses. Then the effect on these parameters is analyzed. The analyzed Rayleigh cycles are observed in Figure 6.23.

For the generated curves, a maximum rate of change on the curvature is determined. This was accomplished by considering The critical angle of attack while climbing, which is considered to be 25 degrees. Hence, if the maximum angle of attack is reached, the system will adjust the slope vector to produce a smoother climb. Note that the information of the wind field is known, allowing the accurate estimation of the angle of attack considering the tangential framing. To appreciate the energy gains, the theoretical airspeed along

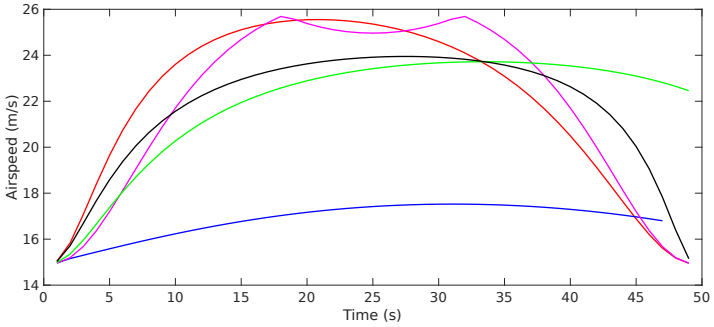


Figure 6.24 This picture shows the different airspeeds computed while tracking the trajectories depicted in Figure 6.23. It is observed that the execution of the Rayleigh cycles is a process of energy gain and loss that varies according to the trajectory type.

these curve was simulated. The control strategy presented in Chapter 5, in which these curves are framed, and then the error is computed. Hence, the airspeed while tracking this trajectory without any sensor error, and also with the starting point at the first waypoint and the orientation of the UAS perfectly aligned to the Bishop frame is computed and shown in Figure 6.24. Note that the aerodynamic parameters utilized for the tracking strategy were taken from the AEROSONDE UAV which consider the following initial conditions and parameters [82]: $V_0 = 15$m/s, $C_{\mathrm{D}_0} = 0.0434$, $C_{\mathrm{L}_0} = 0.036$, $C_{\mathrm{L}_\alpha} = 5.6106$.

In Figure 6.24, it can be observed that the maximum airspeed occurs with the lower tension value and when the three slope unitary vectors at waypoint 1, 2 and 3 are aligned to the wind. Nevertheless, there is an obvious decrease in the dive-out maneuver that goes from waypoint 2 to waypoint 3. The magenta curve, which has a lower tension value present the fastest increase of the airspeed during the dive-in maneuver that goes from waypoint 1 to waypoint 2. The blue line, that only considers the next waypoint as the slope unitary vectors $\tilde{M}_i$ and not the information of the wind, has the lower growth of airspeed, however there is still a slight growth which can be utilized.

6.2.3 Trajectory generation assessment on a full flight plan

The previous tests demonstrated the different types of curves that can be produced with the Hobby method [66] for interpolating splines to replicate the Rayleigh cycles, and showed in a rough manner the effect that each curve has in the case of an ideal shear. Moreover, the generation algorithm has to be able to produce a full flight path with the desired characteristics of continuity and smoothness that are necessary to produce flyable trajectories. Although the intention is not to construct Rayleigh cycles throughout an

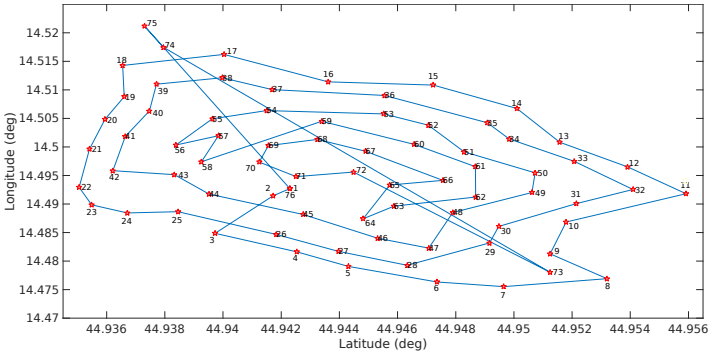


Figure 6.25 A waypoint sequence for a complex area considering a footprint size of $f =$ 300m showing the straight path between the waypoints.

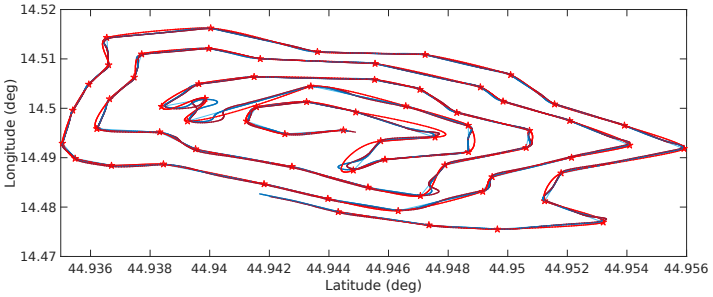


Figure 6.26 Spline interpolation of the flight path of a given waypoint sequence considering an UAS with maximum bank angle of 45º. Note that the alignment of the slope unitary vector $\tilde{M}_i$ is the previous and next waypoint for the ith. and $i+1$ waypoints respectively. The blue line represents the resultant flight path with the native PX4 guidance algorithm, whereas the red line represents the interpolated spline with a lower tension value. The purple line, which is at most covered by the red line shows the interpolated spline with higher tension value.

entire flight plan, the smooth curves and sometimes, the alignment of simply the slope directional vector $\tilde{M}_i$ to the wind, may have a significant impact on the energy saving.

Consider the waypoint sequence depicted in Figure 6.25. The goal is to create a continuous path through the whole waypoints to determine if the trajectory generation algorithm is able to produce a continuous path and also to assess the effect of aligning the slope unitary vector direction of the predominant wind to see if there is still a continuous and flyable path. Firstly the construction of the path with the Hobby algorithm without considering the alignment of the slope unitary vector to the wind of the curves is shown in Figure 6.26.

After zooming into the center of the Figure 6.26, which corresponds to the area of

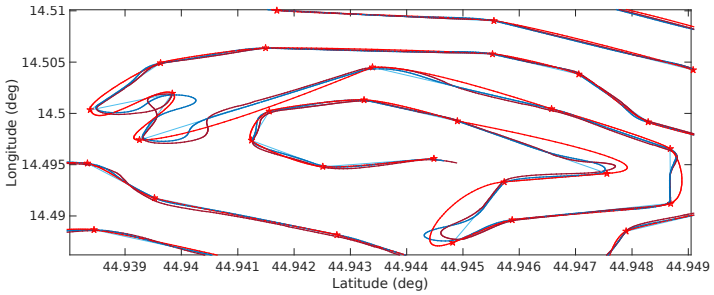


Figure 6.27 A close view of the center area of Figure 6.26, the different curves depicting the effects on tension changes, which are very significant only in the case of sharp turns.

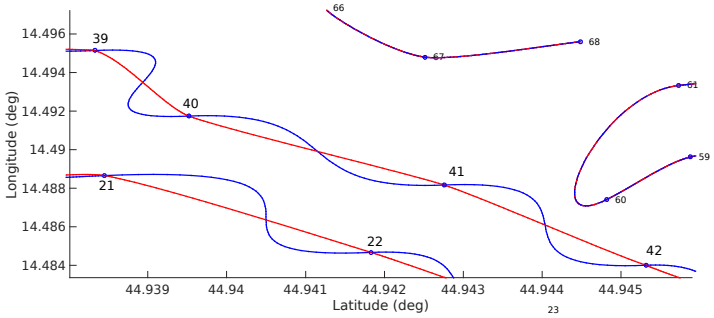


Figure 6.28 A close view of the center area of Figure 6.26 when the desired curve considers the alignment of the wind to the slope unitary vector. This could be considered as the base of the dynamic soaring, since the Rayleigh cycles will require attitude changes.

greatest concentration of points, it can be the effect on the interpolation splines can be observed more in detail while considering the effect of changing the slope to the direction of the next waypoint. This is depicted in Figure 6.28.

If some of the waypoints are aligned to a theoretical wind direction, a close up effect on the resultant trajectory of the middle area can be seen in Figure 6.27.

From the experiment performed in , a first solution on an easy implementation of continuous Rayleigh cycles among mission waypoints was extracted. If the resultant horizontal profiles have a sinusoidal vertical profile, then the resultant curve may produce a Rayleigh cycle.

The next step was the consideration of the generation of the trajectory spline and the vector-field based tracking considering the effects of the interaction of the ocean and the atmosphere in the boundary layer. For this, a fully developed ocean was considered; the

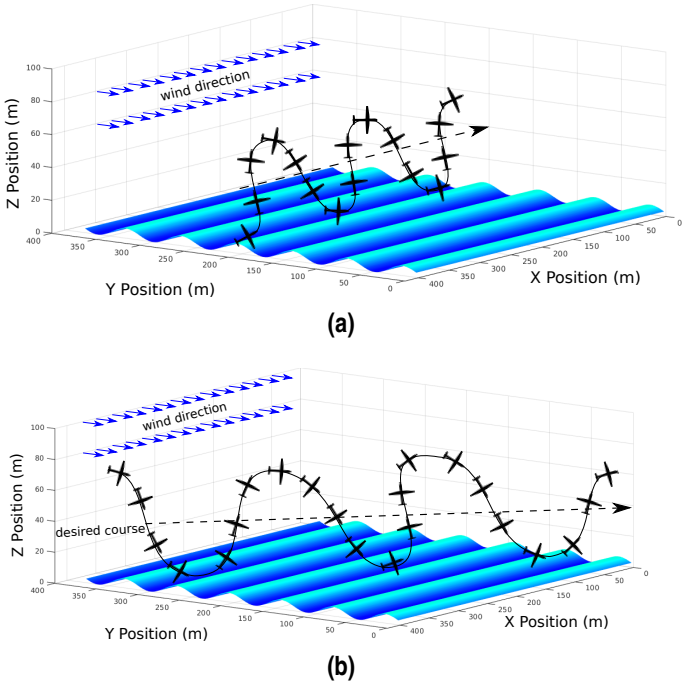
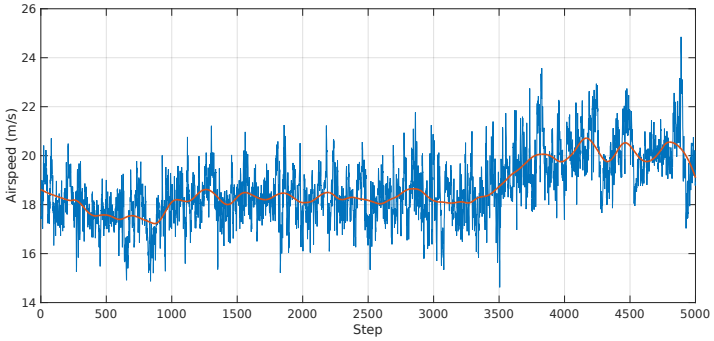


Figure 6.29 Two test cases considered in the simulation of the full Rayleigh cycles. In (a), 2 full Rayleigh cycles are performed: starting from the dive-out maneuver at full throttle, the aircraft aims to dive-in into the next wave below the lee-eddy formation zone to maximize the energy gain, the tension τ_i was adjusted manually at every cycle to match the wave frequency. In (b), 3 Rayleigh cycles are performed, the actual dive-in/dive-out maneuvers are winder with a general diminution of tension value τ_i compared to the first case. Note that in both cases the cycles are performed in a cross-wind direction.

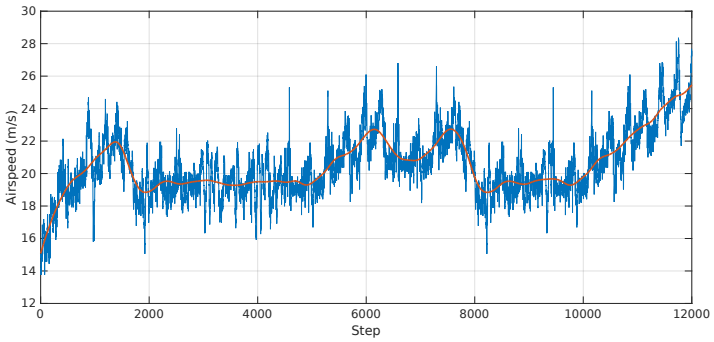
main objective was to assess the energy gains and losses that occur after performing 2 or 3 Rayleigh cycles. The two cases are illustrated in Figure 6.29.

