ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MODELING, SIMULATION, AND TEST FLIGHT VALIDATION OF A FIXED-WING UNMANNED AIRCRAFT SYSTEM

M.Sc. THESIS

Ertuğrul Barış ÖNDEŞ

Department of Astronautics and Aeronautics Engineering

Astronautics and Aeronautics Engineering Programme

JUNE 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MODELING, SIMULATION, AND TEST FLIGHT VALIDATION OF A FIXED-WING UNMANNED AIRCRAFT SYSTEM

M.Sc. THESIS

Ertuğrul Barış ÖNDEŞ
(511171110)

Department of Astronautics and Aeronautics Engineering

Astronautics and Aeronautics Engineering Programme

Thesis Advisor: Asst. Dr. İsmail BAYEZİT

JUNE 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SABİT KANATLI BİR İNSANSIZ HAVA ARACININ MODELLENMESİ SİMÜLASYONU VE TEST UÇUŞLARIYLA DOĞRULANMASI

YÜKSEK LİSANS TEZİ

Ertuğrul Barış ÖNDEŞ
(511171110)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Asst. Dr. İsmail BAYEZİT

HAZİRAN 2019

Ertuğrul Barış ÖNDEŞ, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 511171110 successfully defended the thesis entitled "MODELING, SIMULATION, AND TEST FLIGHT VALIDATION OF A FIXED-WING UNMANNED AIRCRAFT SYSTEM", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Dr. İsmail BAYEZİT** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Fikret ÇALIŞKAN** Istanbul Technical University	.....
	Asst. Dr. Muhammed GARİP Yildiz Technical University	.....

Date of Submission : **03 May 2019**
Date of Defense : **14 June 2019**

To my family,

FOREWORD

Firstly, I would like to thank my advisor Dr. İsmail Bayezit for his continuous support. We have been working together for almost 4 years, and it was a great pleasure for me to have him as my advisor during undergraduate and also graduate studies.

I am particularly grateful for the assistance given by my friends Emre Saldıran, Fırat Turgut, Tamer Şener and Yusuf Demiroğlu. We spent the last 4 years together in and out of the campus with memorable moments. Also, their work ethic has always been influential.

I would like to express my very great appreciation to all members of Team Hedef that we worked together and achieved not only the championship but also the honor by presenting our work at an international competition.

I should mention the support of Faculty of Aeronautics and Astronautics by providing us with the workshops. Also, I am grateful to the sponsors of the Team Hedef such as ASELSAN, TAI, TEI, HAVELSAN, Green & Black Logistics. During the test flights, Hezarfen RC Model Club always provided us with their flight field, especially the supports of Salih ağabey and Hadi Dostgül were valuable.

Finally, I wish to thank my family and my fiancé, Kübra Sekmen, for their continuous support and encouragement throughout my study.

June 2019

Ertuğrul Barış ÖNDEŞ
Aeronautical Engineer &
Control and Automation Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

AC	: Alternative Current
CFD	: Computational Fluid Dynamics
CoG	: Center of gravity
DC	: Direct Current
EKF	: Extended Kalman Filter
ESC	: Electronic Speed Controller
FBW	: Fly-by-wire
FCU	: Flight Control Unit
FW	: Fixed-Wing
GCS	: Ground Control Station
GPS	: Global Positioning System
IMechE	: Institution of Mechanical Engineers
IMU	: Inertial Measurement Unit
LtA	: Lighter-than-Air
NED	: North-East-Down
PID	: Proportional-Integral-Derivative
PWM	: Pulse Width Modulated
RC	: Radio-Control
RF	: Radio Frequency
RPAS	: Remotely Piloted Aircraft System
ROS	: Robot Operating System
RTK	: Real Time Kinematic
RW	: Rotary-Wing
SITL	: Software-in-the-loop
TECS	: Total Energy Control System
UAS	: Unmanned Aircraft System
UAV	: Uninhabited Aerial Vehicle
VLM	: Vortex Lattice Methods

SYMBOLS

ϕ, θ, ψ	: Roll, Pitch and Yaw Angle
L, M, N	: Roll, Pitch and Yaw Moments
f_x, f_y, f_z	: Forces on the X, Y and Z axes
u, v, w	: Linear Velocities
p, q, r	: Roll, Pitch and Yaw Angular Velocities
F_{lift}, F_{drag}, F_y	: Lift, Drag and Side Forces
$\delta_a, \delta_e, \delta_r$	: Aileron, Elevator and Rudder Deflections

LIST OF TABLES

Page

LIST OF FIGURES

Page

MODELING, SIMULATION, AND TEST FLIGHT VALIDATION OF A FIXED-WING UNMANNED AIRCRAFT SYSTEM

SUMMARY

Unmanned Aircraft Systems (UAS) gain even more interest nowadays, and their application area is very broad. Due to eliminating the risks of human life, their usage in the military increases every year, and also the civil market expands its usage in agriculture, aerial mapping, cargo delivery, etc. The authorities are looking forward to certificating these systems safely to be used in urbanized areas as well, which brings the researchers in a critical position to provide them with satisfactory systems. Therefore, assessing the risks in a computer environment, and creating regulations and rules based on these risk is important, which brings us to the point where the modeling and simulation has a crucial role.

There exist different types of UAS such as fixed-wing, rotary-wing, flapping-wing, etc. Their advantages/drawbacks vary according to their mission requirements. The holy-grail of these systems are the fixed-wing versions due to their aerodynamically efficient configuration, and in this work, a custom-built UAS developed by Team Hedef is studied. The UAS is named as Tulpar which belongs to Small UAS (SUAS) category, and it is developed for two missions: payload delivery and reconnaissance. These missions are designed for UAS Challenge 2017 & 2018 by the professionals at the Institution of Mechanical Engineers (IMechE), and they are a part of the natural disaster scenario that might occur in coastal areas.

The main objective of this thesis is to develop a high-fidelity simulation environment in MATLAB/Simulink to evaluate the flight dynamics of a new designed Small UAS considering its flight stability and control and also the mission performance to optimize the equipment selection and also the aerodynamic sizing of the aircraft for various configurations. To validate the developed mathematical model, Tulpar UAS is used as a testbed which is designed by ITU Team Hedef and after the validation, it is aimed to use that UAS in graduate or undergraduate level courses to make students gain hands-on experience on the UASs.

The mathematical modeling foundation is introduced starting from the reference frames, and then the flight dynamics, forces and moments of UAS are explained briefly. The aerodynamic model is obtained using these dynamics and also using the aerodynamic coefficients and stability derivatives that are obtained from XFLR5. Alternative approaches and analysis programs are covered briefly as well. Also, the subsystems' models are given for propulsion systems and actuators. They are obtained experimentally and numerically combined considering the time limitation for the detailed experiments. Finally, the environment model including gravity, atmosphere, and the wind is introduced, and they are given in detail with look-up tables so that the developed model can be used not only for UAS but also for other types of aerial vehicles, too.

The flight control system, also used in Tulpar, is introduced and modeled in MATLAB/Simulink environment. Autopilot software is chosen to be ArduPilot due to its open-source development, and the control scheme in Simulink is directly obtained by reading the code structure on GitHub and also from the previous similar works. Due to having a widespread community consists of experienced users and its open-source structure, Ardupilot is thought to be one of the best choices in the literature.

Autopilot hardware is chosen to be Pixhawk 2.1 mainly due to its relatively low-cost price and being suitable for Ardupilot as well. Pixhawk 2.1 has low-cost sensor selections as well for airspeed, navigation, and laser distance measurements, which makes it a convenient choice for this application. The communication system is introduced by selection methodology followed, too. Telemetry modules are chosen to be work at 868 MHz, and their power settings are checked according to the regulations.

The mathematical model is simulated in MATLAB/Simulink environment. Before the simulation, the trim condition is calculated using the small perturbations theory for the given flight condition and features. In addition to custom visualization figure in Simulink, Flight Gear is introduced to be connected as an extra option. Besides that, several simulation environments are mentioned with their advantages/drawbacks; however, mainly Simulink's visualization figure has been used to eliminate the time-consuming tasks such as converting the aircraft drawings in computer-aided drawing software to the required versions. The simulation provides the UAS designer with information about the flight characteristics of the custom design vehicles. The flight analysis is not covered in this work, but it can be included easily in the simulation environment to have a feature to plan the missions with better approximations as well. Also, the simulation environment can be used as a pilot training tool as well by adding a USB Controller, which is another future work of this study.

In the final part, the developed model's simulation results are compared with test flights as mentioned in the objectives of the thesis. Additional equipment used during the test flights is introduced and the test flight scenarios are explained. The test flights are conducted at Hezarfen airport according to these scenarios, and the results are compared with the simulation. Simulation is the ideal environment, and since it is obtained via modeling with a lot of assumptions and approximations, there exists a difference with real-life results. Especially considering the aerodynamic model details, the analysis programs such as XFLR5 do not calculate the UAS characteristics with high accuracy, because of the vortex lattice method (VLM), and especially the drag model brings errors. Also, the platform is manufactured by ITU Hedef UAS Team with low-cost facilities and limited calendar time, so there exist minor differences due to that condition, also. On the other hand, it is aimed to publish the source code and other related files as open-source in the future, and this study will be a basis for that purpose, too.

In this study, it is also aimed to share the experiences during this project work with others, so the reader can find additional information in the appendices about SUAS operations, such as checklists, lessons learned, additional resources.

SABİT KANATLI BİR İNSANSIZ HAVA ARACININ MODELLENMESİ SİMÜLASYONU VE TEST UÇUŞLARIYLA DOĞRULANMASI

ÖZET

İnsansız Hava Araçları (İHA) bugünlerde daha fazla ilgi görüyor ve uygulama alanları oldukça geniştir. İnsan hayatının risklerini ortadan kaldırması nedeniyle, askeriyede kullanımları her yıl artmakta ve sivil pazarda da tarım, hava haritalaması, kargo teslimi vb. alanlarındaki kullanımını genişletmektedir. Otoriteler bu araçları yerleşim alanlarında da kullanılmak üzere güvenli bir şekilde sertifikalandırmak istemektedir ve bu durum araştırmacıları güvenlik açısında tatmin edici sistemler geliştirmek üzere kritik bir konuma getiriyor. Bu nedenle, riskleri bilgisayar ortamında değerlendirmek ve bu riske dayanan düzenlemeler ve kurallar oluşturmak önem kazanıyor.

Sabit kanatlı, döner kanatlı, kanatlı kanat gibi farklı İHA tipleri mevcuttur. Avantajları/dezavantajları görev gereksinimlerine göre değişiklik gösteriyor. Bu araçlar içinde aerodinamik olarak verimli konfigürasyonları nedeniyle önemli bir yere sahip olan tür ise sabit kanatlı versiyonlardır. Bu çalışmada, İTÜ Hedef Takımı tarafından geliştirilen özel yapım bir İHA incelenmiştir. Tulpar isimli bu İHA, küçük boyuttaki İHA sınıfına ait olup yük teslimatı ve yer gözlemi olmak üzere iki görev için geliştirilmiştir. Bu görevler, İngiliz Makine Mühendisleri Kuruluşu uzmanları tarafından UAS Challenge 2017 & 2018 yarışmaları için kurgulanmıştır ve kıyı bölgelerinde oluşabilecek doğal afet senaryosunun bir parçasıdır.

Bu tezin asıl amacı, yeni tasarlanmış bir küçük boyutlu İHA'nın uçuş dinamiklerini değerlendirmek üzere MATLAB/Simulink ortamında yüksek detaya sahip bir simülasyon ortamı geliştirmektir. Bu değerlendirmeyi yaparken uçuş kararlılığı ve kontrolü ve görev performansı göz önünde bulundurulacaktır ve bu değerlendirmeye göre aerodinamik boyutlandırmayı, konfigürasyon seçimi ve aynı zamanda ekipman seçimini gibi evrelerin optimize edilmesi amaçlanmaktadır. Geliştirilen matematiksel modeli doğrulamak için İTÜ Hedef Takımı tarafından tasarlanan ve üretilen Tulpar test aracı olarak kullanıldı. İlerleyen süreçlerde bu İHA'nın lisansüstü veya lisans seviyesindeki derslerde öğrencilerin uygulamalı deneyim kazanmaları için kullanılması amaçlanmaktadır.

Simülasyonlar için oluşturulan matematiksel modelin açıklanmasına referans noktalarından başlanarak İHA'nın uçuş dinamikleri, üzerine etkiyen kuvvetleri ve momentleri kısaca açıklanmaktadır. Sonrasında bahsedilen aerodinamik model ise bu uçuş dinamiklerimi kullanarak ve ayrıca XFLR5'ten elde edilen aerodinamik katsayıları ve kararlılık türevlerini kullanarak elde edilmektedir. Aerodinamik model için kullanılabilecek alternatif yaklaşımlar ve analiz programları da kısaca ele alınmaktadır. Ayrıca, hava aracında bulunan itki ve eyleyici elemanlarına ait alt sistemlerin modelleri de verilmiştir. Bu çalışma için belirlenen süre sınırlaması göz önünde bulundurularak deneysel verilerin detayı ve sayısal olarak elde edilebilecek sonuçlar belirlenerek ortak bir şekilde kullanılmıştır. Atölyede yapılan ölçümlere ek

olarak pervane üreticilerinin sağladığı veriler de model içinde kullanıldı. Son olarak, yerçekimi, atmosfer ve rüzgarı denklemlerini de içeren çevre modeli tanıtılmakta ve bu modelin sadece İHA'lar için değil, diğer hava araçları için de kullanılabilmesi için detaylı veri tablolarıyla birlikte hazırlandı.

Tulpar'da da kullanılan uçuş kontrol sistemi MATLAB/Simulink ortamında tanıtıldı ve modellendi. Açık kaynak olarak geliştirilmesinden dolayı otopilot yazılımı ArduPilot olarak seçildi ve Simulink'teki kontrol şeması GitHub'taki kod yapısının okunmasıyla ve önceki benzer çalışmalardan faydalanarak doğrudan elde edildi. Yaygın bir topluluğa sahip olmasından dolayı deneyimli kullanıcılardan ve açık kaynak yapısından oluşan Ardupilot'un literatürdeki en iyi seçeneklerden biri olduğu düşünülmektedir.

Otopilot donanımı, nispeten düşük maliyetli fiyatı nedeniyle ve özellikle Ardupilot için de uygun olması nedeniyle Pixhawk 2.1 olarak seçildi. Pixhawk 2.1, hava hızı, navigasyon ve lazer mesafe ölçümleri için düşük maliyetli sensör seçeneklerine sahip olduğu için bu uygulamada uygun bir seçenek haline getirmektedir. Haberleşme sistemi de seçim metodolojisi ile beraber açıklanmaktadır. Telemetri modülleri 868 MHz'de çalışacak şekilde seçildi ve güç ayarları hava sahasının yönetmeliklere göre kontrol edilerek ayarlandı.

Matematiksel model MATLAB/Simulink ortamında simüle edilmiştir. Simülasyondan önce, denge/düzeltme koşulu, verilen uçuş şartlarına göre küçük pertrbasyon teorisi kullanılarak hesaplanır. Simulink'teki özel görselleştirme şekillerine ek olarak, Flight Gear ekstra bir seçenek olarak sunuldu. Bunun yanı sıra, birçok simülasyon ortamına artıları ve eksileriyle birlikte değinilir. Fakat, çoğunlukla Simulink'in görselleştirme şekli, bilgisayar destekli çizim yazılımındaki uçak çizimlerini istenen versiyonlara dönüştürmek gibi zaman alıcı görevleri ortadan kaldırmak için kullanılmıştır. Simülasyon ortamı, İHA tasarımcısına özel tasarım araçlarının uçuş özellikleri hakkında üretim ve test uçuşları öncesinde bilgi sağlar. Uçuş performans analizi bu çalışmada ele alınmamıştır, ancak görevleri daha iyi yaklaşımlarla da planlayabilme özelliğine sahip olmak için simülasyon ortamına kolayca dahil edilebilir. Ayrıca, geliştirilen bu simülasyon ortamı, bu çalışmanın gelecekteki bir çalışması olan bir USB kumandası eklenerek pilot eğitim aracı olarak da kullanılabilir.

Son bölümde, geliştirilen modelin simülasyon sonuçları, tezin hedeflerinde belirtilen deneme uçuşlarıyla karşılaştırıldı. Test uçuşlarında kullanılan ek donanımlar tanıtıldı ve test uçuş senaryoları açıklandı. Test uçuşları belirlenen senaryolara göre Hezarfen havalimanında gerçekleştirildi. Simülasyon ortamının ideal olması, birçok varsayım ve yaklaşımla modelleme yoluyla elde edilmesi gerçek uçuşta elde edilen sonuçlardan farklıdır. Özellikle aerodinamik model detaylarını göz önünde bulundurarak, XFLR5 gibi analiz programları, girdap kafesi yöntemi (VLM) nedeniyle İHA karakteristiklerini yüksek doğrulukla hesaplamaz ve özellikle de aerodinamik sürükleme modeli hatalar getirir. Ek olarak, kullanılan test platformu, İTÜ Hedef Takımı tarafından düşük maliyetli imkanlarda ve sınırlı takvim süresinde üretildi. Bu nedenle de hesaplanarak tasarlanan araç ile uçurulan araç arasında çeşitli farklılıklar oluşmaktadır.

Bu çalışmada kullanılan kaynak kodunun ve diğer ilgili dosyaların gelecekte açık kaynak olarak yayınlanması amaçlanmaktadır ve bu çalışma da bu amaç için bir temel oluşturacaktır. Bu çalışmada, edinilen deneyimlerin başkaları ile paylaşılması

da hedeflenmiştir. Böylece okuyucu, kontrol listeleri, öğrenilen dersler, ek kaynaklar gibi İHA operasyonları hakkında ek bilgiler bulabilir.

1. INTRODUCTION

1.1 Objectives

The main objective of this thesis is to develop a high fidelity simulation environment in MATLAB/Simulink to evaluate the flight mechanics of a new designed Small UAS considering its flight stability and control and also the mission performance to optimize the equipment selection and also the aerodynamic sizing of the aircraft for various configurations. To validate the developed mathematical model, Tulpar UAS is used as a testbed which is designed by ITU Team Hedef and after the validation, it is aimed to use that UAS in graduate or undergraduate level courses to make students gain hands-on experience on the UASs.

1.2 Unmanned Aircraft Systems

With the growing market of Unmanned Aircraft Systems, there are several different names used such as Unmanned Aircraft System (UAS), Uninhabited Aerial Vehicle (UAV), Remotely Piloted Aircraft System (RPAS), Drone, etc. There is no difference among these terms except their usage among organizations. [12]

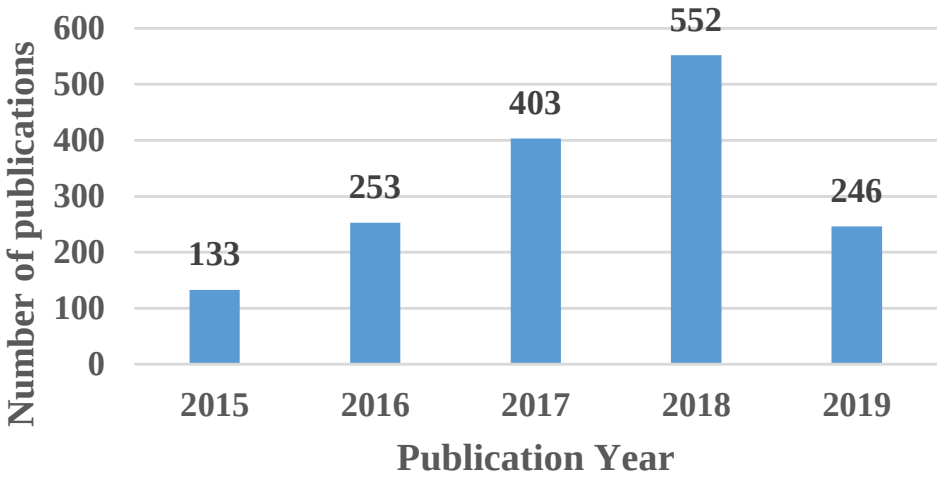


Figure 1.1 : Web of Science results on the number of publication relating to the keywords "unmanned" and "drone". [3].

1.2.1 UAS classification

A most common one used by the authorities is that classifying the UAS based on their take-off mass. Another classification suggested is based on their endurance and maneuverability given in Fig. 1.2. The list of the classes are as follows:

- Rotary-Wing (RW)
- Convertibles (Hybrid)
- Bio-inspired (Flapping-Wing)
- Fixed-Wing (FW)
- Lighter-than-Air (LtA)

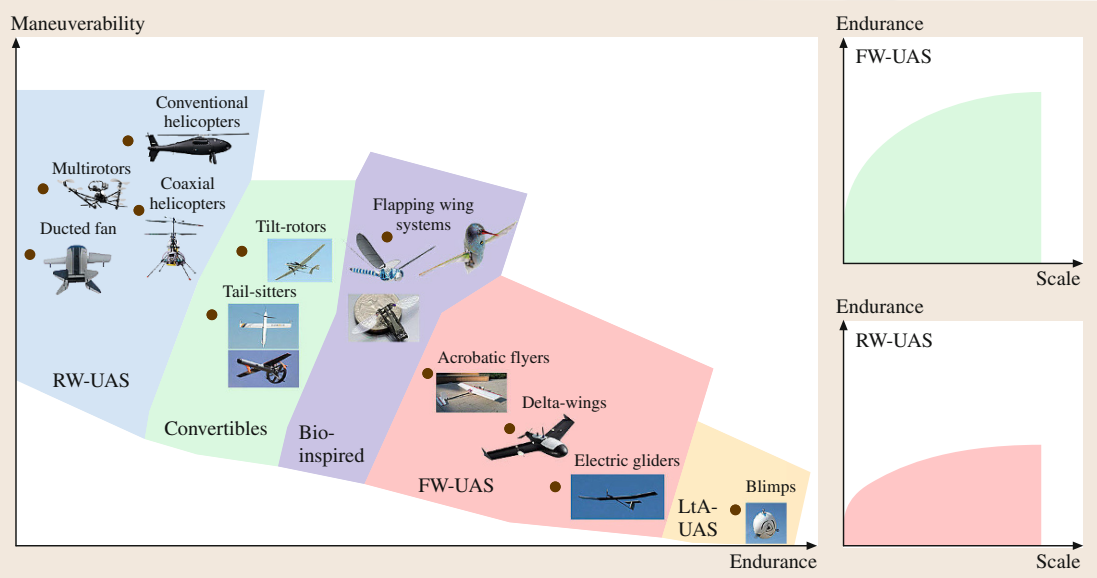


Figure 1.2 : UAS Classification based on their endurance and maneuverability properties [4].

The UAS covered in this work belongs to FW category, and FW-UAS has the benefit of longer endurance than the other options except LtA-UAS. On the other hand, their maneuverability is low compared to RW-UAS which brings a drawback considering the aggressive control in a confined space. Also, they lack the ability to take-off and land from any field, but this can be eliminated via using a launch pad, too.

1.2.2 Turkey developments

There are 4 defense companies from Turkey which were listed among Top 100 for 2018 by DefenceNews, and 3 of these companies, ASELSAN, TAI, and STM, are working UAS [13]. Besides these companies, Vestel Savunma and Baykar Makina are the other big and important ones in the defense industry in Turkey, and the whole UAS Industry information can be found in Fig. 1.3 which is prepared for the year of 2018.

Figure 1.3 : UAS Industry in Turkey [5].

As it can be seen, Turkey has a lot of companies working on UAS, and also this industry is prone to grow considering drones' innovative and useful effects in the society and also in the military applications.

1.3 Test-bed Platform: Tulpar UAS

Tulpar is a Grand Champion UAS developed by Team Hedef. The development process has started in November 2016, and the team participated in two international competitions organized by the Institution of Mechanical Engineers in the United Kingdom. In 2018, the team achieved the Grand Champion Title among other

Figure 1.4 : IMechE's UAS Challenge 2018 - Grand Champion Team Hedef

international participants [14]. In this competition, teams are required to design a UAS that can conduct payload delivery and reconnaissance missions.

The UAS is designed based on the requirements provided by the competition judges, and they are listed below:

- **Maximum Take-off Mass (MTOM):** 6.9 kg,
- **Take-off and Landing Criteria:** Within a 30 m x 30 m box,
- **Wind resistance:** Up to 20 kts, gusting to 25 kts,
- **COTS limitation:** £1,000.

Considering the mission and design requirements, Tulpar was decided to have an inverted V-tail (A-Tail), pusher configuration. The design methodology is given in Fig. 1.5 with other selection reasoning. Firstly, the camera location was the starting point considering the reconnaissance mission, and the nose provides the best viewpoint for that. Then the motor, wing, and tail configuration were chosen considering the lightweight requirement and aerodynamic performance of the UAS. Finally, the payloads are decided to be carried under the fuselage to reduce the volume requirement inside the aircraft.

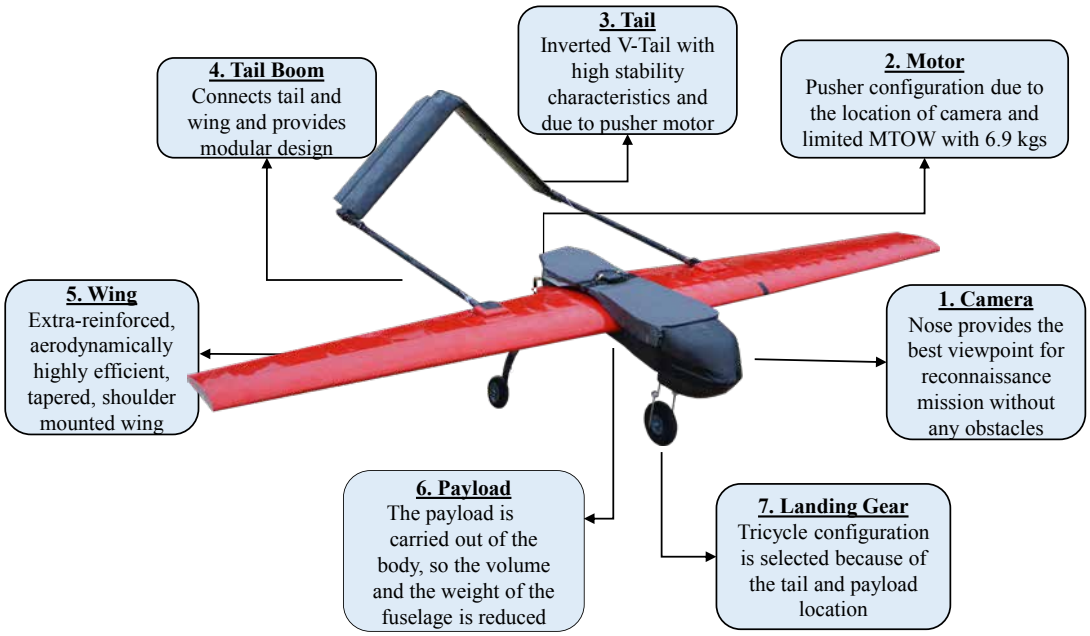


Figure 1.5 : Tulpar's design methodology

In this thesis, a high-fidelity mathematical model for Tulpar was developed and then compared the results with test flight data to validate the model. It is aimed to be used as a test-bed not only in research facilities but also in undergraduate courses given by MoDeL Lab.

1.4 Literature Review

Before starting the work, a wide-spread literature review has been conducted and all the related references were organized using the Mendeley software provided by Elsevier. In the following subsections, they are divided considering their main topics.

1.4.1 Aerodynamic modeling

According to the previous works, the methods besides CFD and wind tunnel testing results in low accuracy for the viscous effect calculation and the drag force estimations [15]. Considering that, to evaluate the performance of the UAS the drag polar is required to be known accurately [16].

Since the dynamics of small UASs results in higher natural frequencies than the bigger ones, it is required to specify different handling and flying qualities for these aircrafts [17].

1.4.2 Propulsion system modeling

About propeller modeling, Brandt's work includes experimental data on several propeller sizes, but the biggest size is 11x7 propeller, which is not useful since Tulpar's propeller was 13x8 [18]. Therefore, APC Propeller's database has been used, which includes the data obtained via NASA's Transonic Airfoil Analysis Computer Program [19].

Since a pusher configuration has been selected, the fuselage's blockage effect on the propulsion efficiency should have been considered. Instead of conducting a time-consuming experiment, a useful reference found to have required information about small propellers and the efficiency drop is calculated accordingly [20].

About the battery model, which was the main energy source for the electric motor, detailed experimental works which has information on high power lithium-polymer types of batteries were found [21]. However, since the mission performance was not the main focus, the battery source is considered to be infinity during the simulations.

1.4.3 Flight test procedures of SUAS

Paw's Ph.D. thesis at the University of Minnesota and Jung's Ph.D. thesis at Georgia Tech provides model validation data in addition to Dr. Beard's SUAS reference book [22–24]

Kabbabe's M.Sc. thesis and Cotting's Ph.D. thesis gives beneficial information about integration UAV platform into education [25, 26].

1.5 Thesis Organization

In this thesis, there are four main chapters as listed below. They are summarized as follows:

- **Chapter 1 - Introduction:** The main objective of this thesis is to develop a high fidelity simulation environment in MATLAB/Simulink to evaluate the flight mechanics of a new designed Small UAS considering its flight stability and control and also the mission performance to optimize the equipment selection and also the aerodynamic sizing of the aircraft for various configurations. To validate the

developed mathematical model, Tulpar UAS is used as a testbed which is designed by ITU Team Hedef and after the validation, it is aimed to use that UAS in graduate or undergraduate level courses to make students gain hands-on experience on the UASs.

- **Chapter 2 - High-Fidelity Modeling of Unmanned Aircraft Systems:** In this chapter, the mathematical modeling foundation will be introduced. Starting from the reference frames, forces and moments of UAS will be explained briefly. Then the aerodynamic model will be obtained using these forces and moments. To obtain the aerodynamic model, several coefficients and stability derivatives are required to be calculated from several tools, as mentioned in this chapter. Then the subsystems' models are given for propulsion systems and actuators. Finally, the environment model including gravity, atmosphere, and wind is introduced.

- **Chapter 3 - Flight Control System:** In this section, the flight control system modeled and also used in Tulpar is introduced. Autopilot software is chosen to be ArduPilot due to its open-source development, and the control scheme in Simulink is directly obtained by reading the code structure. Hardware is chosen to be Pixhawk 2.1 mainly due to its relatively low-cost, and it is suitable for Ardupilot as well. In the final subsection, the sensors and additional equipment are mentioned with the selection methodology followed by the ITU Hedef UAS Team.

- **Chapter 4 - Software-in-the-loop Simulation Environment:** In this section, the mathematical model will be simulated in MATLAB/Simulink environment. Before the simulation, the trim condition is calculated accordingly. Additional environments are mentioned as well, and in addition to custom visualization figure in Simulink, Flight Gear is connected as an extra option.

- **Chapter 5 - Test Flight Validation:** The developed model is validated with test flights as mentioned in the objectives of the thesis. In this section, additional equipment is introduced and the test flight scenarios are explained. The test flights are conducted at Hezarfen airport according to these scenarios, and the results are compared with the simulation. Simulation is the ideal environment, and since it is obtained via modeling with a lot of assumptions and approximations, there exists a difference with real-life results. Also, the platform is manufactured by ITU Hedef

UAS Team with low-cost facilities and limited calendar time, so there exist minor differences due to that condition, too. Just a reminder to the reader, this work is conducted to validate the aerodynamic model of the Tulpar, and the performance calculations are not in the scope of it. It could be included in future works.

- **Chapter 6 - Conclusion:**

In addition to these chapters, several appendices are introduced in this work. There is information about UAS parameters, autopilot parameters, MATLAB codes used inside Simulink blocks among these appendices. Other important information is included about test flights, and the lessons learned by ITU Team Hedef is summarized here so that the future works can get benefit from them.

If you reported an issue/error in the thesis to the author and would like to check the updated document, please check the following link: bit.ly/ondes-thesis-msc-pdf

Also, if the research group decides to share the files used in this thesis, the reader can find them in the following link: bit.ly/ondes-thesis-msc-files

2. High Fidelity Modeling of Unmanned Aircraft Systems

In this chapter, the mathematical modeling foundation will be introduced. Starting from the reference frames, forces and moments of UAS will be explained briefly. Then the aerodynamic model will be obtained using these forces and moments. To obtain the aerodynamic model, several coefficients and stability derivatives are required to be calculated from several tools, as mentioned in this chapter. Then the subsystems' models are given for propulsion systems and actuators. Finally, the environment model including gravity, atmosphere, and wind is introduced.