Once the simulations are performed, the next step is to analyze the airspeed results, which are presented in Figure 6.30.

With the results presented in Figure 6.30 at this stage it is very difficult to conclude wether the changes on airspeed will lead to an overall energy gain, specially in the context of a full mission. However, the impact on changing the curve parameters is significant. In addition, it was concluded that the air/ocean interaction models have to be improved, since the observations performed by Richardson [23, 24] suggest that the airspeed changes must be more dramatic, which will be translated into an actual energy gain.



(a)



(b)

Figure 6.30 Airspeed of the UAS after the execution of the Rayleigh cycles depicted in Figure 6.29. (a) shows smaller airspeed changes, compared to (b), in which a lower tension value leads to greater changes in airspeed.

6.3 Validation Experiments

The final step in the development of the FDES was an experimental campaign of several months that assessed the different functionalities on-board aiming to collect a big amount of data in different places and conditions for this analysis. This section will describe the details on such experimental campaign, which was performed in the southern region of Spain and in different seasons of the year, in which every aspect of the system was tested to evaluate its behavior and the proper integration with the system.

6.3.1 Experimental platforms

There are two experimental platforms that were utilized in the various experiments of this test campaign.

Figure 6.31 The SkySurfer X8 is a 2000 mm wingspan fixed-wing gliding UAS; manufactured with EPO, it has a maximum takeoff weight of 1.7 kg. The fuselage length is 1.35 mm, it has a total surface area of approximately 46.3 dm^2 and a loading area of approximately 29.2 dm^2, its center of gravity is located at 75 mm from the leading edge.

The first one was The SkySurfer X8 (see Figure 6.31), which is a gliding UAS, mainly used for data collection for off board data processing but also for the online test of the wind field estimation and also the wind feature characterization algorithms.

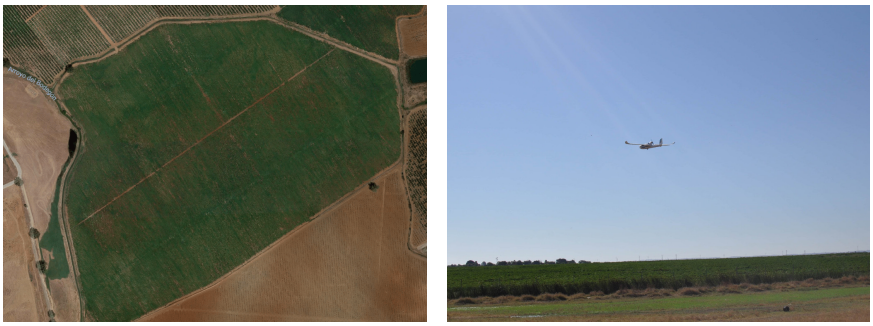
.

The SkySurfer X8 was equipped with a PX4 controller, the actuators of the control surfaces are 6 servos of 17g, with a torque of 1.8kg/cm, and a speed capable of turning 60º in 0.18s. The utilized battery was an 11.1v 2200mAh (20C) Lithium-Polymer battery, which powers the autopilot, the actuator and a brushless motor 3536-KV1200. Due to its limited space and payload the wind estimation function, when was performed online, the algorithms were calculated at the ground station, taking the instantaneous information from the telemetry. The second platform is the Proskytec V1 (see Figure 6.32), this platform is a V-Tail platform with better capabilities of both, weight and endurance, allowing testing at various levels of the different functionalities of the FDES.

The Proskytec V1 was equipped with a PX4 controller, with two 9 g servos and two 17 g servos. It was equipped with a 22.2v 10000 mAh battery for maximum endurance.

Also, the FDES module, the Odroid XU4, was also packed into the avionics compartment. This platform result ideal for the full implementation of the different functionalities and due to its relatively large payload, allows additional mission-related payload without any drawbacks related to the CG or to the takeoff weight.

Figure 6.32 The Proskytec-V1 is a 2000 mm wingspan fixed-wing V-tail UAV; manufactured with EPO, it has a maximum takeoff weight of 3.5 kg, also it has a fuselage length of 1220 mm, a surface area of 70 dm^2. Its center of gravity is located at 100mm from the leading edge.



(a) (b)

Figure 6.33 An aerial view of the Brenes facility (b), and the SkySurfer X8 (b) in a flight on this area. It can be observed that there are no bigger obstacles which are ideal for the analysis of the wind in a wide extension.

6.3.2 Experimental sites

Two main sites were chosen as the testing facilities for the FDES modules, the first one is located in the Northern Metropolitan Area of Seville (37°31’08.6"N 5°51’31.5"W), close to the city of Brenes. This testing site has a wide area without any obstacles, such as mountains or threes, which could affect significantly the surface roughness, impacting the wind estimation. An aerial picture of this site can be observed in Figure 6.33.

The second testing location, was the beach of El Terrón, located at the south of Huelva,

(a) (b)

Figure 6.34 Images of the experiments performed in the region of Huelva, with the Proskytec V1 flying in the horizon while performing a soaring maneuver.

Andalusia, Spain (37°12'26.0"N 7°10'24.5"W), which is a beach without population, which was ideal for the coastal tests. The winds at this region are predominantly coming from the west, and the average wind is of around 10 knots with typical gusts that can reach the 30 knots during the winter. Pictures of the tests conducted in this region can be observed in Figure 6.34.

6.3.3 Tuning experiments

The PX4 autopilot, offers a control system based on a standard control with two main loops. The outer loop, also known as the guidance loop aims to determine the positional error with respect the waypoint in a mission (under the mission mode), and the inner loop which defines the control commands that are going to be given to the different actuators to move the surfaces. The architecture of such system is depicted in Figure 6.35. The PX4 manual suggests a general parameter configuration, based on the work presented by Meier et. al [83] and Park et. al [84]. The Park method uses a L1 control [77], and it has proved to be relatively efficient since the guidance logic uses the vehicle's inertial speed for determining the acceleration command, which helps on the rejection of some disturbances, such as wind.

After some experiments in the first testing site, it was determined that most of the suggested values in [75] are adequate for low winds. However, they have room for improvement, since the low winds ($\mu W <= 10$ knots) allow the controller to have a more aggressive response without unwanted oscillations. The baseline calibration was determined with the suggested values of the PX4 tuning guide [75] and a temperature of $25^{o}C$. The goal was to modify different parameters to achieve the optimal characteristics of the flight. These parameters are: the feed-forward roll gain K_{ff}^{ϕ}, the proportional (roll-rate) gain K_{P}^{ϕ},

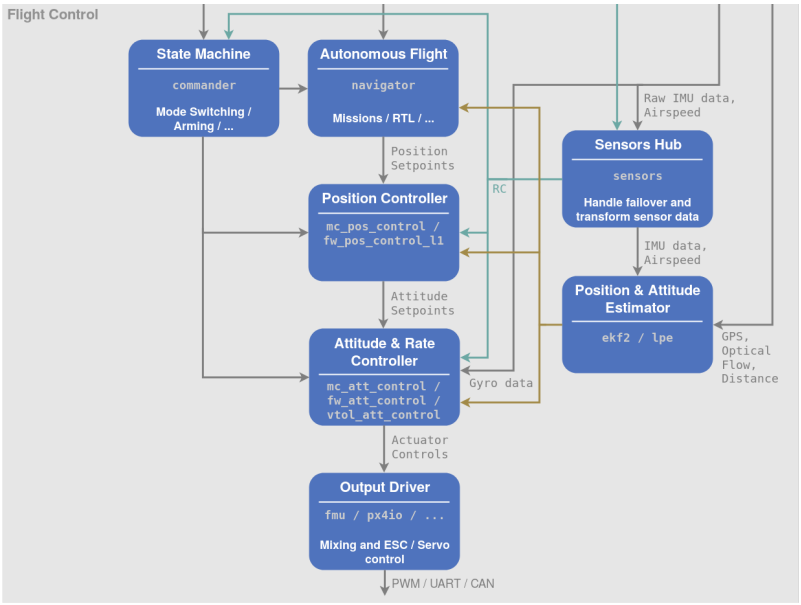


Figure 6.35 The flight control strategy of the PX4 shows the two loops of control. This architecture was presented in [75], and shows the different stages for the different control modes that are available.

the integral (roll trim offset) gain K_I^{ϕ}, the feed-forward pitch gain K_{ff}^{θ}, the proportional (pitch rate) gain K_P^{θ}, the integral (pitch trim offset) gain K_I^{θ}, and the acceptance radius Ω_a. However, during the experiments, it was found that the reduction of the acceptance radius does not provide a significant impact on the performance, compared to the impact of changing the L1-period τ_{L_1}.

The resultant parameters were adjusted according to the increases suggested in [75], until the desired behavior was achieved. Table 6.10 show the baseline parameters, and the enhanced parameters for the given conditions.

The selected mission for tuning was designed in conjunction with other fellows of the MarineUAS project. The primary goal was to solve a coverage problem; nevertheless, the type of mission suited perfectly the objectives of this research. The task consisted in visiting 55 waypoints, intending to coverage the green portion of Figure 6.33a. This mission is shown in Figure 6.36.

After performing the different experiments, the tracking of the different waypoint showed significant differences with the selected values. The system was able to reject the wind disturbances, and the positional error was mainly related to the tuning parameters (see Figure 6.37).

Table 6.10 Tuned Parameters for the SkySurfer and Proskytec V1 in the region of Brenes.

Parameter	Baseline Value	Enhanced Value
K_{FF}^{ϕ}	0,20 rad/s	0,36 rad/s
K_{P}^{ϕ}	0,060 rad/s	0,075 rad/s
K_{I}^{ϕ}	0,010 rad	0,031 rad
K_{FF}^{θ}	0,32 rad/s	0,44 rad/s
K_{P}^{θ}	0,032 rad/s	0,46 rad/s
K_{P}^{θ}	25 m	16 m
Ω_a	2.0 m	0.1 m

Figure 6.36 The selected mission that was utilized for the different tuning experiments. This mission was executed with the SkySurfer X8 UAS.