2.1 Modeling of Unmanned Aircraft Systems

2.2 Reference Axes Selection

Since we will be dealing with flight dynamics, the reference axes (or reference frames) must be defined properly and clearly. The coordinate frames (or axis systems) used for UAS are listed as follows:

1. Inertial Frame
2. Body Frame
3. Wind Frame

The inertial frame is also named as Earth frame, or as North-East-Down (NED) frame. They are required to defined separately because of the following reasons:

- Inertial frame is the base frame and all Newton's equations are derived relative to this frame. All of the mission plans are defined based on this frame. Also, the GPS sensor provides the measurements (position, ground speed, course) in this frame.
- Body frame is used for obtaining the attitude of the aircraft from the onboard measuring equipment such as accelerometer and gyros.

- Wind frame is used for aerodynamic calculations to be used in the equations of motion because they are referenced to the free stream velocity vector [27].

2.2.1 NED to body frame transformation

The inertial frame is referred to be North-East-Down (NED) reference frame as shown in 2.1. The origin of this frame is defined to be the home location and it is an earth-fixed coordinate system. The vehicle frame is the frame that considers the center of mass of UAS to be its origin, and the vectors of the frame are defined to be the same as the inertial frame. The transformation from vehicle frame to body frame consists of the three rotation using Euler angles in the sequence of Yaw-Pitch-Roll ($\psi - \theta - \phi$).

From Beard's book the transformation is summarized as follows [24]:

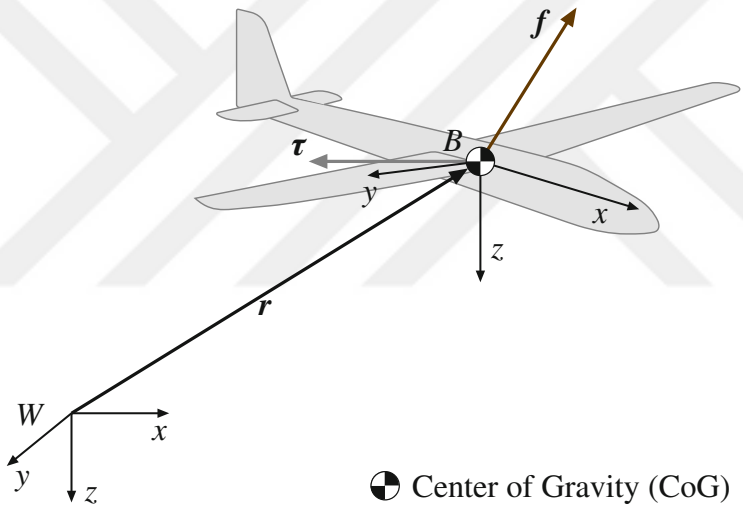


Figure 2.1 : North-East-Down (NED) reference frame [4].

$$R^{body}_{vehicle} = R^{b}_{v}(\phi,\theta,\psi) = R^{b}_{v2}(\phi)R^{v2}_{v1}(\theta)R^{v1}_{v}(\psi) \tag{2.1}$$

where,

$$R^{b}_{v2}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, R^{v2}_{v1}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, R^{v1}_{v}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The derivations of these matrices are not given in this study, but they can be found in the reference book with visual illustrations.

By multiplying the matrices, the transformation matrix can be obtained as follows:

$$R^{b}_{v}(\phi,\theta,\psi) = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ (-c\phi s\psi + s\phi s\theta c\psi) & (c\phi c\psi + s\phi s\theta s\psi) & s\phi c\theta \\ (s\phi s\psi + c\phi s\theta c\psi) & (-s\phi c\psi + c\phi s\theta s\psi) & c\phi c\theta \end{bmatrix} \tag{2.2}$$

where $c_\theta \triangleq cos\theta$ and $s_\psi \triangleq sin\psi$.

This transformation is named as "NED to Forward Transformation" and shown as C_{ned}^{frd} in Steven's reference book where "frd" means forward [28]. Both C_{ned}^{frd} and R_v^b notations have the same definition.

2.2.2 Quaternions

Even though the Euler angler provides the reader with a better insight and understanding, there exists a mathematical singularity when the pitch angle (θ) is equal to $\pm 90^\circ$. In that condition, which results in computational stability, the yaw angle ψ is not defined due to being coincident with pitch axis. In the literature this situation is referred as "Gimbal Lock", and a useful demonstration can be found YouTube https://www.youtube.com/watch?v=0VAc_G79POE.

To avoid this singularity issue and also have a more computationally efficient representation, quaternions can be used. The quaternion e can be represented in $\mathbb{R}^4$ as follows:

$$e = \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

where e_i, for $i = 1,2,3,4$ are scalars. A quaternion must be a unit quaternion to be used to represent a rotation, therefore:

$$\|e\| = 1$$

The conversion from yaw, pitch, roll (ψ, θ, ϕ) Euler angles to quaternions is given as follows:

$$\begin{aligned}
e_0 &= \cos\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} \\
e_1 &= \cos\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} - \sin\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} \\
e_2 &= \cos\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} \\
e_3 &= \sin\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} - \cos\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2}
\end{aligned}$$

And also, using the following equations, the quaternion representation can give information about Euler angles as follows:

$$\phi = \operatorname{atan2}\left(2\left(e_0 e_1 + e_2 e_3\right), \left(e_0^2 + e_3^2 - e_1^2 - e_2^2\right)\right)$$
$$\theta = \operatorname{asin}\left(2\left(e_0 e_2 - e_1 e_3\right)\right)$$
$$\psi = \operatorname{atan2}\left(2\left(e_0 e_3 + e_1 e_2\right), \left(e_0^2 + e_1^2 - e_2^2 - e_3^2\right)\right)$$

2.3 Forces and Moments

In this section, the forces and moments are given from the references, which are the core equations of the flight dynamics. All of their derivations can be found in the following references [29–31]

2.3.1 Euler

From the reference book selected, the Flat-Earth, Body-Axes 6-DoF 12 states' equations are given as follows [24].

Transitional velocity states:

$$\begin{pmatrix} \dot{u}_r \\ \dot{v}_r \\ \dot{w}_r \end{pmatrix} = \begin{pmatrix} r v_r - q w_r \\ p w_r - r u_r \\ q u_r - p v_r \end{pmatrix} + \frac{1}{\mathrm{m}} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} - R_v^b(\phi, \theta, \psi) \begin{pmatrix} \dot{w}_n \\ \dot{w}_e \\ \dot{w}_d \end{pmatrix} \tag{2.3}$$

Angular rate states:

$$\begin{pmatrix} p \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 \left(p^2 - r^2\right) \\ \Gamma_7 pq - \Gamma_1 qr \end{pmatrix} + \begin{pmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{I_m} m \\ \Gamma_4 l + \Gamma_8 n \end{pmatrix} \tag{2.4}$$

Angular position states:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2.5}$$

Position in NED states:

$$\begin{pmatrix} p_n \\ p_e \\ p_d \end{pmatrix} = R_b^v(\phi, \theta, \psi) \begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} + \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix} \tag{2.6}$$

where

$$\Gamma_1 = \frac{J_{xz}\left(J_x - J_y + J_z\right)}{\Gamma} \tag{2.7}$$

$$\Gamma_2 = \frac{J_z\left(J_z - J_y\right) + J_{xz}^2}{\Gamma} \tag{2.8}$$

$$\Gamma_3 = \frac{J_Z}{\Gamma} \tag{2.9}$$

$$\Gamma_4 = \frac{J_{xz}}{\Gamma} \tag{2.10}$$

$$\Gamma_5 = \frac{J_z - J_x}{J_y} \tag{2.11}$$

$$\Gamma_6 = \frac{J_{xz}}{J_y} \tag{2.12}$$

$$\Gamma_7 = \frac{\left(J_x - J_y\right)J_x + J_{xz}^2}{\Gamma} \tag{2.13}$$

$$\Gamma_8 = \frac{J_x}{\Gamma} \tag{2.14}$$

2.3.2 Quaternion

The quaternion representation is summarized in the appendices of the Beard's Book [24], and they are given as follows.

Position in NED states:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \begin{pmatrix} e_1^2 + e_0^2 - e_2^2 - e_3^2 & 2\left(e_1e_2 - e_3e_0\right) & 2\left(e_1e_3 + e_2e_0\right) \\ 2\left(e_1e_2 + e_3e_0\right) & e_2^2 + e_0^2 - e_1^2 - e_3^2 & 2\left(e_2e_3 - e_1e_0\right) \\ 2\left(e_1e_3 - e_2e_0\right) & 2\left(e_2e_3 + e_1e_0\right) & e_3^2 + e_0^2 - e_1^2 - e_2^2 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{2.15}$$

Transitional velocity states:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{2.16}$$

$$\begin{pmatrix} e_0 \\ \dot{e}_1 \\ \dot{e}_2 \\ e_3 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} \tag{2.17}$$

Angular rate states:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6\left(p^2 - r^2\right) \\ \Gamma_7 pq - \Gamma_1 qr \end{pmatrix} + \begin{pmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{I_y} m \\ \Gamma_4 l + \Gamma_8 n \end{pmatrix} \tag{2.18}$$

$$\begin{pmatrix} \dot{e}_0 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & -p & -q-r & \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} - \lambda \frac{\partial J}{\partial e} \tag{2.19}$$

$$\mathbf{f}_g^b = \mathbf{m}g \begin{pmatrix} 2(e_1e_3 - e_2e_0) \\ 2(e_2e_3 + e_1e_0) \\ e_3^2 + e_0^2 - e_1^2 - e_2^2 \end{pmatrix} \tag{2.20}$$

2.4 Aerodynamic Model

To obtain the aerodynamic mathematical model of the UAS, the parameters required are listed in Table 2.1. Since wind tunnel testing or using computational fluid dynamics (CFD) techniques would be extremely time-consuming, the tools using Vortex Lattice Methods (VLM) such as XFLR5 is used during this work to obtain these parameters. The other option could be the one uses empirical approaches such as DATCOM. The list of parameters is given in the appendix.

Table 2.1 : List of parameters.

Physical	Aerodynamics	Propulsion
m	$C_{L_0}, C_{L_\alpha}, C_{L_q}, C_{L_{\delta_e}}, C_{L_{\delta_f}}$	k_{motor}
J	$C_{D_0}, C_{D_\alpha}, C_{D_q}, C_{D_{\delta_e}}, C_{D_{\delta_f}}$	S_{prop}
b	$C_{m_0}, C_{m_\alpha}, C_{m_q}, C_{m_{\delta_e}}$	q_{motor}
c	$C_{Y_\beta}, C_{Y_p}, C_{Y_r}, C_{Y_{\delta_a}}, C_{Y_{\delta_r}}$	C_{prop}
S	$C_{l_\beta}, C_{l_p}, C_{l_r}, C_{l_\delta}, C_{l_{\delta_r}}$	
	$C_{n_\beta}, C_{n_p}, C_{n_r}, C_{n_{\delta_a}}, C_{n_{\delta_r}}$	

2.4.1 Longitudinal

Forces:

$$F_{\text{lift}} = \frac{1}{2}\rho V_a^2 S C_L\left(\alpha, q, \delta_e, \delta_f\right) \tag{2.21}$$

$$F_{\text{drag}} = \frac{1}{2}\rho V_a^2 S C_D\left(\alpha, q, \delta_e, \delta_f\right) \tag{2.22}$$

$$M = \frac{1}{2}\rho V_a^2 S c C_m\left(\alpha, q, \delta_e, \delta_f\right) \tag{2.23}$$

Since lift and drag forces are defined in stability frame, the following rotation is required to find the force in the body frame:

$$\begin{bmatrix} f_x \\ 0 \\ f_z \end{bmatrix} = R_w^b(\alpha, \beta) \begin{bmatrix} -F_{\text{drag}} \\ 0 \\ -F_{\text{lift}} \end{bmatrix} \tag{2.24}$$

Coefficients:

Lift force coefficient:

$$C_L(\alpha, q, \delta_e) = \left[C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e + C_{L_{\delta_f}}\delta_f \right] \tag{2.25}$$

Drag force coefficient:

$$C_D(\alpha, q, \delta_e) = \left[C_{D_0} + C_{D_\alpha}\alpha + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e + C_{D_{\delta_f}}\delta_f \right] \tag{2.26}$$

Pitch moment coefficient:

$$C_m(\alpha, q, \delta_e) = \left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e + C_{m_{\delta_f}}\delta_f \right] \tag{2.27}$$

2.4.2 Lateral

Forces:

$$f_y = \frac{1}{2}\rho V_a^2 S C_Y(\beta, p, r, \delta_a, \delta_r) \tag{2.28}$$

$$L = \frac{1}{2}\rho V_a^2 S b C_l(\beta, p, r, \delta_a, \delta_r) \tag{2.29}$$

$$N = \frac{1}{2}\rho V_a^2 S b C_n(\beta, p, r, \delta_a, \delta_r) \tag{2.30}$$

Coefficients:

Side force coefficient:

$$C_Y(\beta, p, r, \delta_a, \delta_r) = \left[C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \right] \tag{2.31}$$

Roll moment coefficient:

$$C_l(\beta, p, r, \delta_a, \delta_r) = \left[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta a}}\delta_a + C_{l_\delta r}\delta_r \right] \tag{2.32}$$

Yaw moment coefficient:

$$C_n(\beta, p, r, \delta_a, \delta_r) = \left[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta a}}\delta_a + C_{n_{\delta_r}}\delta_r \right] \tag{2.33}$$

2.4.3 Other tools available to estimate stability derivatives

Comparison of the tools can be found in the reference, and it provides information about several tools, including aerodynamic analysis tools [32]. One of them also stated in that reference is called CEASIM, and also there is a master's thesis uses CEASIM also, which could be beneficial for the reader [33].

2.5 Inertia Model

All of the subsystems are added into CATIA and then their masses are identified to calculate the inertia of Tulpar. Another alternative is to use a pendulum test to identify. The details are given in the following subsections, and the inertia parameters are given in Table 2.2, where the location of the center of gravity (CoG) is taken in the x-axis, longitudinal axis, the distance from the leading edge of the lifting surface, wing.

Table 2.2 : Inertia parameters of Tulpar UAS.

Parameter	Value
J_x	0.275 kg.m^2
J_y	0.474 kg.m^2
J_z	0.716 kg.m^2
J_xz	-0.049 kg.m^2
Location of CoG	14.7 cm

Besides these methods, a MATLAB code and visualization were developed by researchers from ETH Zürich during Sky-Sailor project [6] and its plot is shown in Fig. 2.2. A similar tool can be developed or adapted in the future for this project, too.

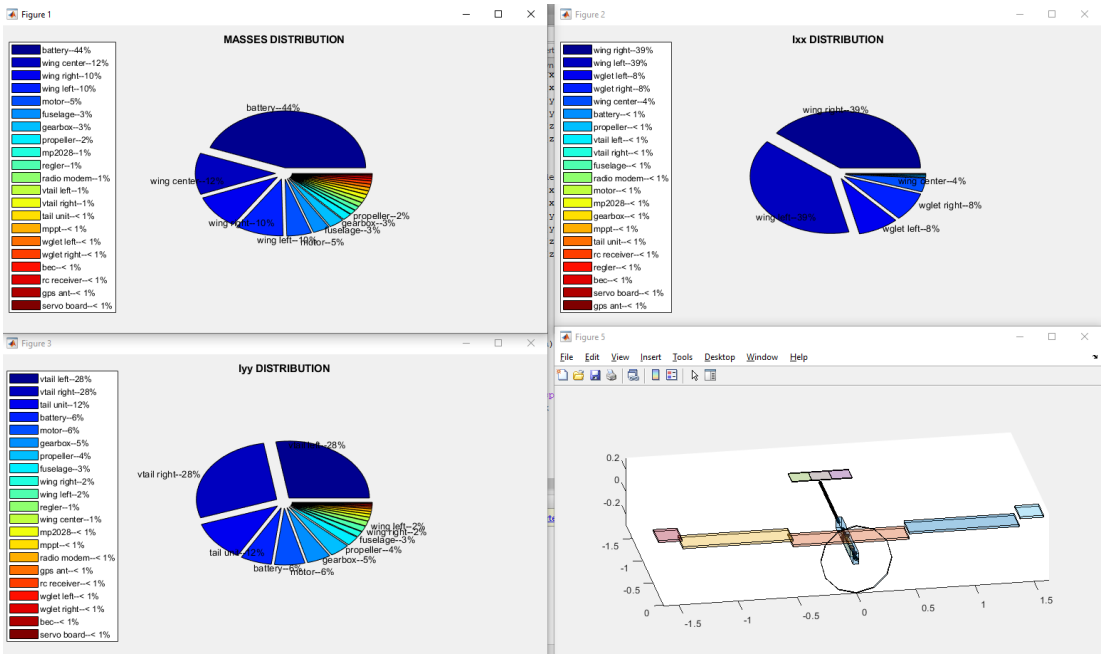


Figure 2.2 : Inertia calculation in MATLAB [6].

2.5.1 CAD software

Tulpar was drawn in CATIA v5 r19 CAD software before the manufacturing to conduct detailed analysis, and also to plan the manufacturing process in an efficient way by the members of Team Hedef as shown in Fig. 2.3. In CATIA, once the mass of each part is identified, the center of gravity and also the inertia matrix is calculated directly as it can be seen from the properties of the assembly product [34].

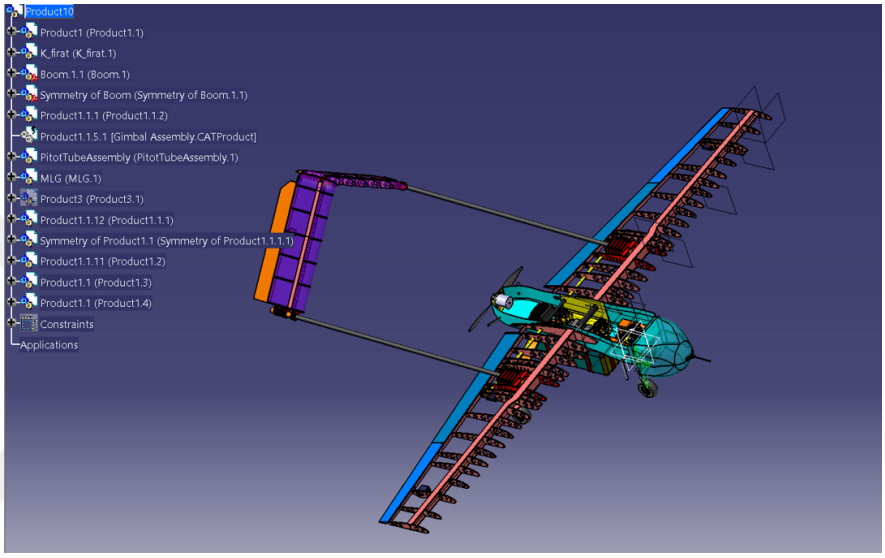


Figure 2.3 : CATIA Assembly of Tulpar.

2.5.2 XFLR5 inertia calculator

Besides the aerodynamic coefficients, XFLR5 has the beneficial usage for the inertia calculation as well. The user enters the mass and also the exact location from the leading edge as given in Fig. 2.4, and the XFLR5 can directly calculate the aircraft's inertia values. After that calculation, the inertia values given in Table 2.2.

2.5.3 Pendulum test

Besides these numerical methods, the pendulum test can be conducted to figure out the inertia of the aircraft as well. According to the torsional pendulum experimental setup from the reference book, the following equations is used [35]:

$$J_\star = \frac{gT_m^2 d^2 m}{4\pi^2 l}, \quad \star = x, y, z \tag{2.34}$$

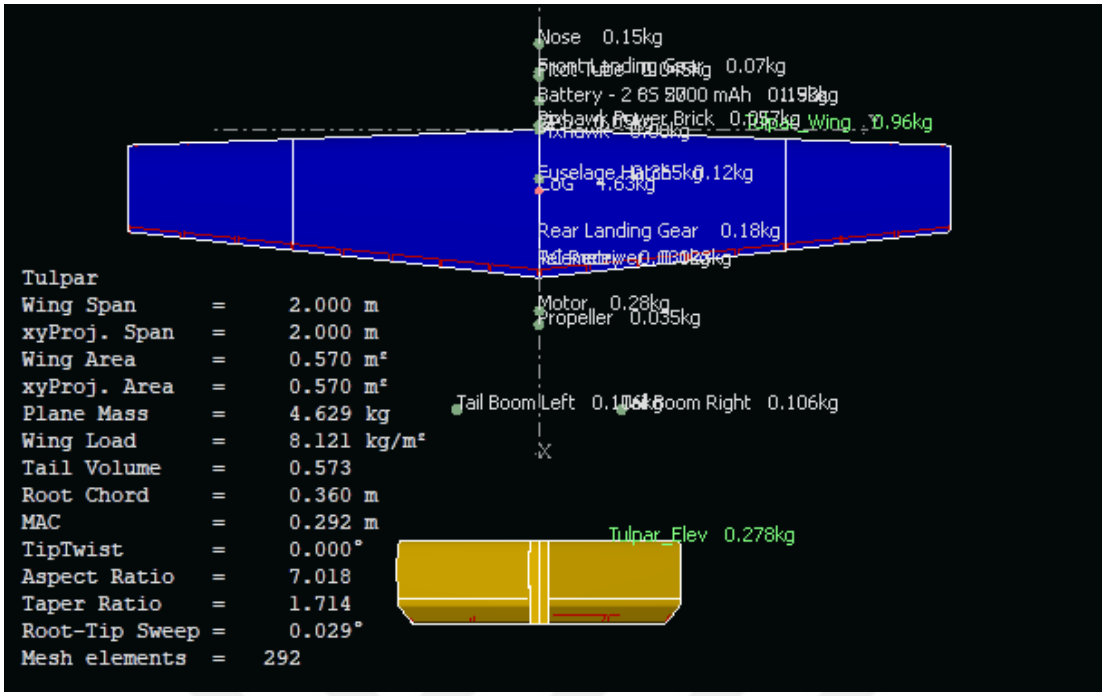


Figure 2.4 : XFLR5 view of Tulpar with masses.

where,

- d is the distance from the center of gravity to the point at which the suspension wire is vertically connected.
- g is the gravity.
- l is the length of the wire from the ceiling
- m is the mass of the vehicle.
- T_m is the period of a small perturbed rotational motion.

The example setup for a pendulum test can be found in the following references [23]. This method is not preferred due to the time limitation for this work.

2.6 Propulsion System Modeling

Based on the experience we had during projects in Team Hedef, the propulsion system selection is crucial considering the performance and the weight limitations in UAVs. As stated in Lundström's work, the original motor parameters provided by the seller or the manufacturer are seen to be inaccurate [36]. Therefore, the experimental

identification of the brushless DC motor is important and is required to be done. To eliminate this identification, dynamic thrust calculation will be used as provided below.

Before that, the static thrust test results which include throttle vs current and throttle vs. static thrust measurement are shown below. This test is conducted to test the motor in the ground and it is expected to have at least 50% of the take-off weight. Also, the engine speed (rpm) is measured too. Static thrust results for the SunnySky X3525 720kV is shown in Fig. 2.5 for 13x8 wooden propeller. Also, the basic setup inside the workshop can be seen in the same figure, where the motor is connected to the thrust test stand, which directly delivers the force to the scale to measure the static thrust value.

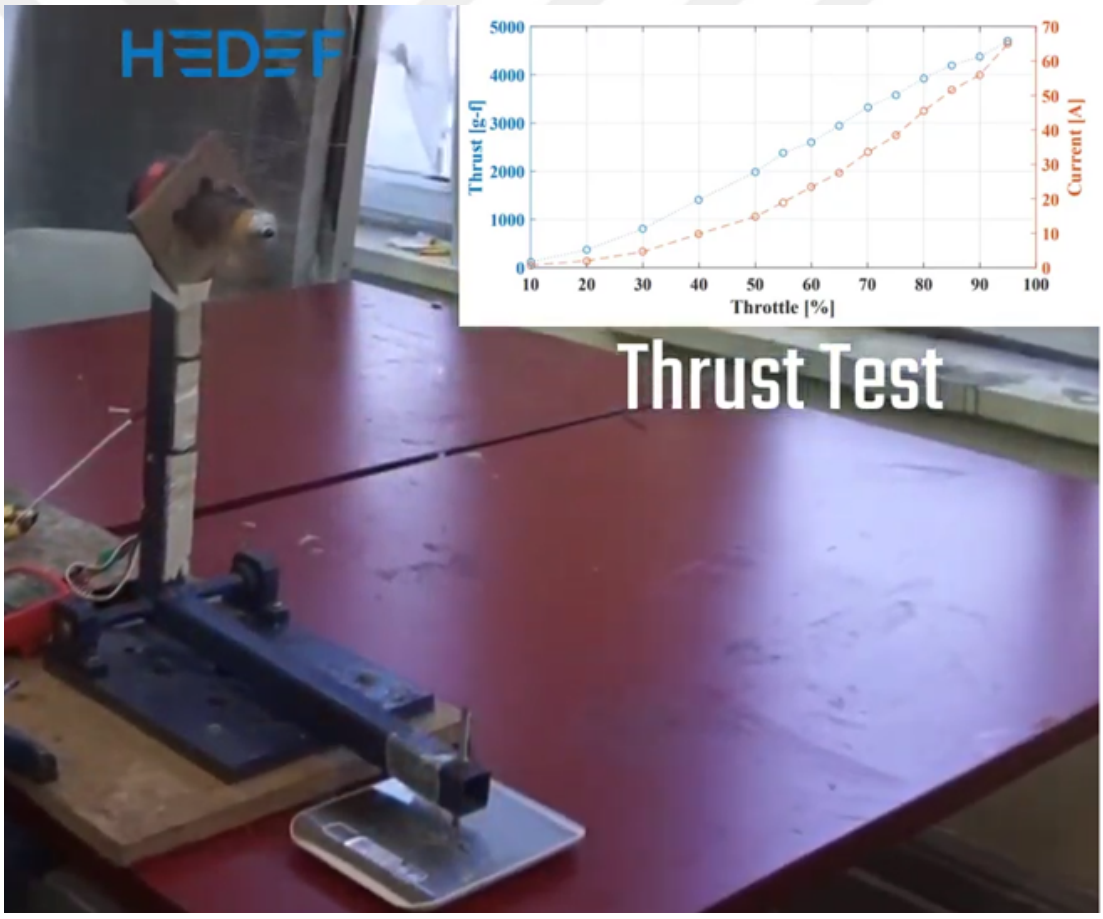


Figure 2.5 : Static thrust test conducted with Team Hedef.

During the flight, the dynamic thrust is required to be known, which is a complicated phenomenon. Therefore, using the APC Propeller database the following advanced ratio J is calculated, which is required to incorporate the effect of forward flight speed

as follows [19]:

$$J = \frac{\pi V_T}{\Omega R_p} \tag{2.35}$$

where, R_p is the propeller radius in meters and Ω the rotational speed in [rad/sec].

Then, the thrust force is calculated as follows:

$$T_{\text{th}} = \frac{4}{\pi^2}\rho\Omega^2 R_p^4 c_T \tag{2.36}$$

Since the motor is placed on the back, the propeller area is reduced due to the fuselage blockage, and according to the work, the 80% reduction in the propeller performance is expected [37]. The motor selection was done with margins, so this does not bring any issue about the flight performance, but it reduces the efficiency which is a known drawback for pusher configuration compared to tractor one.

2.7 Actuator Modeling

As mentioned in the previous section, brushless DC motor is not modeled directly, but the thrust values are used that are obtained experimentally and also from the propeller manufacturer's database. In this section, the actuators used for the control surfaces will be modeled, and the Simulink diagram is shown in Fig. 2.6.

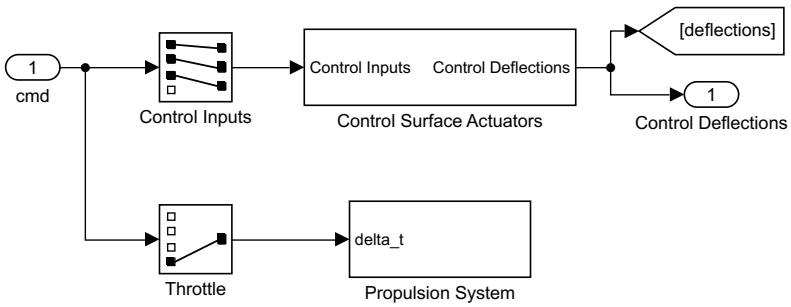


Figure 2.6 : Simukink layout of the actuators subsystem block.

2.7.1 Servo test

Servo motors system identification and delay test are done to obtain the transfer function of the actuators. By comparing the lag between the input and the output, and also identifying the first-order response to step inputs the actuator system models are improved.

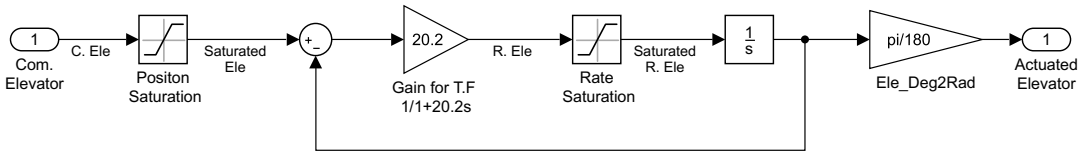


Figure 2.7 : Simukink layout of the elevator actuator model.

Also, the angle limitations are directly obtained from Tulpar as shown in Fig since it is manufactured by Team Hedef but not a factory product. After these measurements, the Table 2.3 is obtained.

Table 2.3 : Angle limitation measurements for Tulpar UAS.

Table 2.4 : Actuator model limitations for Tulpar UAS

Control Surface	Deflection Angle Range [deg]
Flap	15 & 30
Aileron	± 35
Ruddervator	± 30

2.8 Environment Models

There are three different environment models used during the simulation:

- Gravity Model (WGS-84)
- Atmosphere Model (U.S. Standard Atmosphere)
- Wind Model (Dryden)

The related Simulink subsystem is shown in Fig. 2.8, and from this subsystem the required environmental states are calculated.

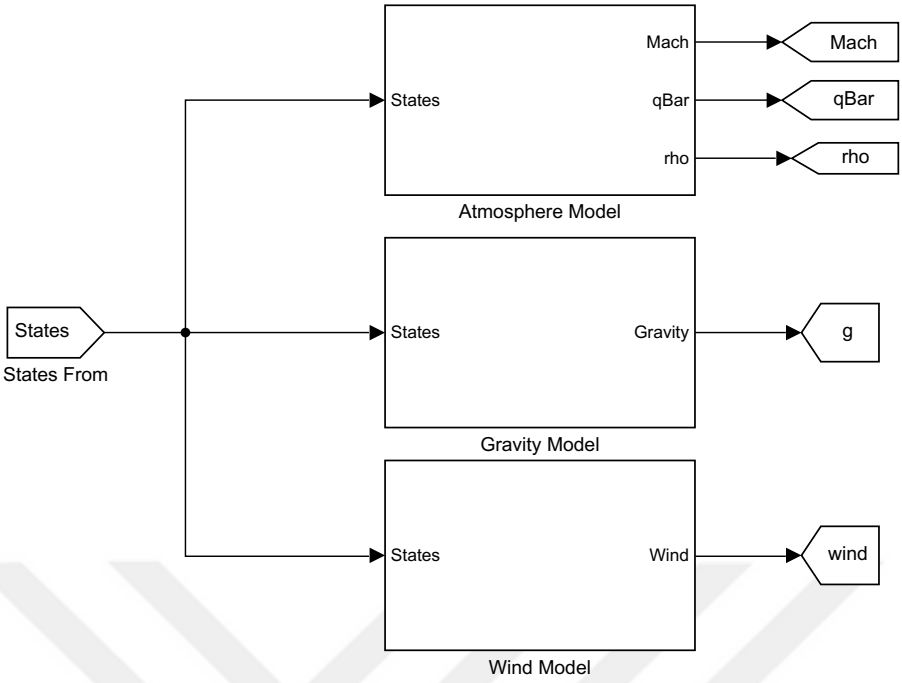


Figure 2.8 : Simukink layout of the environment subsystem.

2.8.1 Gravity model

Since the Global Positioning System (GPS) relies on World Geodetic System 1984 (WGS-84) for the ephemerides of its satellites, this model developed by the Defense Mapping Agency is chosen instead of Earth Gravitational Model 96 (EGM-96) that has 130,676 coefficients which result in higher computational time compared to WGS-84 [1]. The parameters used in the model is given below in Table 2.5.

Table 2.5 : WGS-84 fundamentals parameters [1].

Parameter	Value
Ellipsoid - Semi-major axis (a)	$6,378,137.0$ m
Ellipsoid - Reciprocal flattening ($1/f$)	298.257223563
Earth's angular velocity (ω_E)	$7,292,115.0 \times 10^{-11}$ rad/sec
Earth's gravitational constant (GM)	$3,986,004.418 \times 10^{8}$ m^3/s^2
Speed of light in a vacuum (c)	2.99792458×10^{8} m/s

2.8.2 Atmosphere model

U.S. Standard Atmosphere model is used and it gives the information based on the altitude [38]. It includes 4 different look-up tables to interpolate for an altitude range

of 0 to 86,000 meters which provide air data. The block taken from AeroSim [7] is shown in Fig. and the outputs of this block:

- p = static pressure, in [Pa].
- T = Outside-Air Temperature, in [K].
- rho = air density, in [kg/m^3].
- a = speed of sound, in [m/s].