The wind magnitude estimates, calculated with the *direct computation method* during the tuning experiments can be observed in Figure 6.38. It was observed that there is an initial jump on the wind speed magnitude. This is attributed to the convergence period of the estimation of the airspeed components at the autopilot. However, the average of the wind speed is relatively close to the average speed reported at the Seville Airport during the days of the flight, which were executed on November 6 to November 10 of 2017. During the 3 days, the average wind velocity was reported to be between 18 knots and 25 knots, with gusts that reached up to 35 knots, in accordance to the *Sevilla Aeropuerto* station. These experiments were performed at an altitude of 50 meters above ground, or

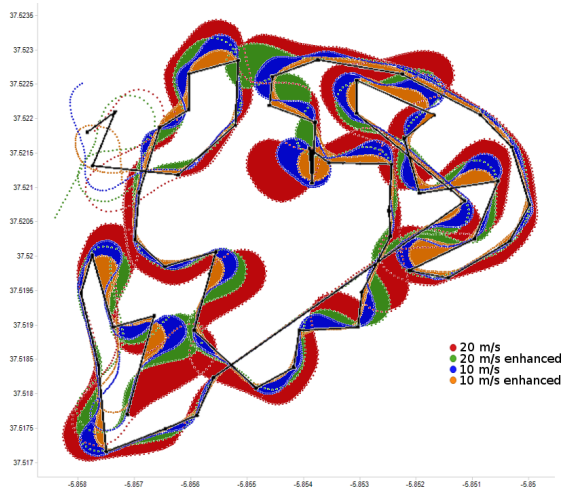
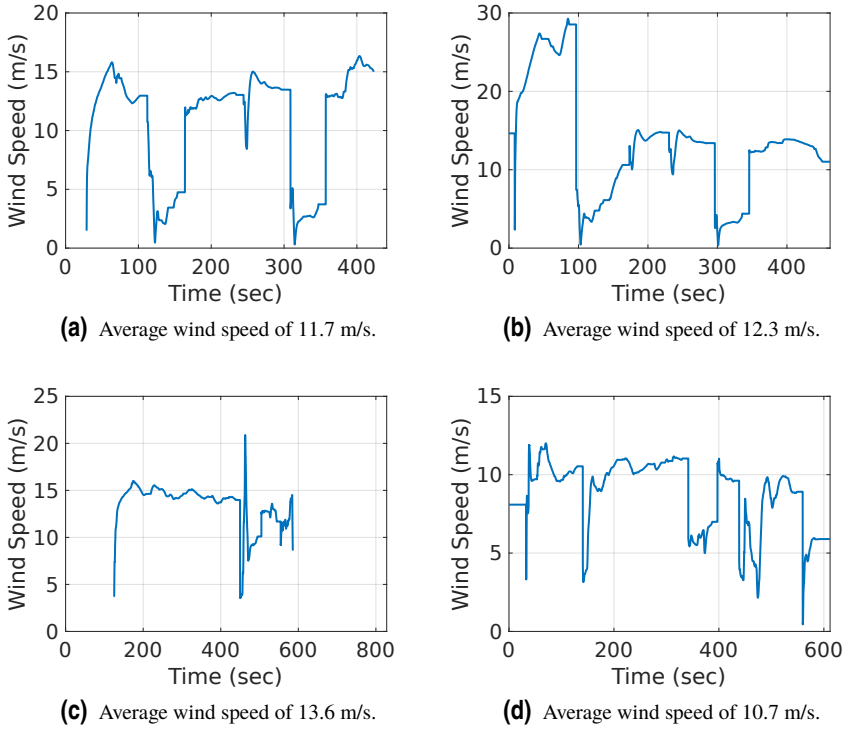


Figure 6.37 The different trajectories and positional errors with the different tracking parameters in 4 flights. The black line represents the aircraft trajectory, the red area represents the error with the low parameters with a higher airspeed, and the green line is the error with higher airspeed with the optimized tuning parameters.

approximately 80 meters. There are different jumps which can be attributed to the presence of discrete gusts among the flight. The FDES detected these as discrete gusts; furthermore, the hops may be attributed also to the changes in heading due to the sharp turns, that cause a sudden increase due to a poor definition of the angle limits on the PX4 native software. The heading of one of the experiments which is similar to the others can be observed in Figure 6.39.

It is easily observed that some of the jumps on the heading calculation are directly cascaded into the wind estimation. Therefore, a correction needs to be performed to avoid this error and to determine the actual wind magnitude. This sudden jumps of the heading are attributed to the assumption of no sideslip in the calculation of the wind, and the direct use of the `sin` and `cosine` functions in the calculation, without considering the range limit changes.

The next step was to produce a corrected wind speed based on the observations made at the previous stage. This correction was performed only by changing the input on the airspeed estimation of the autopilot by modifying the portion of the code that was illustrated in Code 2.1, which consider a zero sideslip into an approximation of the sideslip angle by comparing the ground speed/track vector to the airspeed/heading vector. The corrected wind speeds are observed in Figure 6.40



(a) Average wind speed of 11.7 m/s.

(b) Average wind speed of 12.3 m/s.

(c) Average wind speed of 13.6 m/s.

(d) Average wind speed of 10.7 m/s.

Figure 6.38 This picture shows the estimation of the wind speed with the direct computation method for the four flights, with its trajectories were depicted in Figure 6.37.

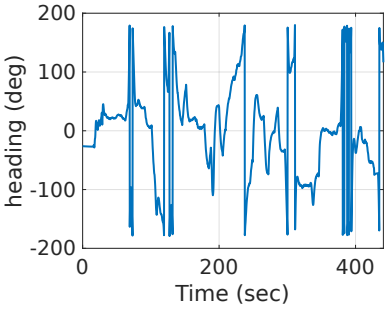
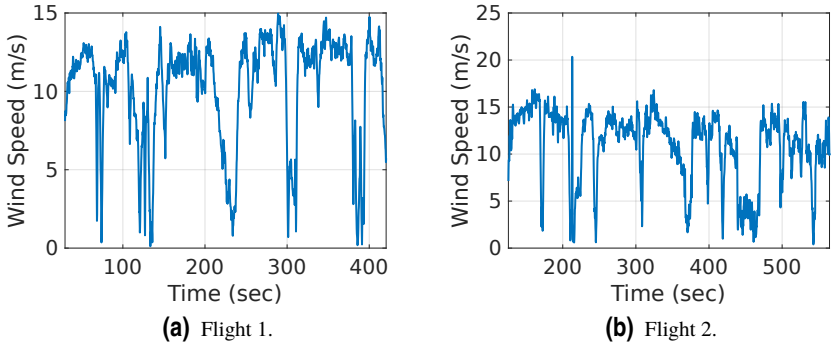


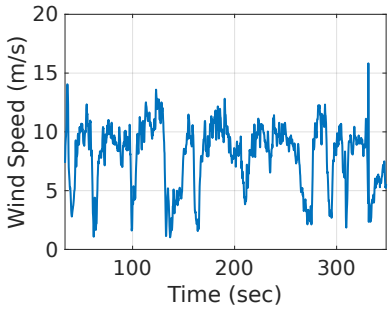
Figure 6.39 The heading vs. time plot of the first run of the tuning experiments, the jumps of the heading are due to the crossing the 180º and -180ª limit.

The corrected estimations were subject to a Weibull distribution analysis, Figure 6.40, from which it was determined the results that the estimates at the mission altitude are indeed Weibull-distributed data. And since the results show consistency with the weather

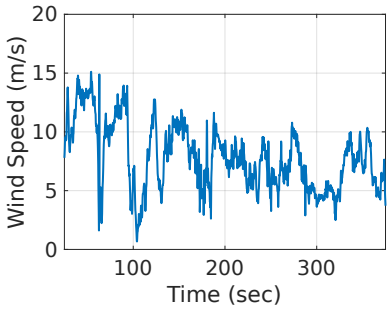


(a) Flight 1.

(b) Flight 2.



(c) Flight 3.



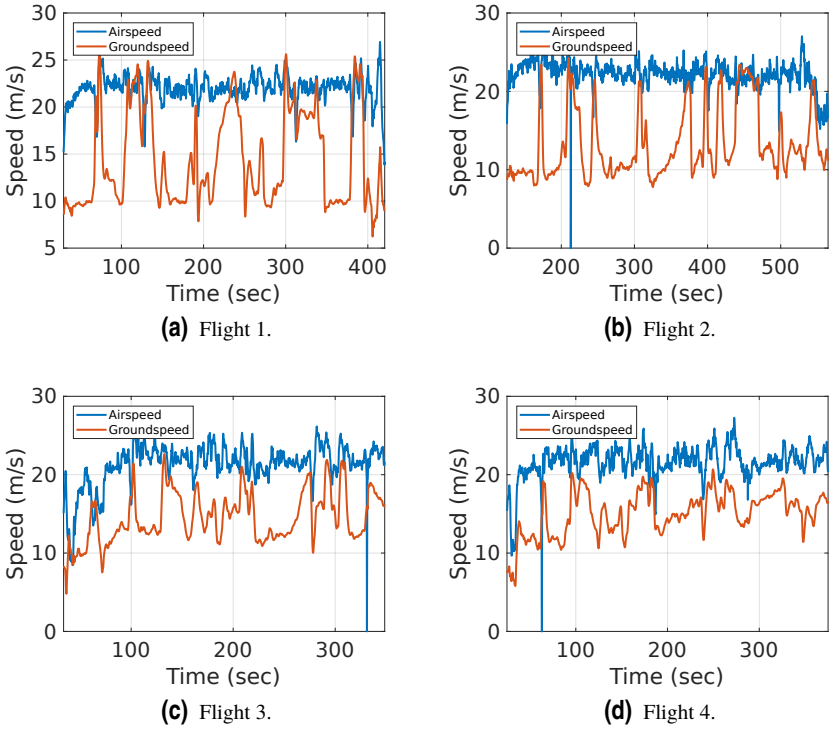
(d) Flight 4.

Figure 6.40 This picture shows the corrected estimation of the wind speed with the direct computation method for the four flights which trajectories were depicted in Figure 6.37. The estimation does not show less jumps but, this jumps now don't coincide with the heading results.

reports gathered from the near meteorological station, extrapolated to that altitude in accordance with the selected wind shear model, the results were validated. Moreover, a change in the wind algorithm was proposed in the PX4 developer site to fix the jumping problem and to allow a more accurate estimation of the airspeed components.

Firstly, to initially test the results of the wind estimation, the groundspeed and airspeed magnitude of the four experiments were plotted, to estimate if the wind estimations were more accurate. This is shown in Figure 6.41.

It can be observed that the groundspeed, directly taken from the GPS measurements does not follow the airspeed, which indicates the presence of a wind. The ups and downs of the groundspeed, suggest changes of the direction of the wind and also changes in the aircraft orientation with respect to the wind. The next stage of the analysis was the utilization of the running mean and standard deviation of the calculated wind speed. This



(a) Flight 1. **(b)** Flight 2.

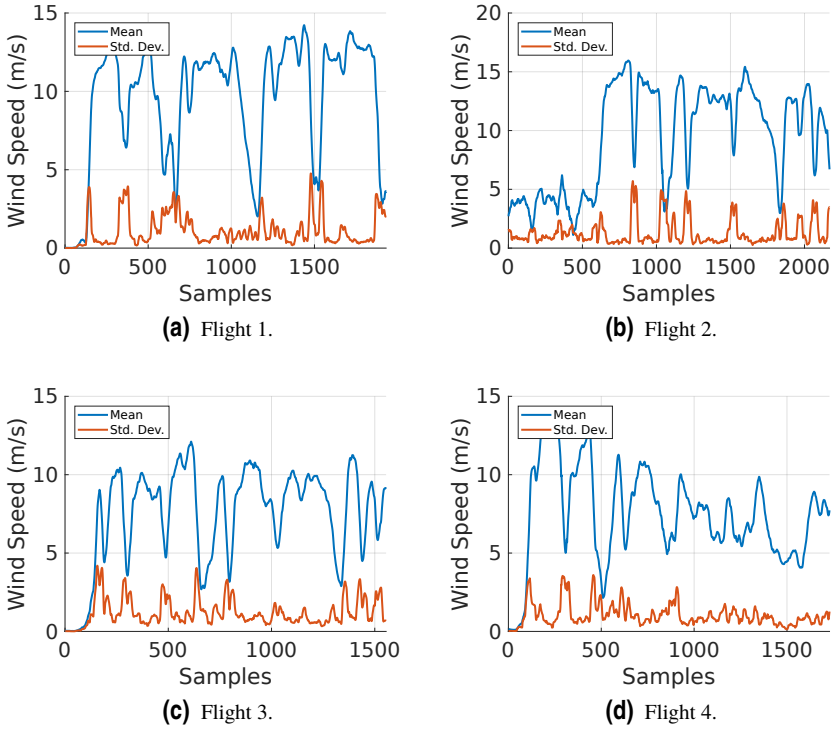
(c) Flight 3. **(d)** Flight 4.

Figure 6.41 This picture shows the groundspeed magnitude in orange and the airspeed magnitude in blue with the direct computation method for the four flights, with its trajectories were depicted in Figure 6.37.

is a crucial part for the feature detection, and since the behavior of the ground speed is different, it is expected that the mean and standard deviation will indicate the presence of gusts. Since the majority of the data is taken at constant altitude, any jump in the running standard deviation will indicate the presence of a gust. Once the gust is detected the feature will be characterized either by determining if the running mean returns to its original value, from which it will be considered a discrete gusts, or with the Gaussian Regression Process. An illustration of the mean and standard deviation of the wind speed magnitude of this flights can be observed in Figure 6.42.

When the peak of the standard deviation goes above 2 meters per second, the system raises an alarm of a potential feature. In this set of examples, there were various gusts identified that coincide to the prediction of the weather report for the nearest meteorological station.