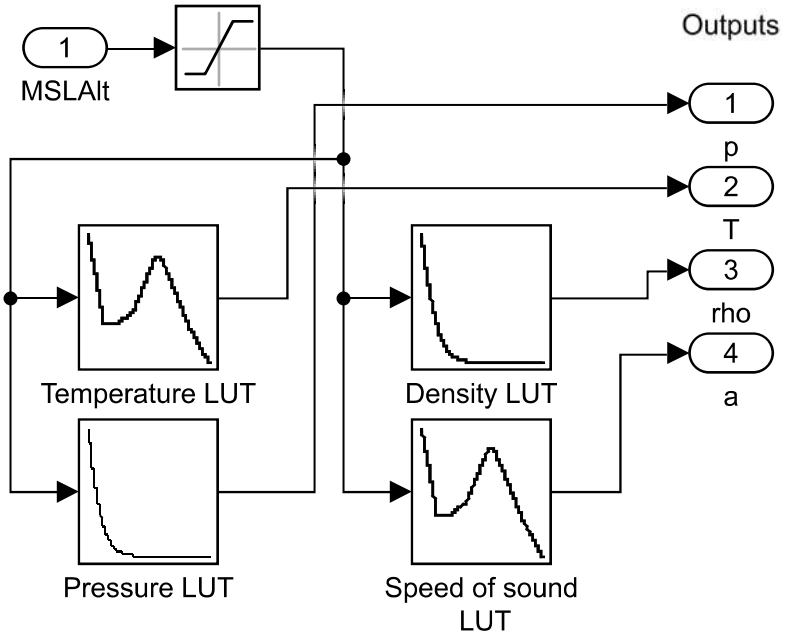


Figure 2.9 : Simukink layout of the atmosphere model [7].

It also includes the speed of sound calculation to have a more generic model. Since this work is about Small UAS, the speed of sound will not be discussed. However, the model can easily be adapted to other aircraft with different flight conditions/regimes.

2.8.3 Wind model

Wind has significant effect on the flight dynamics, and considering the size and speed limitations, Tulpar's flight conditions were limited to winds up to 20 kts, gusting to 25 kts as mentioned in first chapter. In the simulation, wind is defined to have two parts as follows:

$$\mathbf{V}_w = \mathbf{V}_{w_s} + \mathbf{V}_{w_g} \tag{2.37}$$

where $\mathbf{V}_{w_s}$ is the ambient wind, and $\mathbf{V}_{w_g}$ is dynamic term that indicates the wind gusts and other atmospheric disturbances [24].

The ambient wind $\mathbf{V}_{w_s}$ is defined in inertial frame as follows:

$$\mathbf{V}_{w_s}^i = \begin{pmatrix} w_{n_s} \\ w_{e_s} \\ w_{d_s} \end{pmatrix} \tag{2.38}$$

And the gust $\mathbf{V}_{w_g}$ is defined in body frame as follows:

$$\mathbf{V}_{w_g}^b = \begin{pmatrix} u_{w_g} \\ v_{w_g} \\ w_{w_g} \end{pmatrix} \tag{2.39}$$

Dryden model transfer functions:

$$H_v(s) = \sigma_v \sqrt{\frac{3V_a}{L_v}} \frac{\left(s + \frac{V_a}{\sqrt{3}L_v}\right)}{\left(s + \frac{V_a}{L_v}\right)^2} \tag{2.40}$$

$$H_w(s) = \sigma_w \sqrt{\frac{3V_a}{L_w}} \frac{\left(s + \frac{V_a}{\sqrt{3}L_w}\right)}{\left(s + \frac{V_a}{L_w}\right)^2} \tag{2.41}$$

$$H_u(s) = \sigma_u \sqrt{\frac{2V_a}{L_u}} \frac{1}{s + \frac{V_a}{L_u}} \tag{2.42}$$

And the values of these transfer functions are taken from Langelaan's work [2] and shown in Table 2.6. After obtaining the wind separately, the combination is obtained

Table 2.6 : Gust field properties [2].

Description	Altitude (m)	L_u (m)	L_w (m)	σ_u (m/s)	σ_w (m/s)
Low altitude, light turbulence	50	200	50	1.06	0.7
Low altitude, moderate turbulence	50	200	50	2.12	1.4
Medium altitude, light turbulence	600	533	533	1.5	1.5
Medium altitude, moderate turbulence	600	533	533	3.0	3.0

as follows in body frame:

$$\mathrm{V}_w^b = \begin{pmatrix} u_w \\ v_w \\ w_w \end{pmatrix} = \mathscr{R}_v^b(\phi, \theta, \psi) \begin{pmatrix} w_{n_s} \\ w_{e_s} \\ w_{d_s} \end{pmatrix} + \begin{pmatrix} u_{w_s} \\ v_{w_g} \\ w_{w_s} \end{pmatrix} \tag{2.43}$$

Then the air speed in body frame is obtained as follows:

$$\mathbf{V}_a^b = \begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = \begin{pmatrix} u - u_w \\ v - v_w \\ w - w_w \end{pmatrix} \tag{2.44}$$

Using the airspeed affected by the wind, the angle of attack α and side slip angle β is obtained as follows:

$$\begin{aligned} V_a &= \sqrt{u_r^2 + v_r^2 + w_r^2} \\ \alpha &= \tan^{-1}\left(\frac{w_r}{u_r}\right) \\ \beta &= \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2+v_r^2+w_r^2}}\right) \end{aligned} \tag{2.45}$$

3. Flight Control System

The flight control system, also used in Tulpar, is introduced and modeled in MATLAB/Simulink environment. Autopilot software is chosen to be ArduPilot due to its open-source development, and the control scheme in Simulink is directly obtained by reading the code structure on GitHub and also from the previous similar works [41]. Hardware is chosen to be Pixhawk 2.1 mainly due to its relatively low-cost, and it is suitable for Ardupilot as well. In the final subsection, the sensors and additional equipment are mentioned with the selection methodology followed by the ITU Hedef UAS Team.

3.1 AutoPilot Software Structure

Due to its flexible environment, open-source distribution, and great developer community ArduPilot was chosen as the autopilot software for this application. Due to these reasons, its architecture is still under development, and also new modules/functions are added on top of each other which makes it difficult to follow at first sight. On the other hand, in addition to the official documentation, by following the commits on GitHub it can be understood in detail without any problem. Some of the contributions are based on the experiences because the contributions are testing the new versions with their own UAS as well.

Important features of ArduPilot is listed below:

- C++ open source-based code.
- High-quality auto-level and auto-altitude control.
- Others both enhanced remote control flight (via a number of intelligent flight modes) and execution of autonomous missions.
- Worldwide spread and tested code with large community support.
- Multiple sensor options.

- Allows different communication channels (i. e. MAVLink protocol).
- Endless options for customization and expanded mission capabilities.

Another benefit of being open-source for ArduPilot is they are participating in the Google Summer of Code to bring additional features integrated to the platform by the undergraduate/graduate student from all over the world [39].

In Fig. 3.1, the architecture of the Ardupilot software is shown [8]. It is seen that it consists of different stages and the software decides about the flow based on its current flight mode set via RC pilot or from the ground control station. The stages

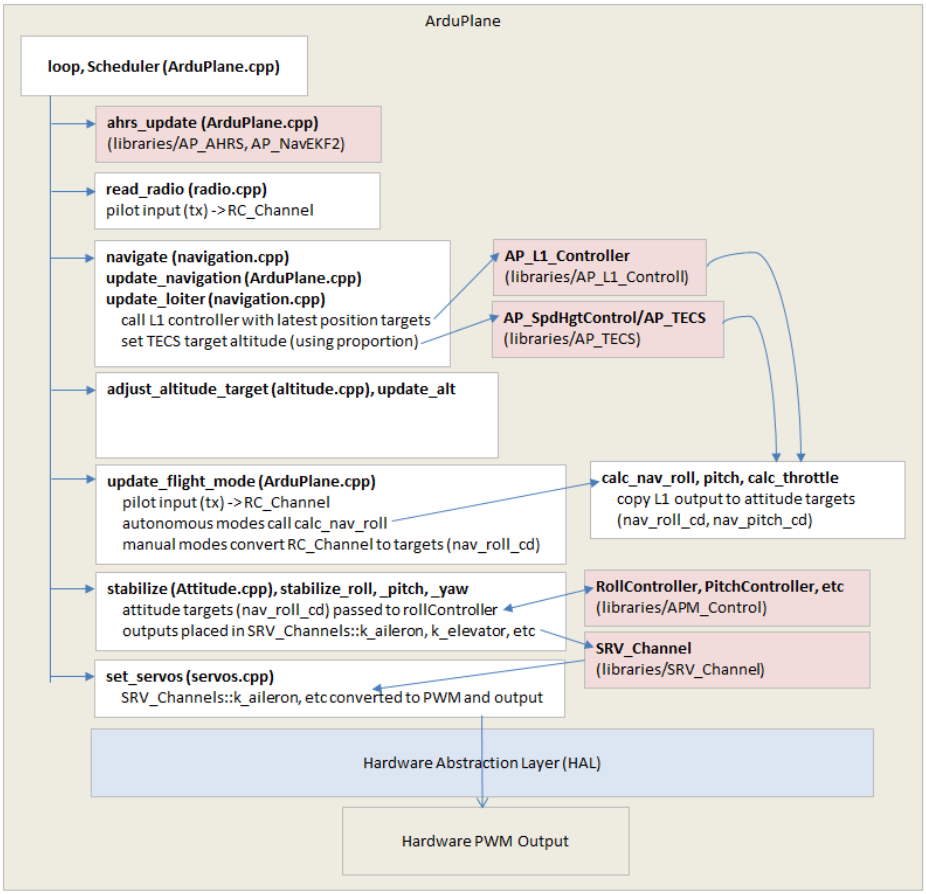


Figure 3.1 : ArduPilot's plane architecture [8].

given in Fig. 3.1 is explained as follows with the explanations from the ArduPilot documentation [8]:

- "*ahrs_update*" processes the last sensor measurement by using the Extended Kalman Filter (EKF), then it results in an estimation of the attitude and the position. An estimator is not included in this work.

- "*read_radio*" takes the input from RC controller on the ground, and based on these inputs it calculates the appropriate attitude or position targets. Even the flight mode is set to be "AUTO", the pilot commands can be overwritten. This feature helps a lot during the test flights.
- "*navigate*" calls high-level controllers which are explained below to convert the targets into references for the low-level controllers.
- "*update_flight_mode*" high level controller's attitude targets (roll and pitch) to "*nav_roll_cd*" and "*nav_pitch_cd*" global variables.
- "*stabilize*" runs the low level controllers.
- "*set_servos*" actuates the servos via PWM signals based on the output from the low-level controllers.

As shown in Fig. 3.2, there are 3 low level controllers and 2 high level controllers implemented inside ArduPilot's plane option. Their details will be given in the following sections, except the navigation controller (L1 controller). L1 and Total

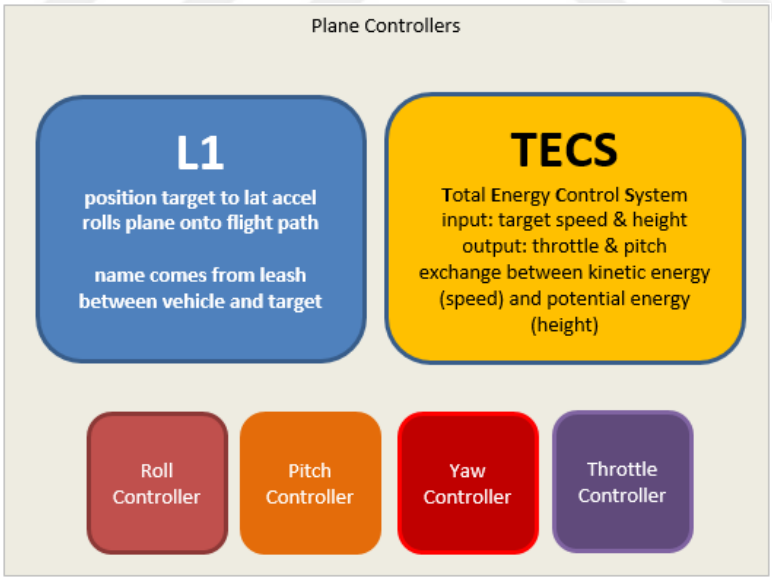


Figure 3.2 : ArduPilot's plane controllers [8].

Energy Control System (TECS) controllers are the high level controllers where they control the flight path, target airspeed and height. These high level controllers calculates the low level controllers inputs, so the ArduPlane has a nested structure.

All of the related parameters of ArduPilot is given in Appendix B.

3.1.1 Low-level controllers

The three different low-level controllers are introduced in the following sections, and among all of them, there is a scaling factor before obtaining the PID gains in ArduPilot. This factor is named scaler and it is equation is given as follows:

$$\text{scaler} = \frac{V_{a,\,\text{scaling}}}{V_a} \tag{3.1}$$

This factor reduces the risk of damaging the control surface mechanisms during higher speeds then the desired value.

3.1.1.1 Roll control loop

Updated ArduPilot's roll controller is given in Fig. 3.3. In this updated version, the spikes in servo output due to the pure derivative term has been eliminated.

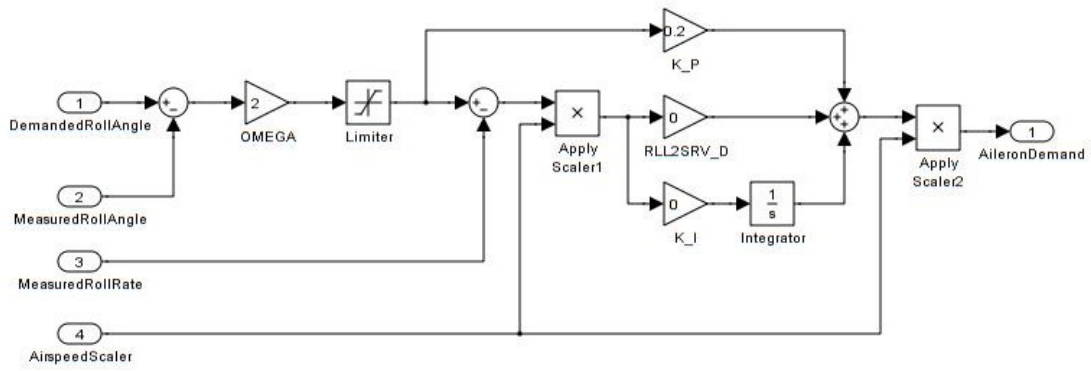


Figure 3.3 : ArduPilot's roll control loop block diagram [8].

In this new structure "OMEGA" indicates:

$$\Omega_\phi = \frac{1}{\tau_\phi} \tag{3.2}$$

In the old version, there was only a discrete PID controller and during the update the developers wanted to make us of these PID gains but also adding new control parameters too. These old gains are used as follows:

$$\tilde{K}_{P_\phi} = \left(K_{P_\phi} - K_{I_\phi}\tau\right)\tau - K_{D_\phi} \tag{3.3}$$

$$\tilde{K}_{I_\phi} = K_{I_\phi}\tau \tag{3.4}$$

In the new controller, the angular rate is taken into account instead of the angle measurement, which improved the performance. A new parameter τ is added as well, which has the limits between 0.4 and 1.0.

3.1.1.2 Pitch control loop

The pitch control loop follows a similar algorithm as mentioned for the roll. However, it also includes the turn compensation which is used when the roll angle does not have a zero value to eliminate the altitude loss results from the lift lose, which is called turn compensation [41]. ArduPilot's pitch controller is given in Fig. 3.4

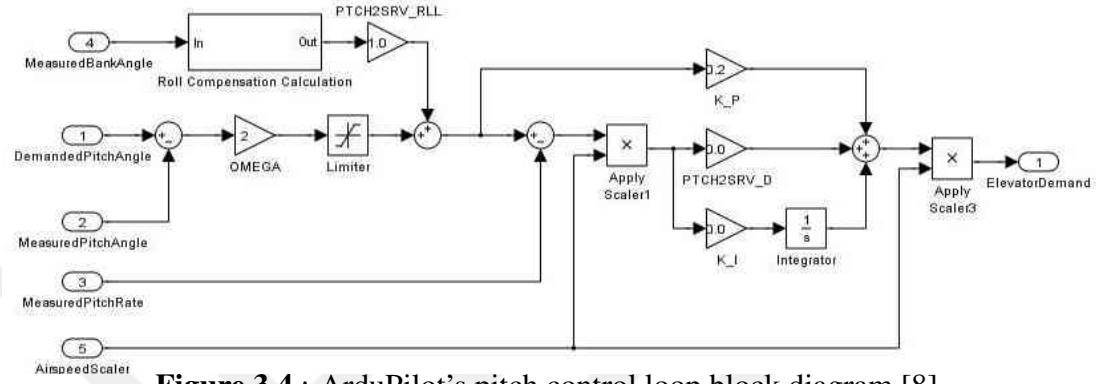


Figure 3.4 : ArduPilot's pitch control loop block diagram [8].

3.1.1.3 Side-slip control loop

ArduPilot's side-slip controller architecture is given in Fig. 3.5. It controls the rudder deflection and also the aileron deflection due to the mixing implementation which avoids the adverse yaw caused by aileron deflections.

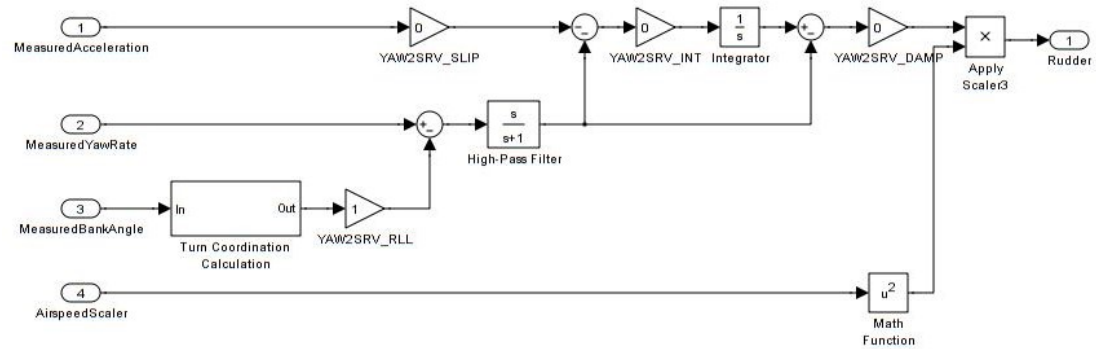


Figure 3.5 : ArduPilot's side-slip control loop block diagram [8].

In the ArduPilot, the yaw rate that brings the lateral force to zero is implemented as follows:

$$r_{\text{coord}} = \frac{g}{V_a} k_{r_{\text{coord}}} \sin\theta \tag{3.5}$$

where $k_{r_{\text{coord}}}$ is a tuned gain, which requires advanced knowledge so it is set to 1 in general.

The other low-level controller, throttle, will be introduced after the high level one.

3.1.2 High-level controllers

3.1.2.1 Total energy control system

Total Energy Control System (TECS) is the one preferred instead of decoupled dynamics, which has unsatisfactory results [40]. In the following, the ArduPilot implementation of TECS will be described.

Considering point mass approach, the specific kinetic and potential energies:

$$e_k \triangleq \frac{1}{2}V_a^2 \tag{3.6}$$

$$e_p \triangleq gh \tag{3.7}$$

And also, specific kinetic and potential energy rates are:

$$\dot{e}_k \triangleq V_a\dot{V}_a \tag{3.8}$$

$$\dot{e}_p \triangleq g\dot{h} \tag{3.9}$$

The energy values are obtained by the desired altitude and the airspeed:

$$e_{k,c} = \frac{1}{2}V_{a,c}^2 \tag{3.10}$$

$$e_{p,c} = gh_c \tag{3.11}$$

$$\dot{e}_{k,c} = V_{a,c}\dot{V}_{a,c} \tag{3.12}$$

$$\dot{e}_{p,c} = g\dot{h}_c \tag{3.13}$$

After defining all these, the specific total energy is introduced as follows:

$$e_T = e_k + e_p \tag{3.14}$$

where the specific total energy rate is:

$$\dot{e}_T = \dot{e}_k + \dot{e}_p = g\dot{h} + V_a\dot{V}_a \tag{3.15}$$

Instead of using this rate directly, TECS controllers use the scaled version of it:

$$\dot{e}'_T = \frac{\dot{e}_t}{gV_a} = \frac{\dot{V}}{g} + \frac{\dot{h}}{V_a} \tag{3.16}$$

The angle between the ground speed and the horizontal plane gives the flight path angle if there is no wind,

$$\dot{h} = V_g \sin\gamma = V_a \sin\gamma \tag{3.17}$$

Considering that for small angles, the scaled specific total energy becomes:

$$\dot{e}'_T = \frac{\dot{V}}{g} + \gamma \tag{3.18}$$

$$F_p - D = g\dot{e}'_T \tag{3.19}$$

where F_p is the thrust force, and D is the drag force.

Thus, the appropriate thrust control strategy can be achieved as follows:

$$\Delta F_p = g\Delta\dot{e}'_T \tag{3.20}$$

3.1.3 Pitch and thrust control

ArduPilot control scheme uses two PID controllers to generate the throttle and pitch commands.

$$T_{ff} = T_D + k_{T,ff}\dot{e}_{T,c} + k_{T_\phi}\left(\frac{1}{\cos^2\phi} - 1\right) \tag{3.21}$$

where

- T_D is the trim thrust needed to eliminate the drag force,
- $k_{T,ff}$ is a feedforward gain,
- k_{T_ϕ} is responsible for the increased drag during the banking.

$$\delta_t = T_{ff} + K_{p_{thr}}\tilde{e}_T + K_{d_{thr}}\dot{\tilde{e}}_T + K_{I_{thr}}\int_{t_0}^{t}\tilde{e}_T(\tau)\delta\tau \tag{3.22}$$

where where $\tilde{e_T} = e_{T,c} - e_T$ and $e_{T,c} = e_{k,c} + e_{p,c}$.

$$e'_b = (2-\varepsilon_{ke})\,e_p - \varepsilon_{ke}e_t = (2-\varepsilon_{ke})\,gh - \varepsilon_{ke}\frac{V_a^2}{2} \tag{3.23}$$

where ε_{ke} is a weighting factor to be used to assign priority to altitude or the airspeed during the pitch and thrust control, which is useful when there is no airspeed sensor connected or when the aircraft goes into a situation of loosing its airspeed. This weighting factor has the value between 0 and 2.0.

The commanded pitch is generated by a PID with a feedforward term and is given by

$$\theta_c = \frac{1}{V_a}\left(K_{P_\Theta}\tilde{e}_b + \frac{\dot{e}'_{b,c}}{g} + K_{D_\Theta}\dot{\tilde{e}}_b + K_{I_\Theta}\int_{t_0}^{t}\tilde{e}_b(\tau)\delta\tau\right) \tag{3.24}$$

3.2 MATLAB/Simulink Implementation

In this section, ArduPilot controllers are implemented in Simulink environment and their layouts are given in the following. Since there is no official Simulink library or toolbox developed for the ArduPilot, the diagrams from the documentation are used with the open-source code on GitHub [8]. Additionally, there is another implementation in the reference as well, but it is missing all the details about the model given inside the report [41].

Instead of giving each Euler angle separately, only the pitch controller layout is given. Other layouts and the Simulink models can be found from the thesis's GitHub repository provided. One of the low-level controller, pitch controller, is shown in Fig. 3.6.

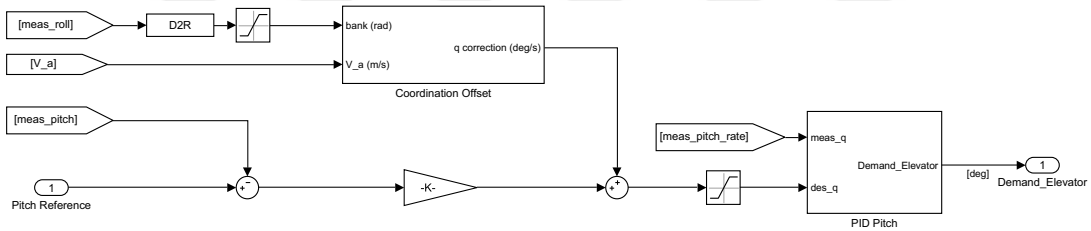


Figure 3.6 : Simulink pitch control loop block diagram.

The full model of the ArduPilot implemented in Simulink is shown in Fig. 3.7, and in addition to low level controllers the high level controllers are shown here additionally.

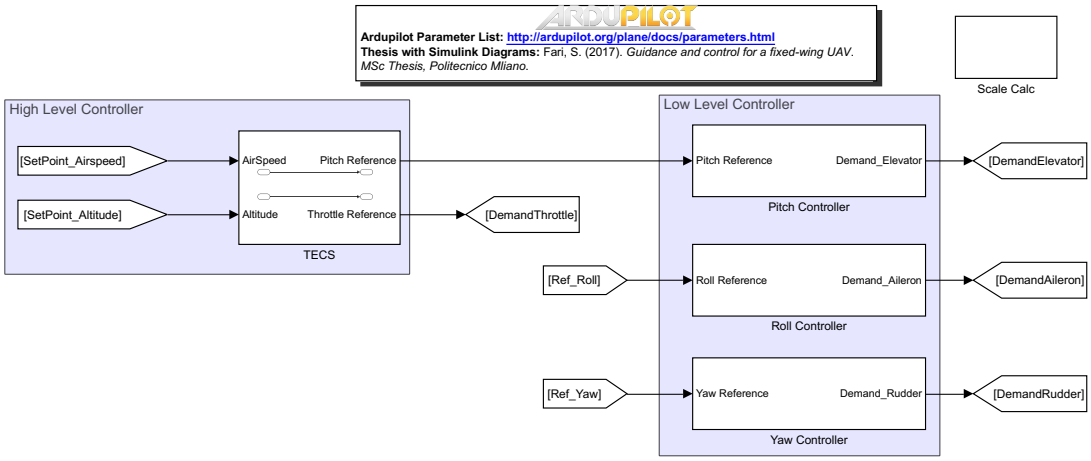


Figure 3.7 : Simulink pitch control loop block diagram.

3.3 Autopilot Hardware on UAS

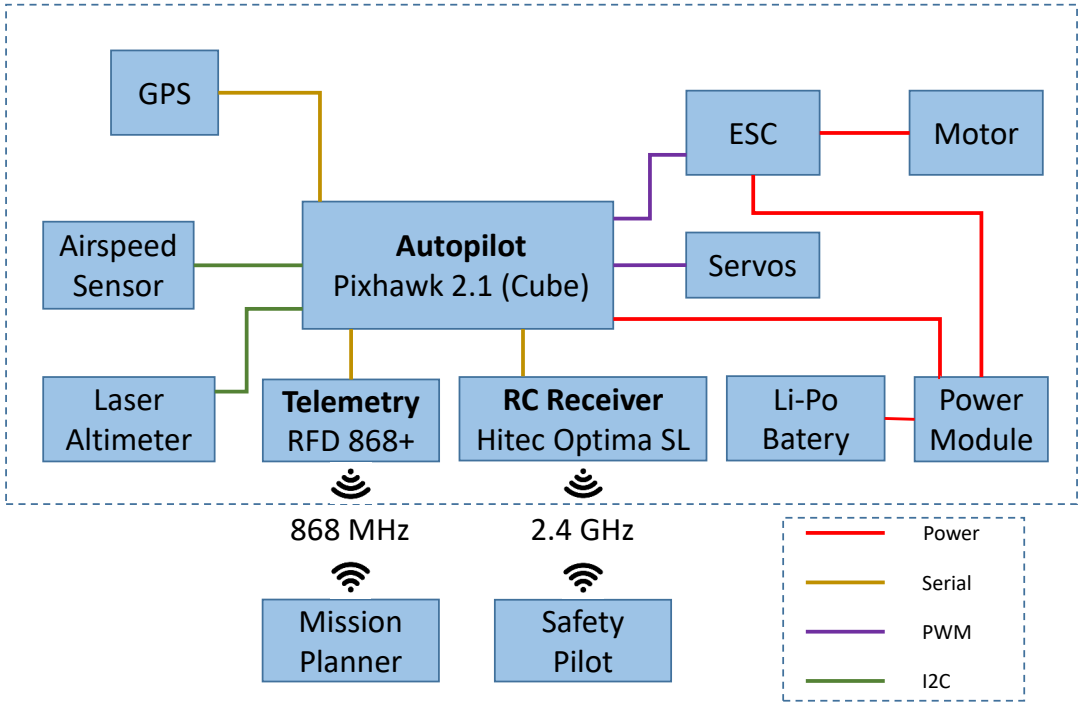


Figure 3.8 : Tulpar's flight control system architecture.

There are several autopilot boards can be used with Ardupilot software such as Pixhawk, PX4, ERLE BRAIN2, NAVIO+, APM (which is discontinued after ArduPilot v2) and among them Pixhawk 2.1 was chosen for Tulpar UAS due to its open-source structure and the wide-spread community where the one can find answers quickly.

Pixhawk 2 contains a triple-redundant IMU system and support for common sensors used in UAS. The ArduPilot firmware is used, because it is one of the best options due to open-source firmware currently available, provides great developer support, and has many essential flight features required for autonomous operation.

An open-source GCS software Mission Planner allows the use of a Software-In-The-Loop process, via SITL Simulator, which would be running the ArduPilot software used on the PixHawk 2, on a desktop computer.

3.3.1 Sensors

3.3.1.1 Autopilot internal sensors

The IMU system of Pixhawk 2.1 (the Cube) includes accelerometer, gyroscope, and magnetometer. Angle, angular velocity, and accelerometer value of the UAS were measured through these sensors.

3.3.1.2 Pitot tube

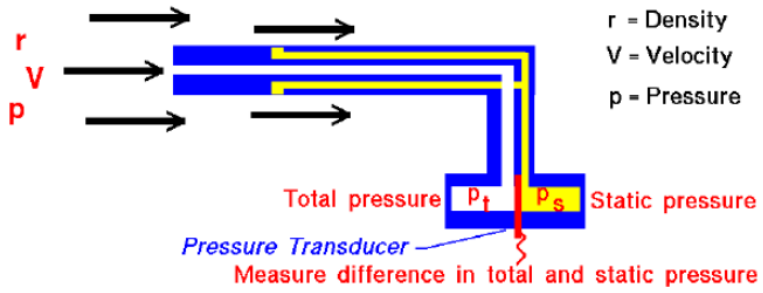


Figure 3.9 : Pitot Tube working principle [9].

The pitot tube is the airspeed sensor, which is also named as speedometers, used in the aircraft that measures the air velocity by comparing the static pressure of the surrounding air with the pressure at the tip of the tube, which is dynamic pressure as shown in Fig. 3.9 [9].

Bernoulli's Equation:

$$P_{\text{total}} - P_{\text{static}} = \frac{\rho V_a^2}{2} \tag{3.25}$$

and the air velocity is depicted as follows:

$$V_a = \sqrt{\frac{2(P_{\text{total}} - P_{\text{static}})}{\rho}} \tag{3.26}$$

mRo Next-Gen MS5525 Airspeed Sensor is chosen due to its improved accuracy and the additional feature of temperature compensation to other off-the-shelf options, and it is connected via I2C protocol to the Cube [42].

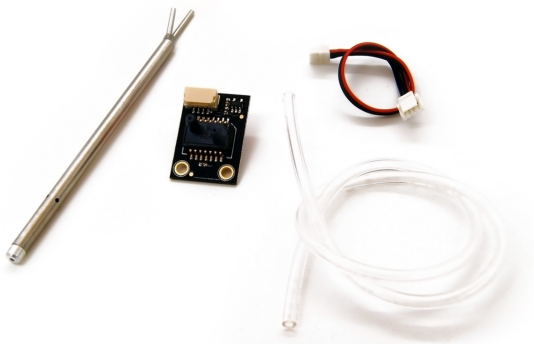


Figure 3.10 : mRo Next-Gen MS5525 airspeed sensor [10].

3.3.1.3 GPS and compass

Figure 3.11 : Here GPS that includes U-blox NEO M8N.

The GPS and compass are the essential sensors for the UAS navigation and course estimation. The Here GPS module preferred in this application is a U-blox NEO M8N, packed together with an extra 3-axis magnetometer [11]. During the estimation, the external magnetometer in Here GPS will be used instead of the internal one in the Cube to eliminate the possible adverse effects due to the magnetic field of other avionics near autopilot board inside the fuselage.