The presence of gusts will indicate high variability of the actual estimates around a



(a) Flight 1.

(b) Flight 2.

(c) Flight 3.

(d) Flight 4.

Figure 6.42 This picture shows the running mean in blue and the standard and the running standard deviation for the four flights, with its trajectories were depicted in Figure 6.37. The running mean and standard deviation was calculated around 30 points for the last sample.

certain altitude. It is expected that the values will not be centered, but will be spread in accordance to the Weibull distribution, for the specific altitude. Although the dispersion may affect the prediction in the short term, being that one of the main drivers of the use of the regression process, which will provide with an understanding of the region stability to see if a generalized model for shear is sufficient to be able to get the maximum energy from a soaring maneuver. The wind/altitude maps of the aforementioned experiments can be observed in Figure 6.43.

The pictures observed in Figure 6.43 show great dispersion in the mission altitude, however, the lower altitude results resemble the shear behavior that is ideal for the soaring applications. The most stable behavior is observed in Flight 3 (see Figure 6.43c), in which dispersion is contained, except from an outlier point that is attributed to the sudden loss of airspeed information that occurs at around 360 seconds in Figure 6.41c.

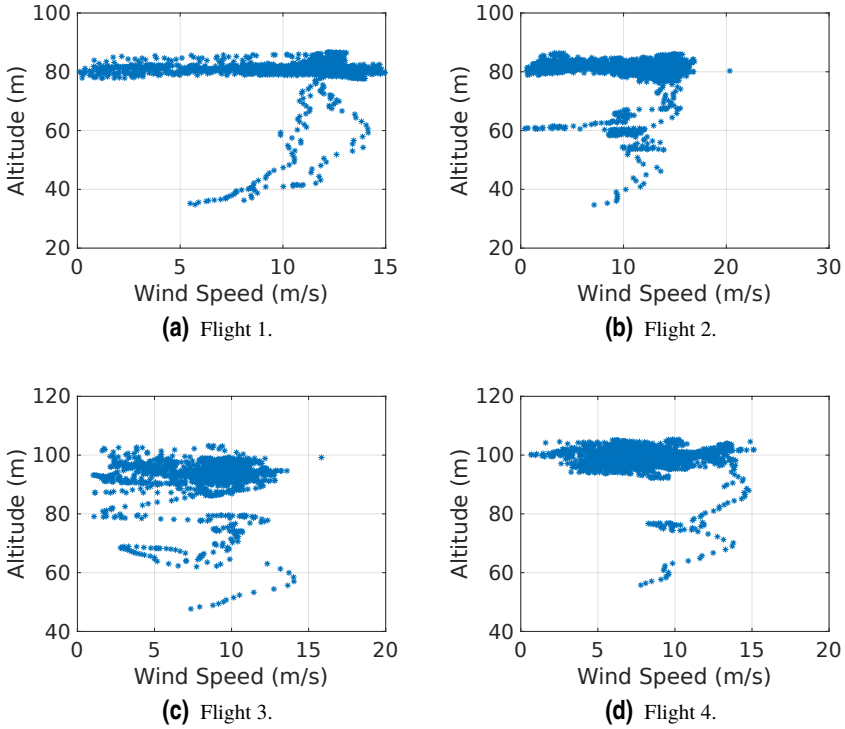
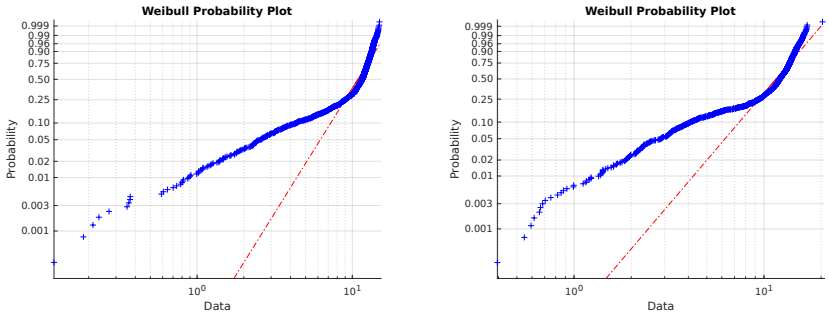


Figure 6.43 This picture shows wind vs. altitude maps of the experiments performed at Brenes for tuning the autopilot gains. All the flights show a different level of dispersion specially during the mission altitude. For the wind/altitude maps, the whole dataset is considered, starting from the wind estimations that are performed from the beginning of the ARMED state.

As it is mentioned before, it is expected that the data obtained with the airspeed measurements will fit into the Weibull distribution or at least a portion of it will be. In the long term, it is highly expected that the collected data at the same region will follow the Weibull distribution. The results for each flight, expressed as a Weibull probability plot can be observed in Figure 6.44

The experiments mentioned above give an insight of the capabilities of the FDES for the estimation of the wind velocity. Although the main purpose was to determine a set of adequate gain parameters for the path following in a surveillance mission, a side objective was to test the ability of the system to provide with wind measurements that will be good enough to determine if a reconfiguration is possible. During this process, different opportunities were found, specially in the airspeed component estimation in the PX4. For



(a) Flight 1. **(b)** Flight 2.

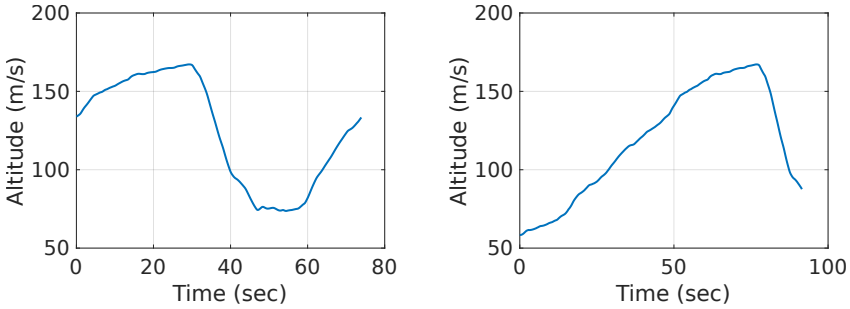
(c) Flight 3. **(d)** Flight 4.

Figure 6.44 The Weibull Probability plots for the tuning experiments, the experiments show that there are portions of data, that are distributed in accordance to the Weibull probability density function. In (a) and (b) there are portions that are away from the distribution, specially in low winds which is attributed to the insufficient data that a single flight can provide or simply to the sensors noise.

this, an update in the calculations was proposed to the PX4 source code and now it is incorporated in the newest firmware version, in which now a 3-dimensional estimation of the airspeed is performed, instead of the 2-dimensional approach shown in Code 2.1. The results show that the estimations correspond to the meteorological reports of the station for the 3 days of of experiments.

6.4 Autonomous swooping maneuver experiments

The next set of experiments were designed to be able to track a swooping curve with the tracking method presented in Chapter 4. The altitude profiles of two portions of the flight in which the tracking mode can be observed in Figure 6.45. The designed segment



(a) Flight segment 1. **(b)** Flight segment 2.

Figure 6.45 This figure shows the altitude profiles of two segments of a tracking experiment. The curve tracking was activated for about 75 seconds in (a) and for over 90 seconds in (a). After the execution of this maneuver, the stabilized flight mode was activated in which the curve tracking control presented in Chapter 4 was activated.

to be tracked consisted in the definition of a curve over 17 random waypoints that are similar to the swooping maneuvers observed in the flight of soaring birds. The goal was to determine the performance of the adaptive control while following a curve designed with the Hobby method presented in Chapter 3. The curve was generated off-board and intends to prove only the tracking part and not to analyze the wind process or the energy extraction functions.

The altitude profiles are similar to swooping patterns. The designed curves are in the 3D space, and due to the complexity the altitude profile has to be shown separately from the latitude/longitude plot. The latitude longitude plot that corresponds to the flight with altitude pattern depicted in Figure 6.45a is shown in Figure 6.46 and the one for the flight in Figure 6.45b is shown in Figure 6.47

The results show that the tracking system has a very good performance when it comes to track a complex curve. It can be seen that the errors are minimum and the performance is adequate for low winds. Note that both experiments were performed in the experimental site of Brenes and the wind disturbance may not affect very much the performance. Nevertheless, the system needs to be tested in coastal conditions to prove the actual performance with a more complex wind situation. The second test site, located in the south of Spain offer the ideal conditions to test the tracking algorithm and to measure its ability to reject a bigger wind disturbance. The results of the tracking algorithm, performed in the beach of El Terrón in Huelva, Spain can be observed in a latitude/longitude plot in Figure 6.48 and in a 3D position plot depicted in Figure 6.49.

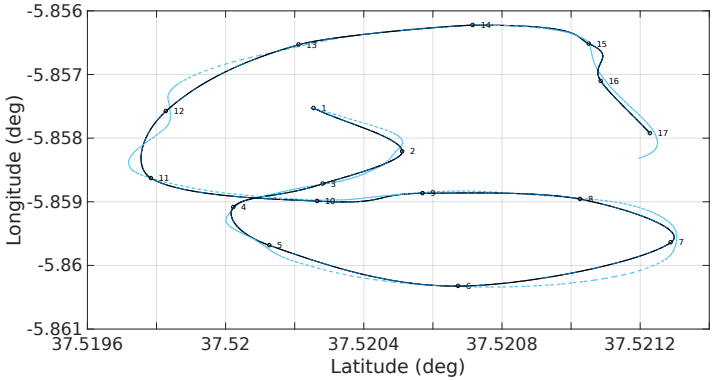


Figure 6.46 The picture shows the actual trajectory (dotted line) and the designed curve with the curve waypoints (solid line and black markers) for the first experiment.

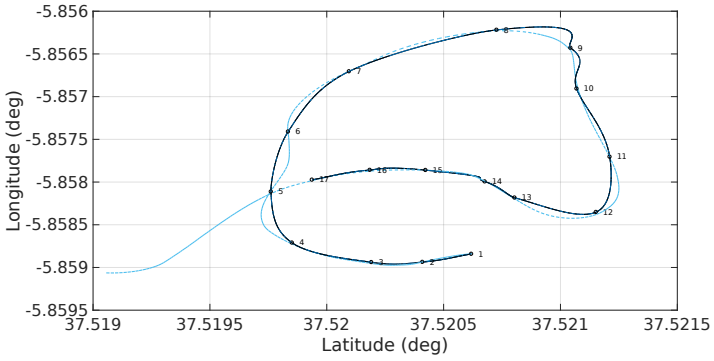


Figure 6.47 The picture shows the actual trajectory (dotted line) and the designed curve with the curve waypoints (solid line and black markers) for the second experiment.

In Figure 6.48 and Figure 6.49 it is observed that the system is still shows precision in a similar level than in the previous experiments. The flights performed in this region were executed with the Proskytec V1, which is a heavier platform and also more responsive than the SkySurfer X8. The results of the wind analysis, show more stable wind measurements (See Figure 6.50). In addition, the dispersion of the measurements among the altitude show a clear shear tendency (See Figure 6.51, which was identified and characterized by the system.

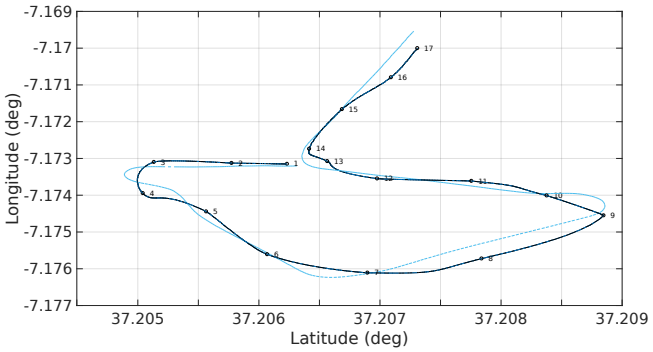


Figure 6.48 The picture shows the actual trajectory (dotted line) and the designed curve with the curve waypoints (solid line and black markers) for the first experiment.

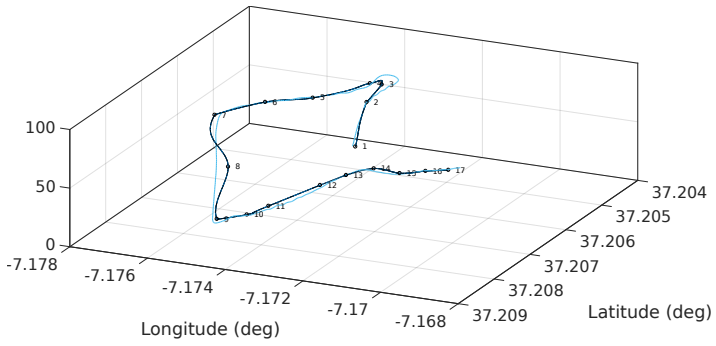


Figure 6.49 The picture shows the actual trajectory (dotted line) and the designed curve with the curve waypoints (solid line and black markers) for the second experiment.

6.5 Integration experiments

The last set of experiments consisted in the full integration of the FDES, starting from the sequence reconfiguration, the feature identification and the definition of legs for swooping maneuvers that included the trajectory generation and the tracking of the generated trajectories. The integrated system was designed to switch between the native PX4 controller, with the optimized tuning parameters, including a modification of the L1-control related parameters for better waypoint tracking, to the adaptive vector-field based control for curve tracking. This experiment consisted on a surveillance mission over an area limited by a crop field in the region. The number of waypoints was selected as 17 and the goal was to reconfigure the mission and to be able to execute a Rayleigh cycle

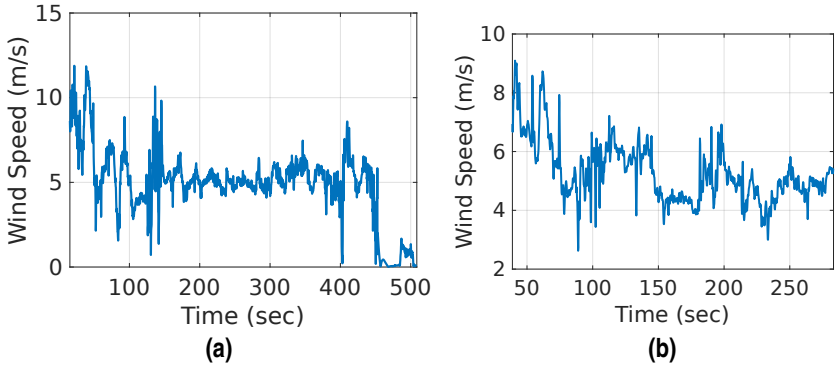


Figure 6.50 These results show the wind velocity estimations of two flights in the region of El Terron, in Huelva Spain, in which a more stable wind estimation is observed.

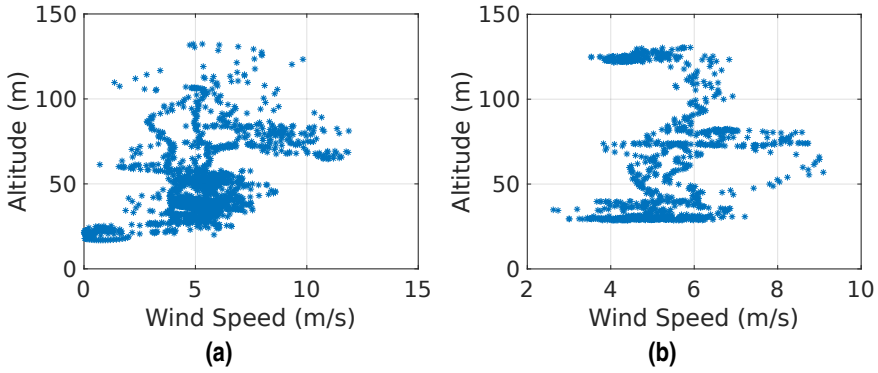


Figure 6.51 These results show the dispersion of the wind estimates with respect of the altitude. In both flights there is a clear tendency of a shear. Contrary to the experiments that were executed in Brenes, most of the measurements were collected at lower altitudes and in a broader spectrum. The wind estimates in (a) show a clearer tendency of increase with altitude, whereas in (b), this tendency is not completely appreciated because there are no sufficient measurements in higher altitudes.

among a predefined leg. The two missions are illustrated in Figure 6.52.

The swooping maneuver was performed between the waypoints 14 and 15 of the reconfigured mission, so only one trajectory was generated with the Hobby method. The position of the waypoints, including the auxiliary waypoints followed the following characteristics: a descent of 40 meters following a dive-in maneuver to the left of the leg, separated approximately 15 meters from the middle line of the leg, and the dive-out maneuver after

(a) (b)

Figure 6.52 Two missions performed with the Proskytec V1 in the region of Brenes, Seville. The mission in (a) shows a surveillance mission with 17 waypoints that go in spiral from the outside to the inside, which prevents overlapping. In (b) the same waypoints are reconfigured in accordance to the detected wind direction, which was reported to be coming from the south east, which is the lower left corner of the pictures.

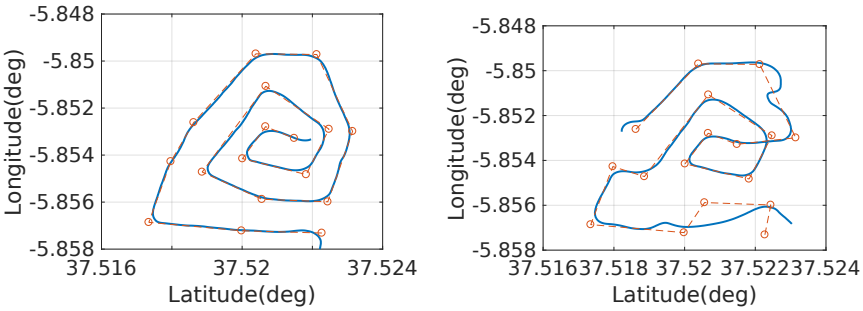


Figure 6.53 These pictures show the latitude and longitude profiles of the two missions presented in Figure 6.52. In (a), the system shows that there is a very accurate tracking among the different mission waypoints. (a) shows some poor tracking among the first and second waypoint, however, in the leg that merges the waypoint 14 and 15, the realization of the swooping maneuver is easily recognizable.

reaching the 40 meters to the right to reach other 15 meters from the middle line of the leg. The resultant trajectories are shown in Figure 6.53.

The experiment of Figure 6.53b shows that it experienced a difficulty in the tracking between the first and second waypoint, this is attributed to the velocity of the upcoming waypoint since the mission was executed not from ground but after the finalization of

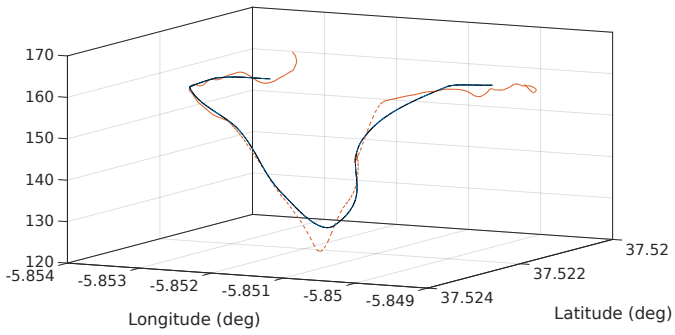


Figure 6.54 The picture shows the actual trajectory (dotted line) and the designed curve with the Hobby method (solid line and black markers respectively) for the first experiment.

the execution of the first mission. In the same figure, one can observe the Rayleigh cycle executed between the waypoints that are at the top right corner. Once the system reaches a threshold altitude after the dive out, the PX4 takes again full control with its native algorithm. To analyze the swooping maneuver, the next step is to present a detail of the trajectory while it was performed. This can be observed in Figure 6.54, in which the generated trajectory and the actual tracked trajectory can be observed.

In Figure 6.54, it can be observed that the aircraft was subject to slight shanking, specially at the beginning of the maneuver. This was attributed to the sensor noise and the wind disturbances, although the wind was oscillating between 2 m/s and 4 m/s, there were gusts that were reported to be over 40 knots. Additionally, after the experiment was performed it was found that the main source of instability was a poor balancing of the aircraft due to the removal of a piece and the front. The sensitivity of the system, specially to the center of gravity disturbances is quite high, therefore, there is a need to take extreme precautions, specially in high disturbances where the wind may affect significantly the controllability of the aircraft. Despite these facts, the performance was remarkable in terms of the tracking and also to recover the normal mode after the realization of the Rayleigh cycle.

After the repetition of this experiment over 5 times. It was observed that the voltage drop was higher in the mission without reconfiguration with an average gain of 14% in most of the cases with one that was executed with a considerable presence of a sustained gusts in the same wind direction that provided with a 19.5% improvement (see Figure 6.55).

The improvement in the discharge rate is attributed to the diminution of the thrust while executing the legs with a tailwind component and also to the energy gain while executing the Rayleigh cycle. Also, in the reconfigured path (see fig. 6.55b) there are portions in

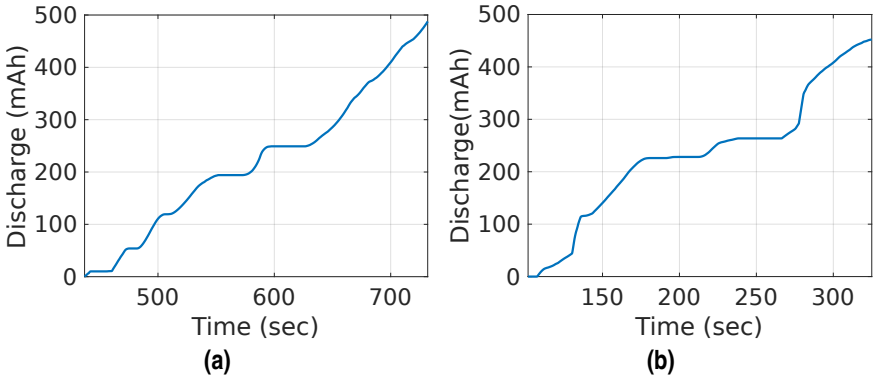


Figure 6.55 In this pictures, the accumulated discharge in mAh during the execution of the missions depicted in Figure 6.52. There is a considerable difference between the discharge presented in (a) with respect of (b), which is up to 50mAh lower..

which the accumulated discharge is kept almost constant. The rapid raise approximately at 275 seconds occurs while executing the dive-out maneuver, in which the thrust reaches 90% of its maximum value for recovering, however, since the recovery is of about 40 meters, the system keeps its tendency. Note that in both cases the discharge attributed to the takeoff, which occurs also at 90% of the throttle is omitted.

6.6 Chapter summary and conclusions

This chapter presented the simulations and experiments that were performed to validate the Flight Duration Enhancement System according to the requirements defined in Chapter 5. The objective of the tests was to cover different aspects on the development of the FDES, starting from the prototype phase of each function to the fully integrated system.

In the SITL and HITL tests, different techniques were utilized to minimize the impact on the experimental phase due to the changes in the functionality or due to the addition of new functions. The chapter presented a sequential approach of the tests starting from the simulation tests of the wind field estimation function and the characterization of the wind features,. The test activities were followed by the validation and verification of the trajectory generation and the trajectory tracking functionalities at simulation level, in which different aspects were considered, such as the interaction of the ocean and the atmosphere and how it will affect the various functions. The results presented provide with an excellent basis for the estimation of the different wind features, thanks to the offline experimental data that was utilized, and the utilization of external sources, such as the

weather reports, the simulation stage was completed successfully and showed promising results. Furthermore, the methodology provides with fundamentals on the integration of the scientific approach with the systems engineering approach which is key to the technology transfer.

The experiments campaign was the most rewarding phase of all the work of this research. There were more than 60 flights that were performed in two testing sites, with four experimental platforms (2 SkySurfers and 2 Proskytec V1) that provided with an immense amount of data, that were analyzed in detail in order to prove the various aspects of the FDES. Also the use of multiple platforms allowed to test the portability of the system and to explore different configurations, of the functionality. For instance, the SkySurfer weight limitation, lead to the implementation of some of the FDES features in the ground station, and instead of using the raw data from the autopilot, the telemetry data was utilized. This approach might be beneficial for the use of smaller platforms that could incorporate the duration enhancement functionalities. Also, during the experiments, numerous opportunities were identified, both at the platform level and at the system level in order to increase the performance of the duration enhancement algorithms. This allowed a gradual improvement in the results until reaching a full system integration.