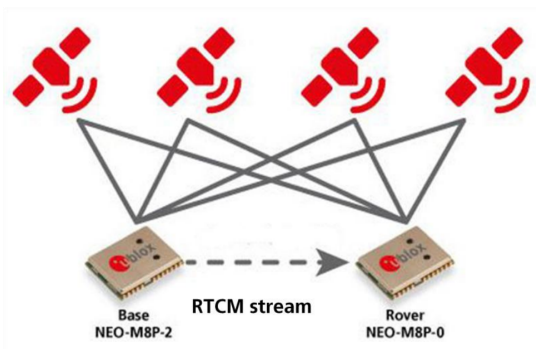


Figure 3.12 : The M8P modules work as a pair, where the Base provides a stream of RTCM messages to the aircraft [11].

About localization, Real-Time Kinematic (RTK) option could also be chosen for more accurate and precise applications, i.e. aerial mapping, and it results in cm-level GNSS performance by combining GPS mentioned above with U-blox's RTK technology [11].

3.3.2 Communication modules

These modules are required to track the UAS states during the flight, and also to prevent any misbehaviour.

3.3.2.1 Remote controller

As shown in Fig. 3.8, the safety pilot has the connection to the UAS all the time with a radio signal at 2.4 GHz via the remote controller on the ground side and the receiver on-board. The frequency band is permitted by the authorities in Turkey and also in the UK for amateur/hobby-style use. The remote controller is chosen to be Hitec Aurora 9X that has 9 channels and the receiver is chosen accordingly as Optima SL, which has 8 channels and also one S-Bus output to connect to Pixhawk 2.1 directly without using a PPM Encoder. The remote controller and the receiver can be found in Fig. 3.13.

After the development processes, the UAS has the capability of flying autonomously;

Hitec Aurora 9X Remote Controller

Hitec Optima SL

Figure 3.13 : Remote controller and the receiver.

however, during the development, the safety pilot and also the remote controller setup plays a crucial role. Therefore, selecting a well-known, high-quality brand like Hitec worth its additional expense based on the experience of Team Hedef.

3.3.2.2 Telemetry

In addition to the remote controller, there is another control point on the ground which is called Ground Control Station (GCS). It consists of a computer(s) and several antennas to deliver the information from the UAS simultaneously. The autopilot delivers the data via telemetry module, which consists of two identical modules on-board and also on the ground communication via 433 MHz, 868 MHz, or 900 MHz depending on the regulations. Since 868 MHz is permitted in Turkey and also in the UK, Team Hedef decided on that band instead of 433 MHz, where 900 MHz

is not an option except the USA. RFD868x or RFD868+ is chosen to be used as the telemetry module, and telemetry set's photo is shown in Fig. 3.14. Telemetry module is connected to GCS via USB using FTDI cable.

Figure 3.14 : RFD 868+ telemetry module set.

3.4 Flight Modes

There are different flight modes that a pilot can operate at, and for UAS operations it is important to make use of these modes during training, tuning, and also adverse weather conditions. The most common mode is called Fly-by-wire (FBW) which is used commonly in the aircraft regardless of its size/type, which is shown in Fig. ,too. Brief explanation about FBW is that it includes a flight computer that delivers the commands from the user,, which is pilot, but it merges these commands with the autopilot limitations on the deflection angles, angular rates, and the attitude.

One of the main features of ArduPilot is to let the user operate under different flight modes, which are interchangeable using a switch command on his radio controller. The main flight modes are:

- **Manual:** the radio controller stick commands of $\delta_a, \delta_e, \delta_r$, and δ_t are replicated to the control actuators as they are. The controller does not play any role, it is the pilot that closes the loop.

- **Fly-by-wire A (FBWA):** the FCU enables the control of the roll and pitch angles, whose reference is given by the user with the radio controller stick commands.
- **Fly-by-wire B (FBWB):** similar to FBWA, the FPU controls altitude and airspeed too, taking as inputs the airspeed and rate of climb from the user radio commands.
- **Autotune:** it is the same of FBWA mode, but meanwhile uses the aircraft response to tune the pitch and roll controllers and adjusts the control parameters online.
- **Auto:** the FPU enables also the guidance logic. The UAV will follow a mission (a set of GPS waypoints and other commands) set by the user.

4. Software-in-the-Loop-Simulation Environment

The mathematical model is simulated in MATLAB/Simulink environment. Before the simulation, the trim condition is calculated using the small perturbations theory for the given flight condition and features. In addition to custom visualization figure in Simulink, Flight Gear is introduced to be connected as an extra option. Besides that, several simulation environments are mentioned with their advantages/drawbacks; however, mainly Simulink's visualization figure has been used to eliminate the time-consuming tasks such as converting the aircraft drawings in computer-aided drawing software to the required versions. The simulation provides the UAS designer with information about the flight characteristics of the custom design vehicles. The flight analysis is not covered in this work, but it can be included easily in the simulation environment to have a feature to plan the missions with better approximations as well. Also, the simulation environment can be used as a pilot training tool as well by adding a USB Controller, which is another future work of this study.

4.1 SITL Options

4.1.1 Mission Planner

There is an internal SITL option in Mission Planner software, where the user can connect with X-Plane of JSBSim. On the other hand, there is a connection with MATLAB/ Simulink yet. This is one of the project ideas at Google Summer Coding School which works together with ArduPilot Developer Team.

4.1.2 MATLAB/Simulink and Flight Gear

The model developed in MATLAB/Simulink is connected to Flight Gear to visualize the UAS states. And also, the user inputs can be taken from USB Joystick so that the simulation environment can be used a pilot training or the responses of the aircraft can

be evaluated the test flight pilot directly from computer environment without going to the real test flight.

4.1.3 Mission Planner and X-Plane

Mission Planner can directly control an aircraft on X-Plane as well, and a screenshot from Team Hedef is shown in Fig. 4.1. The drawback of this simulation is that the airframe needs to be built in X-Plane Plane Maker, which is a cumbersome process like drawing the aircraft in CAD software from scratch as it is done during the thesis by Lidbom and Kiniklis at Chalmers University of Technology [43].

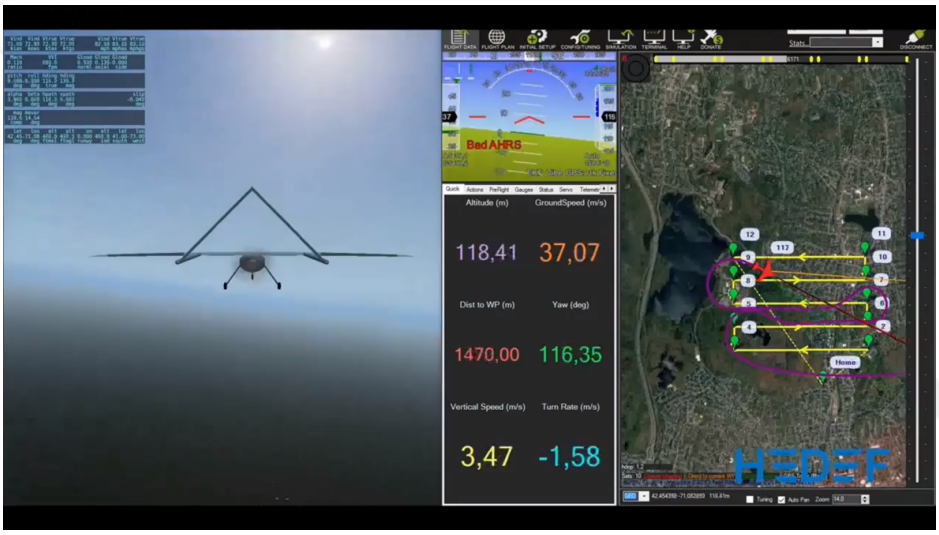


Figure 4.1 : Mission Planner & X-Plane simulation.

4.1.4 Other options

There are other options as listed below.

- ROS - Gazebo
- Real Flight 8
- AirSim
- JSBSim Open Source Flight Dynamics Model

Some of these options are still under development for fixed-wing UAS. Even though Real Flight provides a useful facility for pilot training, it is not a freeware software which makes is limited. JSBSim supports internal combustion engines, therefore it

is not useful for the electric motor cases. Robot Operating System (ROS) - Gazebo simulator is becoming even more popular with its growing community. AirSim uses the Unreal Engine which brings the benefits of having a detailed environment model to test the air vehicles. Also, ROS - Gazebo, and AirSim could be used to create collaborative or formation flight applications as well.

4.2 MATLAB/Simulink Simulation Results

The UAS figure according to its flight states is shown simultaneously on Simulink, and a sample is shown in Fig. 4.2.

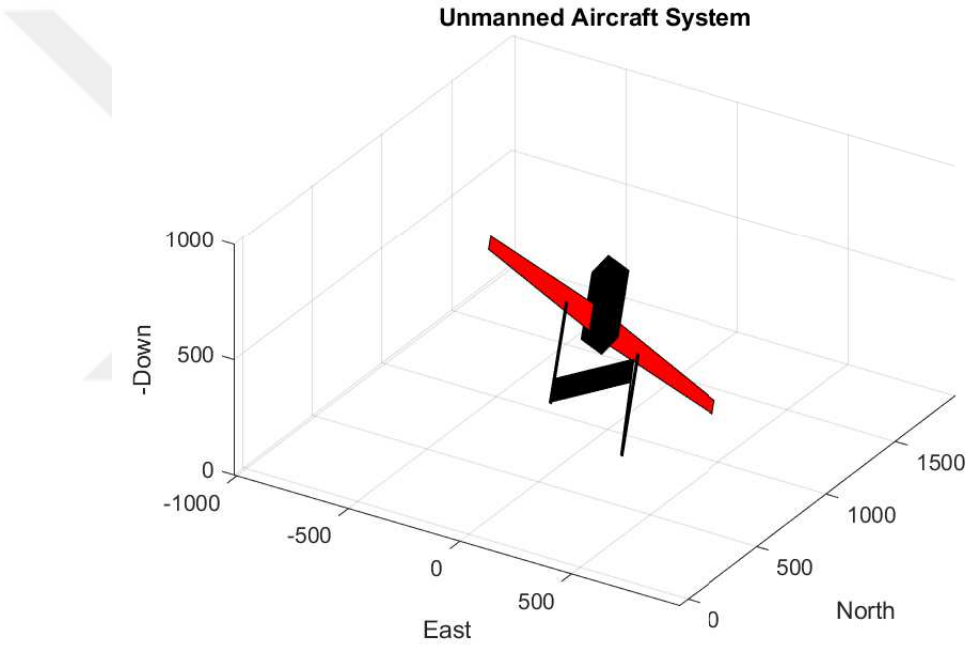


Figure 4.2 : Simulink scope of Tulpar visulization.

To see all of the states in one figure window including the reference inputs to the actuators, the plot file provided by Dr. Beard's book is modified to be used in this thesis Fig. 4.3.

Aircraft crash warning is included inside Simulink to avoid irrelevant results during the simulation.

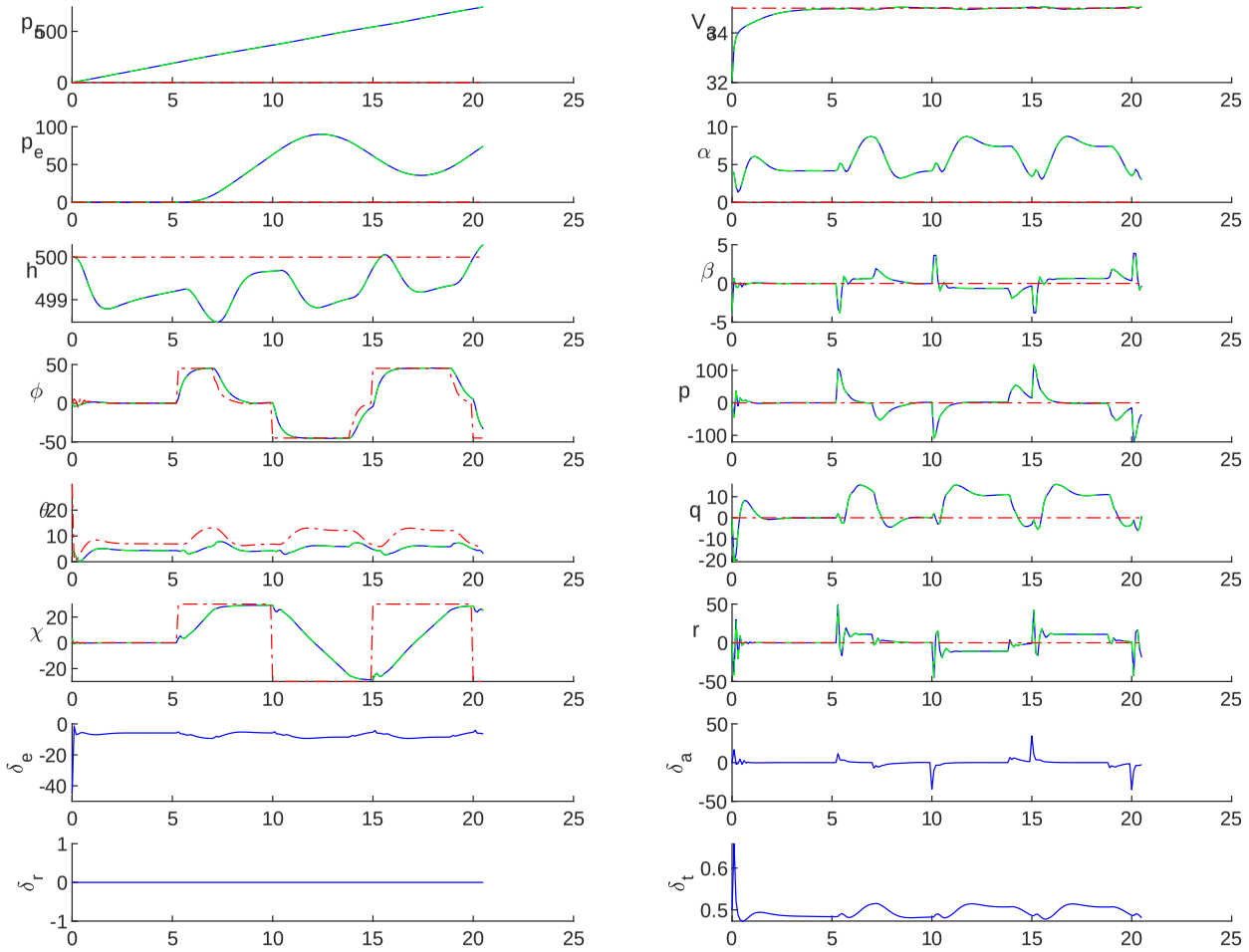


Figure 4.3 : Simulink scope of the UAS states.

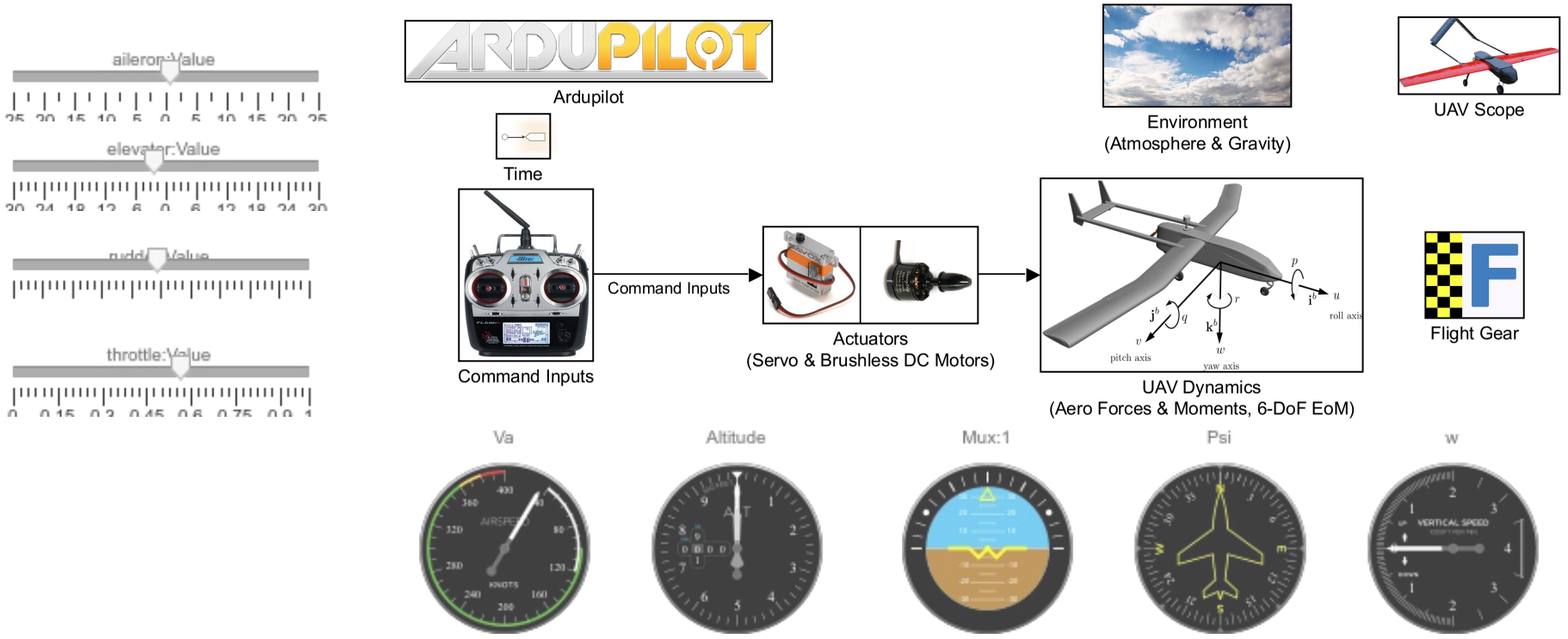


Figure 4.4 : Simulink layout of the main model.