7 Conclusions

How lucky I am to have something that makes saying goodbye so hard.

WINNIE THE POOH

This thesis presented the development of an integrated system that enhances the flight duration of small UAS's by employing different algorithms at three main stages: 1) the wind field estimation and the wind feature characterization, 2) the energy-efficient trajectory generation and 3) the trajectory tracking.

For the first stage, an extensive study based on meteorology studies was performed. Firstly, the estimation of wind was accomplished with the direct estimation method, which eases the calculations of the wind velocity and wind velocity rate of change. However, other means which use a Kalman Filter (KF) or a Moving Horizon Estimator (MHE) are analyzed, concluding that they could be utilized and potentially provide with more accurate estimations. Nevertheless, for computational reasons, and implementation time, the first was fully implemented in the system for the last experiments. The drawback of the direct computation method is that it requires the estimation of airspeed components from the autopilot. The main disadvantage of the use of a KF or an MHE is that the system needs continuous excitations at some level for convergence.

Regarding the identification of features, a novel method, which considers complex models for wind shear, discrete gusts, and continuous gusts was presented. A dataset of wind estimation is fed to an identification function that characterizes the models based on the analysis of the running mean and standard deviation. These models use a complex GA algorithm to model the distribution of the dataset; however, the use of GA does not affect the computation time, due to the design of the fitness function, that aims to determine only a single parameter. The system is also able to identify, based on the behavior of the standard deviation any potential gust feature that could be characterized by merely quantifying the change of wind speed (in the case of discrete gusts) but also on the distribution of the dataset. For the continuous gusts, a Gaussian Process Regression (GPR) was chosen

to provide with short-term predictions. Due to the lack of time, the GPR was not fully implemented, and up to this point in the research, the system is only able to provide with an alarm of the existence of a continuous gust. Besides, the models as mentioned earlier consider the regional stability and the surface roughness as input parameters to determine the capacity to extend the predictions to a broader area. The interaction of the atmosphere-ocean boundary layer was deeply analyzed, and the models consider this interaction to increase the accuracy. Also, the ability to understand the influence of the ocean on the formation of the features allow the determination of critical points in the space that could be utilized for a more efficient soaring.

In the case of the trajectory generation, the system considers the observation of the Albatross performed in previous research activities, and come up with a convenient manner to replicate these trajectories for an unmanned aerial system. Typically, these trajectories were treated as a circular; however, in reality, the paths are continuously changing in shapes and distances for a more efficient flight. Therefore, the system can determine the best possible curve for a given scenario with parametric design, providing with a 3D solution which considers certain platform limitations in the curve design. The curvature smoothness and curve velocity could be related to the platform turning rate and load factors, allowing the design of flyable trajectories. The results show that the curves are indeed adequate and also provide with a smooth path for the UAS even in extreme conditions, or in complex scenarios derived from airspace restrictions.

The tracking was a complicated process that was divided into two subprocesses. The first one considers a simple reconfiguration of the waypoint sequence of the existing mission to be able to maximize the energy on the system by flying longer distances with a favorable wind component. This is accomplished with a method of sequencing generation that weights the cells of a partitioned area according to user-defined parameters. Therefore, the weights can be adjusted to prioritize the neighboring cells that produce a positive component of the wind. This process can be executed as part of the mission planning but also as part of an online process that could recognize a change in the wind patterns and adjust the sequence accordingly. The soaring curves, namely Rayleigh cycles, have to be treated carefully. In the context of surveillance mission is unlikely that if these are executed continuously, the mission information preserves its integrity due to the continuous attitude changes. Therefore, only certain portions of the flight could be utilized to perform the Rayleigh cycles, so that the mission is not compromised but also saves energy overall. The tracking process of these Rayleigh cycles considered the use of a moving frame and the determination of the error of the aircraft vehicle with the attitude frame. The methodology utilized allowed a quick error resolution and also a rapid determination of the control

parameters that were fed to the autopilot, preventing latency issues and fast convergence. However, the control system still has room for improvement due to the lack of consideration of complex wind disturbances, such as gusts, and also needs the integration of a thrust control to prevent a massive discharge during the dive-out maneuvers of the Rayleigh cycles.

The system integration was a vital part of this work, and the chosen methodology simplified the selection of a testing methodology. The architecture utilized a well-known publisher and subscriber scheme with the meta-operating system ROS version Kinetic Kane, but architecture-wise the essential part of the design was that the primary goal was to prevent the function coupling, keeping the wind-related functions, the trajectory generation and the trajectory tracking functions completely independent. The selected architecture is a crucial part of the desired upgradeability and maintainability of the system, simplifying the bug detection and the solutions calculation. Each functionality was tested separately, at a prototype level and at the implementation level, which ensures a reliable system integration. The wind tests showed the different detection capabilities with convergence times of less than 30 seconds to find a solution on the feature identification, and also permits the integration of additional external data, such as weather reports and information of previous flights to enhance the prediction. The trajectory tests ensure that the system produces smooth and flyable trajectories that could be framed to generate the error equations. The tracking process, considering its complexity, allow a versatile way of finding a solution by using a two-stage approach, which ensures to keep the mission as the main priority of the system. The overall results of the integration tests prove that the system provides with energy savings that go up to 20%. Furthermore, the total energy gain depends a lot on the mission and may be able to diminish the energy loses in very harsh conditions. Coastal areas and oceanic missions often have these rough conditions and this system have proved to be efficient and reliable in such circumstances.

An important goal during the development of the FDES was the realization of a massive experiments campaign. This campaign was performed in various stages, including the functional prototyping, the simulations with a software-in-the-loop and hardware-in-the-loop benches and a series of over 60 real experiments with the use of different platforms and in different site and weather conditions. The tests included the evaluation of the functionalities in coastal areas and also in various meteorological scenarios. The results achieved at this stage are the substantial grounds on the foundation of the FDES, which proves that the system is not only feasible but provides with actual energy gains. This thesis shows only a small portion of the whole experimental campaign. The most significant results were presented and analyzed in this thesis, but the entire dataset that includes

the logs and calculations of all the experiments is available for consultation in an OSF repository.

The sensor noise continuously influences the system, and most importantly the wind variability, so it is strongly suggested to keep with the tests for full implementation on different platforms since the integration was optimized for specific ones. The whole tuning parameters and the determination of the aerodynamic characteristics of the system to feed the control system were based on the platform only. At this stage, if the system has to be moved to a different platform, an in-depth analysis has to be performed to keep the results at least with the current performance.

7.1 Perspectives and Future Work

There has been a considerable number of lessons learned with the design of the system, but also there are opportunities that were identified as future directions of research. These are discussed in the present section.

- The identification of the wind vector can be improved with the integration of an optimal estimator, preventing the dependability of angle of attack and sideslip angle measurement or to a means of estimating the airspeed components, which sometimes introduce unwanted noise. Even though the methods are discussed, these were not implemented into the final system, which would be a definitive step for the increase of the accuracy on the estimations. Nevertheless, means for dealing with the necessity of exciting specially the vertical components have to be taken care of, always considering the mission because the continuous changes on attitude have a significant impact on the footprint and as a result, in the coverage.
- The feature identification was proved to be efficient; however, the dependability on the accuracy of the model is critical and sometimes the experiments needed to be suspended due to the inadequate response on the system in very severe conditions. One idea that arose during the implementation of the algorithms was the utilization of machine learning techniques that will improve or remove altogether the dependability of complex models that are often not very accurate. Even though the utilized models show promising results, a self-identification on features that could potentially be harvested would be a massive step in the development of this framework.
- The continuous gust identification was not fully implemented, mainly due to the complexity of the algorithm and lack of time. However the use of another solution

for gust identification and exploitation could be considered to be integrated with the system suite.

- The 3D trajectory generation may not be sufficient for the mission parameters. The integration of UAS to either segregated or non segregated airspace. Therefore, an increase the precision on the trajectory not only in space but in time is required. The system could integrate a 4D trajectory generation system that considers not only the kinematic constraints but also the mission time, and even with that to find room for improvement with the energy harvesting. A cost function could be employed.
- The switching of control strategies while performing the Rayleigh cycles may not be the best strategy, especially in harsh environments. Therefore a more robust control strategy has to be explored, preventing from switching, but to adapting the control to the current leg type, e.g., if it is a straight test, the system can generate an optimal trajectory on a leg-to-leg basis and still be able to perform Rayleigh cycles in preselected legs.
- Regarding the architecture, two points have to be improved. The use of the ODROID XU4 permits that a multi-thread approach can be utilized regarding the processing of items. As of now, there are delays in the computation of the system since the request of a specific data of a function is the trigger of for its execution. Multi-threading will permit uninterrupted estimation of wind while at the same time identify the features and generate the trajectory. These minor delays are not that significant, but the impact was observed in some of the field experiments.
- The addition of the multi-UAS functionality is a necessary step to construct a complete suite for a real application. A single vehicle may not be able to perform a given mission by itself, but it may need to coordinate the information with other vehicles. The selected approach allow the scalability, and the use of COTS components will enable the reproduction of the system in others. The next functionality that could be added is the use of multiple UAS to extend the wind prediction capabilities and also to understand the impacts of the terrain changes to a broader prediction. For this, a commercial middle-ware for communications can be utilized, which will simplify the network integration process.

Appendix A
Software versions and hardware characteristics

This appendix shows the different software versions and hardware characteristics utilized in the development and testing of the FDES.

A.1 Software versions

The most relevant software versions utilized in the development and testing of the FDES are listed below. These versions supported the SITL and HITL testing as well as the full development and experiments in the design of the FDES.

1. MATLAB ©2016b (9.1.0.441655) and 2017a(9.2.0.538062)
2. QGroundControl Version 3.3.2. Stable Release
3. PX4 Firmware Version v1.7.3 Stable Release
4. XPlane 10
5. Ubuntu 17.04 (Zesty Zapus) and Ubuntu 16.04 (Xenial Xerus)
6. ROS Kinetic Kame and ROS Lunar Loggerhead
7. Mavlink 2 Specification
8. ODROID Ubuntu Minimal 16.04 (20170731)
9. Jetbrains PyCharm 2018.1.4 / 2017.1.3 / 2016.1.4
10. QtCreator 3.5.1
11. Qt 5.5.1 (GCC 5.2.1 20151129, 64 bit)
12. GCC Compiler Version 5.4.0 20160609
13. std:: C++ specification ISO/IEC 14882:2017
14. Python3 version 3.6.3
15. Python2 version 2.7.14

A.2 Hardware versions

Next, the hardware characteristics are enumerated in the following tables. For the autopilot, the characteristics are listed in Table A.1:

Table A.1 Pixhawk PX4 Relevant Characteristics.

CPU	32-Bit ARM Cortex M4 core with FPU
RAM	168 MHz/256 KB RAM/2 MB Flash
Coprocessor	Failsafe 32-Bit
Input Power	4.1V to 5.7V
Input Power (Servos)	4.1V to 5.7V
Input Power (USB)	4.1V to 5.7V
Serial Interface	5x UART serial ports
Satellite Input	DSM/DSM2/DSM-X Satellite input
S.Bus Input	Futaba

Figure A.1 The Pixhawk PX4 is an open source project that provides with high-end industry standard autopilot hardware for unmanned applications.

The characteristics of the ODROID-XU4, which is the main module of the duration enhancement system are enumerated in Table A.2.

Table A.2 ODROID-XU4 Relevant Characteristics.

CPU	Samsung Exynos5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core CPUs
RAM	2GB LPDDR RAM
Flash Storage	eMMC5.0
Power	5V/4A input
ADC	10bit SAR 2 channels
Size	83 x 58 mm (3.35 x 2.2 inch)
Weight	40g (1.41oz)

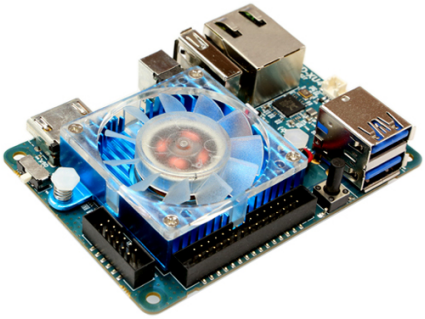


Figure A.2 The Hardkernel®ODROID-XU4 is a new generation of computing device.

In addition the PX4 utilized a weather board, which is a low-cost environmental sensing solution with the BME280 and Si1132. The BME280 is is an integrated environmental sensor developed specifically for mobile applications where size and low power consumption are key design constraints. The integrated sensor includes a humidity, pressure and temperature measurements. The SI1132 measures the UV intensity but is not utilized in this work.

Appendix B
Energy Estimation

The propulsion energy model $\bar{E}$ as a function of the trajectory $\mathbf{q}(t)$ can be estimated with the method presented originally by Zeng et al. [81], in which the following forces are considered:

- *Weight* $W = mg$, where m is the aricraft mass and g is the gravitational constant $g \approx 9.81 \mathrm{m\,s}^2$
- *Drag force D*
- *Lift force L*
- *Thrust force F*

The drag can b expressed as a function of the drag coefficient C_{D_0}

$$D = \frac{1}{2}\rho C_{D_0} S V_a^2 + \frac{2L^2}{\pi e_0 \mathscr{A}_{\mathscr{R}} \rho S V^2} \tag{B.1}$$

where ρ is the air density, C_{D_0} is the zero-lift drag coefficient, S is the reference area or the wing area, e_0 is the Oswald efficiency and $\mathscr{A}_{\mathscr{R}}$ is the aspect ratio of the wing.

The drag force can be rewritten as:

$$D = c_1 V_a^2 + c_2 \eta \frac{1}{V_a^2} \tag{B.2}$$

The energy required for a coordinated turn with airspee V_a and tangential acceleration a_T and centrifugal acceleration a_C is:

$$E = c_1 V_a^3 + \frac{c_2}{V_a}\left(1 + \frac{a_C^2}{g}\right) + m a_T V_a \tag{B.3}$$

From here, one can define the definition of the tangential and centrifugal accelerations with respect the acceleration vector **a** and the velocity vector **v**, so

$$a_T = \frac{\mathbf{a}^T \mathbf{v}}{|v|} \tag{B.4}$$

$$a_C = \sqrt{|\mathbf{a}|^2 - a_T^2} \tag{B.5}$$

Substituting (B.4) and (B.5) into (B.3) yields:

$$c_1 |\mathbf{v}|^3 + \frac{c_2}{|\mathbf{v}|} \left(1 + \frac{|\mathbf{a}|^2 - \frac{\mathbf{a}^T \mathbf{v}}{|v|}}{g^2} \right) + m\mathbf{a}^T \mathbf{v} \tag{B.6}$$

The total energy required, given a trajectory expressed in time $\mathbf{q}(t)$, integrating (B.6) the total energy over the trajectory $\bar{E}(\mathbf{q}(t))$ can be expressed as:

$$\begin{aligned} \bar{E}(\mathbf{q}(t)) = \int_0^T \left[c_1 \|\mathbf{v}(t)\|^3 + \frac{c_2}{\|\mathbf{v}(t)\|} (1 + \frac{\|\mathbf{a}(t)\|^2 - \mathbf{m}}{g^2}) \right] dt \\ + \frac{1}{2} m (\|\mathbf{v}(T)\|^2 - \|\mathbf{v}(0)\|^2) \end{aligned} \tag{B.7}$$

$$\mathbf{m} = \frac{(\mathbf{a}^T(t)\mathbf{v}(t))^2}{\|\mathbf{v}(T)\|^2} \tag{B.8}$$

$\mathbf{v}(t)$ corresponds to the first derivative of the trajectory function:

$$\mathbf{v}(t) \triangleq (\ddot{q})(t) \tag{B.9}$$

$\mathbf{a}(t)$ is the second derivative of the trajectory function:

$$\mathbf{a}(t) \triangleq (\dot{q})(t) \tag{B.10}$$

List of Figures

List of Tables

List of Codes

Bibliography

[1] A. Klimkowska, I. Lee, and K. Choi, "Possibilities of uas for maritime monitoring," in *The international Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016 XXIII ISPRS Congress*, vol. XLI-B1, 2016. [Online]. Available: https://goo.gl/M2iuCK

[2] "Steam - ship's sulfur trails emissions aerial measurements | esa's artes applications," European Space Agency. [Online]. Available: https://artes-apps.esa.int/projects/steam

[3] "AR5 LIFE RAY," Tekever. [Online]. Available: http://airray.tekever.com/ar5

[4] A. S. Biradar and S. B. S. Saripalli, "Wind Estimation and Effects of Wind on Waypoint Navigation of UAVs," Ph.D. dissertation, Arizona State University, 2014.

[5] P. Osenar, J. Sisco, and Catharine Read, "Advanced propulsion for small unmanned aerial vehicles. the role of fuel cell-based energy systems for commercial uavs," feb 2017. [Online]. Available: http://www.unmannedsystemstechnology.com/wp-content/uploads/2017/02/White-Paper-Fuel-Cell-Energy-Systems-for-UAVs.pdf

[6] K. C. Reinhardt, T. R. Lamp, J. W. Geis, and A. J. Colozza, "Solar-powered unmanned aerial vehicles," in *IECEC 96. Proceedings of the 31st Intersociety Energy Conversion Engineering Conference*, vol. 1, Aug 1996, pp. 41–46 vol.1.

[7] E. Cestino, "Design of solar high altitude long endurance aircraft for multi payload & operations," *Aerospace Science and Technology Cestino / Aerospace Science and Technology*, vol. 10, no. 10, pp. 541–550, 2006. [Online]. Available: www.elsevier.com/locate/aescte

[8] A. Chakrabarty and J. W. Langelaan, "Flight path planning for uav atmospheric energy harvesting using heuristic search," *AIAA Guidance, Navigation and Control Conference*, vol. Toronto, Ontario, Canada, 2010.

[9] M. J. Cutler, T. W. McLain, R. W. Beard, and B. Capozzi, "Energy harvesting and mission effectiveness for small unmanned aircraft," *AIAA Guidance, Navigation and Control Conference*, vol. Toronto, Ontario, Canada, 2010.

[10] J. A. Cobano, D. Alejo, S. Sukkarieh, G. Heredia, and A. Ollero, "Thermal detection and generation of collision-free trajectories for cooperative soaring uavs," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2948–2954, 2013.

[11] C. Gao, "Autonomous soaring and surveillance in wind fields with an unmanned aerial vehicle," Ph.D. dissertation, University of Toronto, 2015.

[12] Z. Ákosl, M. Nagy, S. Leven, and T. Vicsek, "Thermal soaring flight of birds and unmanned aerial vehicles," *Bioinspiration and Biomimetics*, vol. 5, no. 4, 2010. [Online]. Available: https://arxiv.org/pdf/1012.0434.pdf

[13] J. Wharington and R. M. I. of Technology (Australia), *Autonomous Control of Soaring Aircraft by Reinforcement Learning*. Royal Melbourne Institute of Technology, 1998. [Online]. Available: https://books.google.es/books?id=_-5yuAAACAAJ

[14] D. J. Edwards, "Implementation Details and Flight Test Results of an Autonomous Soaring Controller," *AIAA Guidance, Navigation, and Control Conference*, no. August, pp. 1–16, 2008. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.559.4025{&}rep=rep1{&}type=pdf

[15] C. Montella and J. R. Spletzer, "Reinforcement learning for autonomous dynamic soaring in shear winds," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. Chicago, IL, USA, 2014.

[16] J. Bird, J. Langelaan, C. Montella, J. SPletzer, and J. Grenestedt, "Closing the loop in dynamic soaring," *AIAA Guidance, Navigation, and Control Conference*, vol. National Harbor, MD, USA, 2014.

[17] J. Langelaan, "Biologically Inspired Flight Techniques for Small and Micro Unmanned Aerial Vehicles," *AIAA Guidance, Navigation and Control Conference*, no. August, pp. 1–13, 2008. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.2008-6511

[18] N. Langelaan, J.W.; Alley and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *Journal of Guidance Control and Dynamics*, vol. Vol. 34, pp. 109–117, 2011.

[19] J. W. S. Rayleigh and Rayleigh, "The Soaring of Birds," *Nature*, vol. 27, no. 701, pp. 534–535, 1883.

[20] C. J. Pennycuick, "The Flight of Petrels and Albatrosses (Procellariiformes), Observed in South Georgia and its Vicinity," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 300, no. 1098, pp. 75–106, 1982. [Online]. Available: http://rstb.royalsocietypublishing.org/cgi/doi/10.1098/rstb.1982.0158

[21] C. PENNYCUICK, *Gust Soaring as a Basis for the Flight of Petrels and Albatrosses (Procellariiformes).*, 2002. [Online]. Available: https://books.google.es/books?id=DHJRtwAACAAJ

[22] P. Lissaman, "Wind Energy Extraction by Birds and Flight Vehicles," *Technical Soaring*, vol. 31, no. Time, pp. 1–13, 2005.

[23] P. L. Richardson, "How do albatrosses fly around the world without flapping their wings?" *Progress in Oceanography*, vol. 88, no. 1-4, pp. 46–58, 2011. [Online]. Available: http://www.elsevierscitech.com/newsletters/Aquatic/Albatross{_}Richardson{_}PiO.pdf

[24] ——, "High-Speed Dynamic Soaring," *R/C Soaring Digest*, vol. 29, no. 4, pp. 36–49, 2012. [Online]. Available: https://pdfs.semanticscholar.org/7101/4341b9f825a2f5d0bb39787cf07023ba422d.pdf

[25] T. A. Johansen, A. Cristofaro, K. Sørensen, J. M. Hansen, and T. I. Fossen, "On estimation of wind velocity , angle-of-attack and sideslip angle of small UAVs using standard sensors," *2015 International Conference on Unmanned Aerial Systems, ICUAS*, 2015. [Online]. Available: http://folk.ntnu.no/torarnj/aoa{_}ssa.pdf

[26] A. Wenz, T. A. Johansen, and A. Cristofaro, "Combining model-free and model based angle of attack estimation for small fixed-wing uavs using a standard sensor suite," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 7-10, 2016, Arlington, VA USA., pp. 624–632.

[27] A. Wenz and T. A. Johansen, "Estimation of Wind Velocities and Aerodynamic Coefficients for UAVs using standard Autopilot Sensors and a Moving Horizon Estimator," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, FL, USA, 2017, pp. 1267–1276.

[28] ——, "Icing detection for small fixed wing UAVs using inflight aerodynamic coefficient estimation," in *2016 IEEE Conference on Control Applications (CCA)*. IEEE, sep 2016, pp. 230–236. [Online]. Available: http://ieeexplore.ieee.org/document/7587841/

[29] T. Larrabee, H. Chao, Y. Gu, and M. R. Napolitano, "Wind field estimation in UAV formation flight," *American Control Conference*, vol. Portland,, pp. 5408–5413, 2014.

[30] W. Premerlani, "IMU Wind Estimation (Theory)," Tech. Rep., 2009. [Online]. Available: https://gentlenav.googlecode.com/files/WindEstimation.pdf

[31] F.-J. Liu, P.-H. Chen, S.-S. Kuo, D.-C. Su, T.-P. Chang, Y.-H. Yu, and T.-C. Lin, "Wind characterization analysis incorporating genetic algorithm: A case study in Taiwan Strait," *Energy*, vol. 36, no. 5, pp. 2611–2619, 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0360544211000788

[32] M. J. Afridi, A. J. Awan, and J. Iqbal, "AWG-detector: A machine learning tool for the accurate detection of anomalies due to wind gusts (AWG) in the adaptive altitude control unit of an aerosonde unmanned aerial vehicle," in *Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications, ISDA'10*, 2010, pp. 1125–1130.

[33] D. Yeo, V. Hrishikeshavan, and I. Chopra, "Gust Detection and Mitigation on a Quad Rotor Biplane," *AIAA Atmospheric Flight Mechanics Conference*, no. January, pp. 1–11, 2016. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2016-1531

[34] J. Etele, "Overview of Wind Gust Modelling with Application to Autonomous Low-Level UAV Control," Tech. Rep. November, 2006.

[35] R. Bencatel, P. Kabamba, and A. Girard, "Perpetual Dynamic Soaring in Linear Wind Shear," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 5, pp. 1712–1716, 2014. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.G000425

[36] J. J. Acevedo, N. R. J. Lawrance, B. C. Arrue, S. Sukkarieh, and A. Ollero, "Persistent monitoring with a team of autonomous gliders using static soaring," *IEEE International Conference on Intelligent Robots and Systems*, no. Iros, pp. 4842–4848, 2014.

[37] J. H. A. Clarke and W. H. Chen, "Trajectory generation for autonomous soaring UAS," *International Journal of Automation and Computing*, vol. 9, no. 3, pp. 248–256, 2012.

[38] D. Piggott, *Gliding: A Handbook on Soaring Flight*, ser. Flying and Gliding. A & C Black, 2002. [Online]. Available: https://books.google.es/books?id=RsJzPwAACAAJ

[39] R. Almgren and A. Tourin, "Optimal soaring via Hamilton-Jacobi-Bellman equations," *Optimal Control Applications and Methods*, vol. 36, no. 4, pp. 475–495, 2015. [Online]. Available: https://www.courant.nyu.edu/{~}almgren/papers/glider.pdf

[40] M. J. Allen and V. Lin, "Guidance and Control of an Autonomous Soaring UAV," Tech. Rep. September, 2007. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20070022339.pdf

[41] N. R. J. Lawrance, "Autonomous Soaring Flight for Unmanned Aerial Vehicles," Ph.D. dissertation, The University of Sidney, 2011.

[42] N. Akhtar, J. F. Whidborne, and A. K. Cooke, "Real-time trajectory generation technique for dynamic soaring UAVs," pp. 1–6, 2008.

[43] Y. J. Zhao, "Optimal patterns of glider dynamic soaring," *Optimal Control Applications and Methods*, vol. 25, no. 2, pp. 67–89, 2004.

[44] Y. Zhu, X. Chen, and C. Li, "A Moving Frame Trajectory Tracking Method of a Flying-Wing UAV Using the Differential Geometry," *International Journal of Aerospace Engineering*, vol. 2016, 2016.

[45] M. Syama, M. Dineshkumar, and W. C. Arun Kishore, "Trajectory optimization of an autonomous dynamic soaring UAV," *2015 International Conference on Control, Communication and Computing India, ICCC 2015*, no. November, pp. 95–100, 2016.

[46] G. C. Bower, "Boundary Layer Dynamic Soaring for Autonomous Aircraft: Design and Validation," Ph.D. dissertation, Stanford University, 2011. [Online]. Available: https://goo.gl/gNoS1S

[47] T. Flanzer, G. Bower, and I. Kroo, "Robust Trajectory Optimization for Dynamic Soaring," *AIAA Guidance, Navigation, and Control Conference*, no. August, pp. 1–22, 2012. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2012-4603

[48] C. X. Miao and J. C. Fang, "An adaptive three-dimensional nonlinear path following method for a fix-wing micro aerial vehicle," *International Journal of Advanced Robotic Systems*, vol. 9, 2012.

[49] "Marineuas- innovative training network on autonomous unmanned systems for marine and coastal monitoring," EU Marie Sklodowska-Curie Actions, 2015. [Online]. Available: http://www.marineuas.eu/

[50] *Military Specification MIL-F-8785C, Flying Qualities of Piloted Airplanes*, U.S. Department of Defense, 5 November 1980.

[51] J. R. Holton, *An Introduction to Dynamic Meteorology*, 4th ed. Burlington, MA :: Elsevier Academic Press, 2004. [Online]. Available: http://fama.us.es/record

[52] M. B. Paskyabi and I. Fer, "Ocean near-surface boundary layer: processes and turbulence measurements," *REPORTS IN METEOROLOGY AND OCEANOGRAPHY UNIVERSITY OF BERGEN*, vol. 1, 2010. [Online]. Available: http://web.gfi.uib.no/publikasjoner/rmo/RMO-2010-1.pdf

[53] K. V. S. Namboodiri, D. P. Krishnan, R. K. Nileshwar, K. Mammen, and N. Kiran, "Coastal Boundary Layer Characteristics of Wind , Turbulence , and Surface Roughness Parameter over the Thumba Equatorial Rocket Launching Station , India," *Journal of Climatology*, vol. 2014, 2014.

[54] M. Y. Leclerc, T. Foken, M. J. Savage, and M. Göckede, *Footprints in micrometeorology and ecology*, 2014.

[55] V. Bonnin, "From albatrosses to long range uav flight by dynamic soaring," Ph.D. dissertation, University of the West of England, 2016. [Online]. Available: https://goo.gl/p5Wvbf

[56] D. J. T. Carter, "Prediction of Wave Height and Period for A Constant Wind Velocity Using the JONSWAP Results," *Ocean Engineering*, vol. 9, no. l, pp. 17–33, 1982.

[57] C. Park, J. Huang, and Y. Ding, "Domain decomposition for fast gaussian process regression," *Journal of Machine Learning Research*, vol. 12, pp. 1697–1728, 2011.

[58] S. Lerch and T. L. Thorarinsdottir, "Comparison of nonhomogeneous regression models for probabilistic wind speed forecasting," *Tellus A*, vol. 65, 21206, 2013.

[59] G. Sachs, "Minimum shear wind strength required for dynamic soaring of albatrosses," *Ibis*, vol. 147, no. 1, pp. 1–10, 2005.

[60] T. P. Chang, "Wind energy assessment incorporating particle swarm optimization method," *Energy Conversion and Management*, vol. 52, no. 3, pp. 1630–1637, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.enconman.2010.10.024

[61] C. Carrillo, J. Cidrás, E. Díaz-Dorado, and A. F. Obando-Montaño, "An approach to determine the weibull parameters for wind energy analysis: The case of Galicia (Spain)," *Energies*, vol. 7, no. 4, pp. 2676–2700, 2014.

[62] J. D. Wilson, "Statistics of the Wind-Speed Difference Between Points with Cross-Wind Separation," *Boundary-Layer Meteorology*, vol. 146, no. 1, pp. 149–160, 2013.

[63] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[64] "Dynamic soaring: How the wandering albatross can fly for free," Institute of Flight System Dynamics, 2018. [Online]. Available: https://www.youtube.com/watch?v=uMX2wCJga8g

[65] P. Carpenter. (2018) Dynamic soaring with rc gliders. [Online]. Available: https://www.rc-airplane-world.com/dynamic-soaring.html

[66] J. D. Hobby, "Smooth, easy to compute interpolating splines," Department of Computer Science Stanford University, Tech. Rep. STAN-CS-85-1047, 1985. [Online]. Available: https://goo.gl/9wQPwO

[67] F. Balampanis, I. Maza, and A. Ollero, "Coastal areas division and coverage with multiple uavs for remote sensing," *Sensors*, no. 17-4, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/4/808

[68] M. Yvinec, "2D triangulation," in *CGAL User and Reference Manual*, 4.7 ed. CGAL Editorial Board, 2015. [Online]. Available: http://doc.cgal.org/4.7/Manual/packages.html#PkgTriangulation2Summary

[69] F. Balampanis, A. P. Aguiar, I. Maza, and A. Ollero, "Path tracking for waypoint lists based on a pure pursuit method for fixed wing UAS," in *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2017 Workshop on*, 2017, pp. 55–59.

[70] C. Silvestre, A. Pascoal, and I. Kaminer, "On the design of gain-scheduled trajectory tracking controllers," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 9, pp. 797–839, 2002.

[71] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Vehicles*, 1st ed. John Wiley and Sons, Ltd., 2010. [Online]. Available: http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470741295.html

[72] D. R. Nelson, D. B. Barber, T. W. Mclain, and R. W. Beard, "Vector Field Path Following for Small Unmanned Air Vehicles," *American Control Conference (ACC)*, vol. 23, no. 3, pp. 519–529, 2006.

[73] D. Carroll, E. Köse, and I. Sterling, "Improving Frenet's Frame Using Bishop's Frame," 2013. [Online]. Available: https://arxiv.org/pdf/1311.5857.pdfhttp://arxiv.org/abs/1311.5857

[74] W. Jung, S. Lim, D. Lee, and H. Bang, "Unmanned Aircraft Vector Field Path Following with Arrival Angle Control," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 84, no. 1-4, pp. 311–325, 2016.

[75] "Pixhawk Flight Control," Dronecode, Computer Vision and Geometry Lab. [Online]. Available: https://pixhawk.org/

[76] Q. Control, *MAVLink Micro Air Vehicle Communication Protocol*, 2016.

[77] B. Michini and J. P. How, "L1 adaptive control for indoor autonomous vehicles: Design process and flight testing," *Proceeding of AIAA Guidance, Navigation, and Control Conference*, no. August, pp. 1–15, 2009.

[78] "JSBSim Open Source Flight Dynamics Model." [Online]. Available: http://jsbsim.sourceforge.net/

[79] "QGC - QGroundControl - Drone Control," Dronecode Project Inc. [Online]. Available: http://qgroundcontrol.com/

[80] jDrones, *Digital airspeed sensor*, 2016.

[81] Y. Zeng and R. Zhang, "Energy-Efficient UAV Communication with Trajectory Optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[82] *AEROSIM Aeronautical Simulation Blockset V1.2, User's Guide*, Unmanned Dynamics, 2005.

[83] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 6235–6240, 2015.

[84] S. Park, J. Deyst, and J. P. How, "A New Nonlinear Guidance Logic for Trajectory Tracking," *Proceedings of the AIAA Guidance Navigation and Control Conference*, no. August, pp. 1–16, 2004. [Online]. Available: http://acl.mit.edu/papers/gnc{_}park{_}deyst{_}how.pdf

Index

Glossary

ABL Atmospheric Boundary Layer. 14, 21, 153

ADC Air Data Computer. 80, 81

AEH Atmospheric Energy Harvesting. 2, 3, 5, 31, 79

AHRS Attitude and Heading Reference System. 32

CDF Cumulative Density Function. 33, 36

CDT Constrained Delauney Triangulation. 60, 63

COTS Commercial-Off-The-Shelf. 5, 9, 30, 73, 81, 83, 145

DCM Direction Cosine Matrix. 5, 30

ECAD European Climate Assessment and Dataset. 63

ECEF Earth-Centered-Earth-Fixed. 15

EKF Extended Kalman Filter. 5, 31, 37

FAA Federal Aviation Administration. 3

FDES Flight Duration Enhancement System. 2, 79–81, 83, 84, 87, 88, 90, 91, 93, 95, 120, 126, 139, 140

GA Genetic Algorithm. 5, 34–37, 99, 107, 153

GNSS Global Navigation Satellite System. 4, 13, 30, 37

GPR Gaussian Process Regression. 7

HITL Hardware-In-The-Loop. 91–93, 95, 139

IDE Integrated Development Environment. 81

IMU Inertial Measurement Unit. 4, 13, 29, 30, 37

J Root Mean Square Error. 36

METAR Meteorological Terminal Aviation Routine Weather Report. 63

MHE Moving Horizon Estimator. 5, 31

NED North-East-Down. 15, 16, 101

NOAA National Oceanic and Atmospheric Administration. 63

PBL Planetary Boundary Layer. 14, 18

PDF Probability Density Function. 33, 99

RPAS Remote-Piloted Aircraft System. 61

SITL Software-In-The-Loop. 91–93, 95, 108, 139

TECS Total Energy Control System. 90

TMO Translational Motion Observer. 32

UART Universal asynchronous receiver-transmitter. 83, 148

UAS Unmanned Aerial System. 1–6, 9, 14, 19, 20, 43, 61, 75, 80–82, 88, 96, 141

UAV Unmanned Aerial Vehicle. 2, 3, 13, 15, 16

UKF Unscented Kalman Filter. 5