5. Test Flight Validation

The developed model is validated with test flights as mentioned in the objectives of the thesis. In this section, additional equipment is introduced and the test flight scenarios are explained. The test flights are conducted at Hezarfen airport according to these scenarios, and the results are compared with the simulation. Simulation is the ideal environment, and since it is obtained via modeling with a lot of assumptions and approximations, there exists a difference with real-life results. Also, the platform is manufactured by ITU Hedef UAS Team with low-cost facilities and limited calendar time, so there exist minor differences due to that condition, too. Just a reminder to the reader, this work is conducted to validate the aerodynamic model of the Tulpar, and the performance calculations are not in the scope of it. It could be included in future works.

5.1 Test Flight Equipment & Procedure & Lessons Learned

5.1.1 Equipment

Ground Control Station (GCS) is the platform that the operator can monitor the flight from ground live. GCS computer is chosen to be an ultrabook, Asus UX310UQ-FB418T, due to its light-weight, sun-readable screen, and long endurance battery. The RFD telemetry is connected via USB Serial port to the GCS, and to eliminate the connection issues during battery changes, a simple switch is included so that the user does not need to plug off the USB cable.

5.1.2 Procedure

According to the Directorate General of Civil Aviation (DGCA) Unmanned Aircraft System Instruction (SHT-IHA), UASs weight more than 500 g must be registered in their system and their operations are strictly limited and forbidden in most the of land in Turkey [44]. A detailed checklist was prepared based on experience and other

examples and before each flight, it was mandatory to go over this list. The most critical part is pilot comments during or after the flight, especially if you design your own UAS. Therefore, having an experienced pilot like Yusuf Demiroğlu in Team Hedef was valuable.

Figure 5.1 : Hezarfen RC Model Club's airfield from the air.

The test flight area is located at Hezarfen Airfield shown in Fig. 5.1, which is owned by Hezarfen RC Club. Since it takes 65 km to reach the destination from campus, the equipment checklist was created and followed before the trips.

5.1.3 Lessons learned

Custom built, low-cost UASs like Tulpar are always open to malfunctioning because the subsystems are bought from hobby-shop retailers. Therefore, having an experienced safety pilot is crucial and extremely beneficial for similar works. Also, one shall never underestimate the risks, and always conduct detailed control before the operation.

- The checklist shall be followed before each flight, even it is a sequential flight. There might have been some parts missed in the air or the landing might have affected the aerial vehicle, as well. One of our flights with the prototype, the fuselage was broken when we landed, but we proceeded to the next flight directly by changing the battery, which resulted in a crash in the air.

- While trying new actuators, brushless motors or servo motors, the flight must be observed very carefully. Ground tests might not be sufficient or there might. One time while we were trying a new motor, it just stopped after 5 minutes and the aircraft lost propulsive control.

- Before the flight, the emergency situations should have been discussed in details, especially for the test flight of missions like payload delivery, etc. Several times we had issue with payload release mechanism even though we tested them on ground before take-off. Therefore, obtaining the emergency action before take-off is crucial.

- Always pay attention to file naming for your flight logs and write down a description in the flight logbook. After a while there will be huge number of log files, so keeping track of them will be difficult.

In addition to the experiences mentioned above, a more comprehensive source would be University of Southampton's Unmanned Aerial Vehicles Team experiences mentioned in a relatively new book [45]. They have been working on UAVs since 2008, I think their valuable experiences are worth mentioning.

5.2 GCS Software

Mission Planner version 2.0 [18] is a full-featured ground station application for the ArduPilot open source autopilot project. Developed by Michael Osborne, it is part of the ArduPilot project. It can be used as a configuration utility or as a dynamic control supplement for autonomous vehicles. The mission interface of APM is shown in the Figure 3.1. The main uses to which APM mission planner is used are:

- Load the firmware (the software) into the autopilot (APM, PX4...) that controls the vehicle.

- Setup, configure, and tune the vehicle for optimum performance.

- Plan, save and load autonomous missions into the autopilot with simple point-and-click way-point entry on Google maps or other.

- Download and analyze mission logs created by the autopilot.

- Interface with a PC flight simulator to create a full hardware-in-the-loop UAV simulator.
- With appropriate telemetry hardware, monitor the vehicle's status while in operation. It is the platform through which the code and initial configuration of the vehicle are uploaded to the PixHawk.

5.3 Data Log Analysis

During the flight, ArduPilot sends telemetry data to the GCS via Pixhawk and RFD telemetry unit and the data is saved as a telemetry log file as well. However, due to the limited rate, telemetry log files lack detailed information with higher frequencies about the flight. Therefore, after the landing data logs are required to be downloaded and then analyzed in detail. These files are in binary format and Mission Planner interface allows the user to review them directly. This method is pretty slow compared to parsing the data in MATLAB or Python. MATLAB is preferred to be used for the post-flight analysis, and binary log data can be converted to ".mat" format by using Mission Planner.
Using Mission Planner converting function will result in a log file without real-time stamps, there are several custom tools developed for that purpose can be found with a quick search online. Ardupilog developed by George Zogopoulos – Papaliakos is the one preferred, the code can be found from the developer's repository [46].

5.3.1 Important log classes

The following log classes are observed during the analysis of the flight dynamics of an aircraft.

- ARSP (Airspeed):
  - Airspeed: Horizontal air speed in m/s.
- ATT (Attitude):
  - DesRoll: The pilot's desired roll angle in degrees (roll left is negative, right is positive)
  - Roll: The vehicle's actual roll in degrees (roll left is negative, right is positive)

  - DesPitch: The pilot's desired pitch angle in degrees (pitch forward is negative, pitch back is positive)
  - Pitch: The vehicle's actual pitch angle in degrees (pitch forward is negative, pitch back is positive)
  - DesYaw: The pilot's desired heading in degrees with 0 = north
  - Yaw: The vehicle's actual heading in degrees with 0 = north

- GPS:
  - Lat: Lattitude according to the GPS
  - Lng: Longitude according to the GPS
  - NSats: The number of satellites current being used
  - SPD: Horizontal ground speed in m/s

- RCOUT (pwm output to individual RC outputs):
  - Channel 1: Aileron Left
  - Channel 2: Ruddervator Right
  - Channel 3: Throttle
  - Channel 4: Ruddervator Left
  - Channel 5: Aileron Right
  - Channel 6: Flaperon Right
  - Channel 7: Flaperon Left

5.3.2 Test flight scenarios and results

The following scenarios are observed during the test flight.

- Sinusoidal Inputs by Safety Pilot
- Doublet Inputs by Safety Pilot
- Loiter Flight by Autopilot
- Altitude Tracking by Autopilot
- Airspeed Tracking by Autopilot

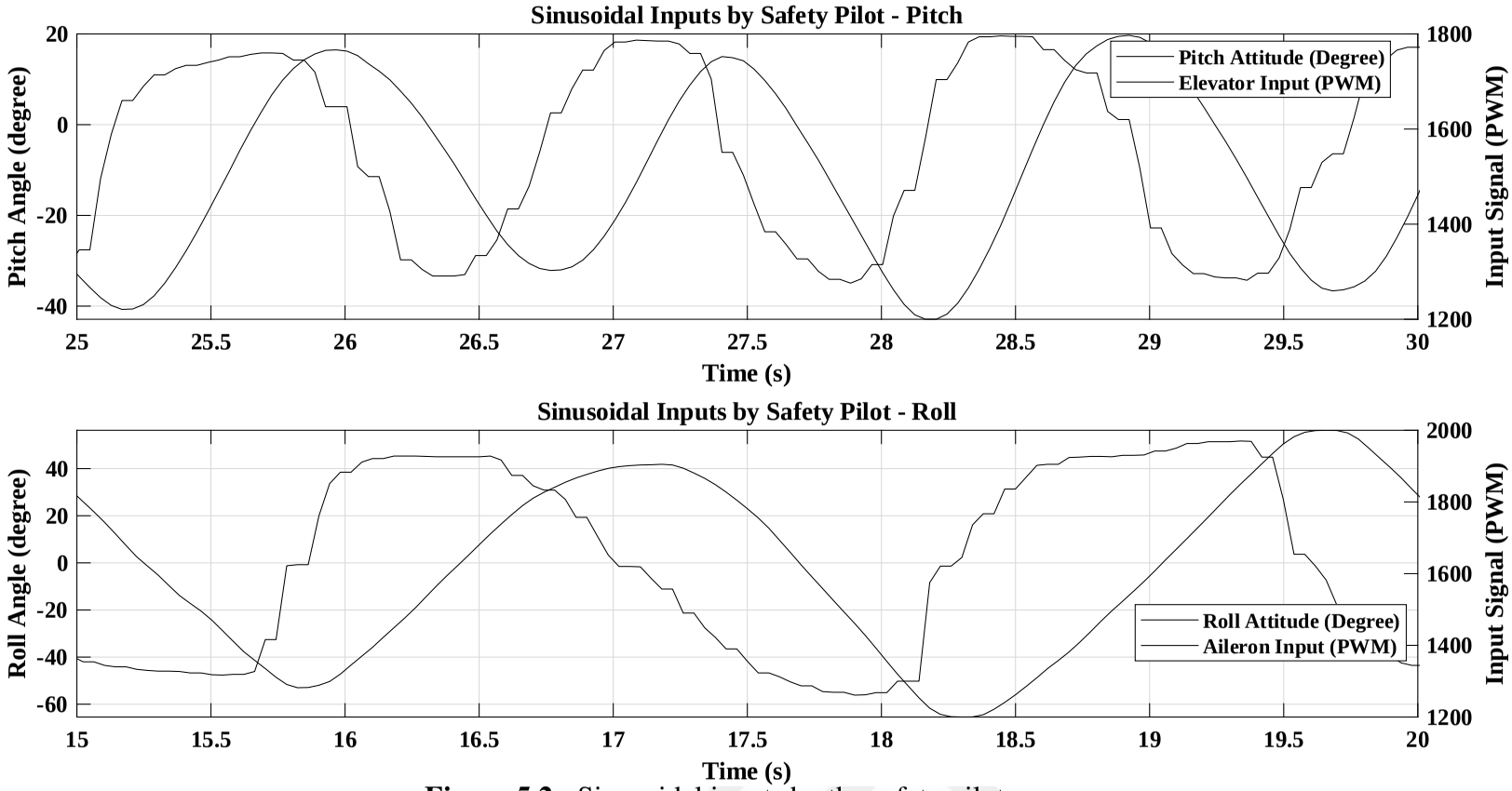


Figure 5.2 : Sinusoidal inputs by the safety pilot.

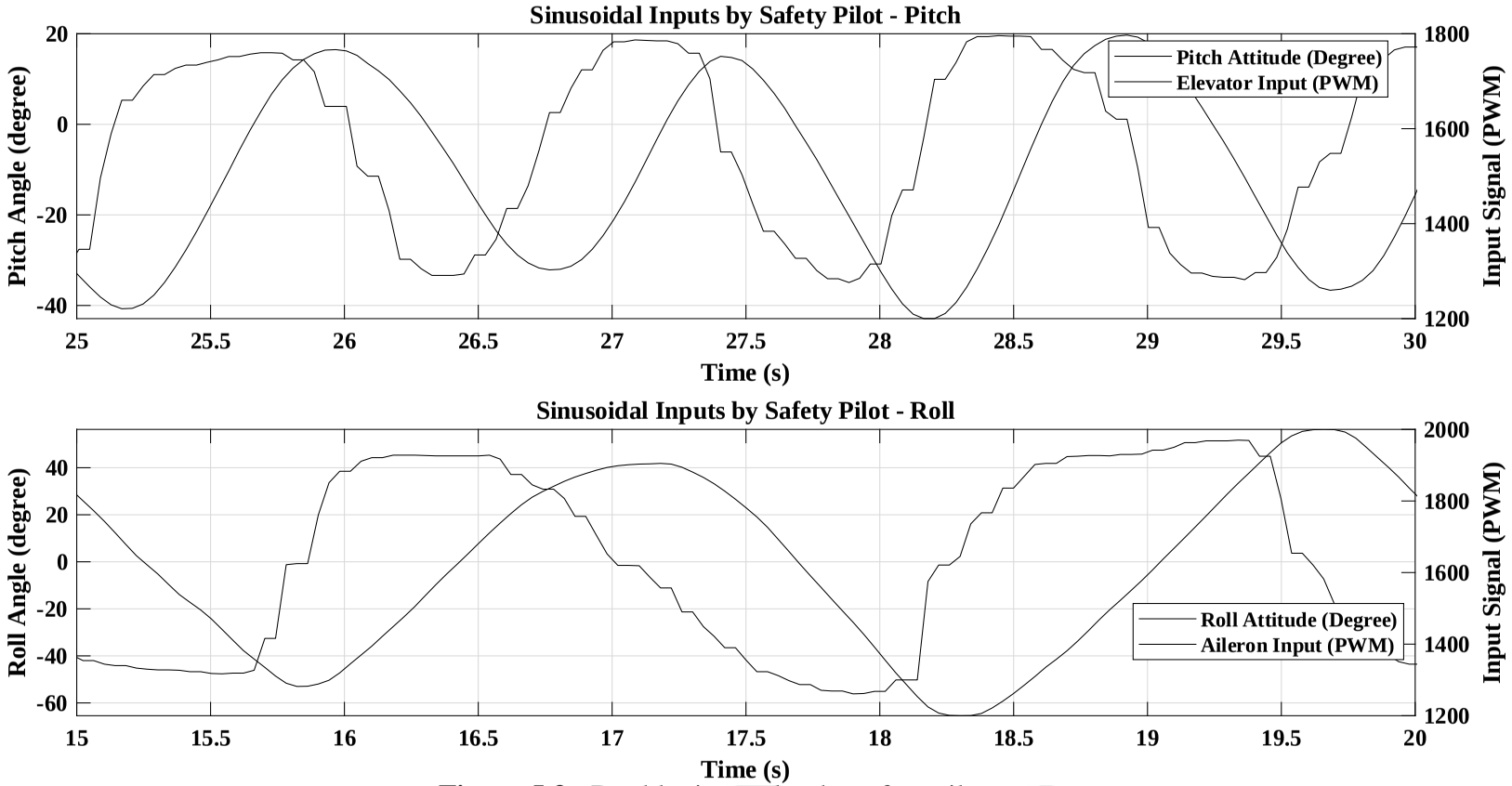


Figure 5.3 : Doublet inputs by the safety pilot.

5.4 Log Analysis Example on Pixhawk Failure

During one of the test flights, there was no control because power brick was down at an altitude of 107 m. Before Pixhawk, Pitot Tube was down, and then GPS was down afterwards. I2C port is not working after this failure, so board related failure. The following power flags are observed in the log data:

- Flag 33 – Binary: 0b100001
  - Bit0 = Power module is active
  - Bit5 = Power status changed
- Flag 41 – Binary: 0b101001
  - Bit0 = Power module is active
  - Bit3 = Peripheral supply is in over-current state
  - Bit5 = Power status changed

In the following page, altitude of this flight is plotted and the flag numbers for power brick, GPS, and pitot tube is shown on the other axis in Fig. 5.5, and the crash's flight path is shown in 5.4.

Figure 5.4 : Flight path of the crash occurred on 14 June 2019.

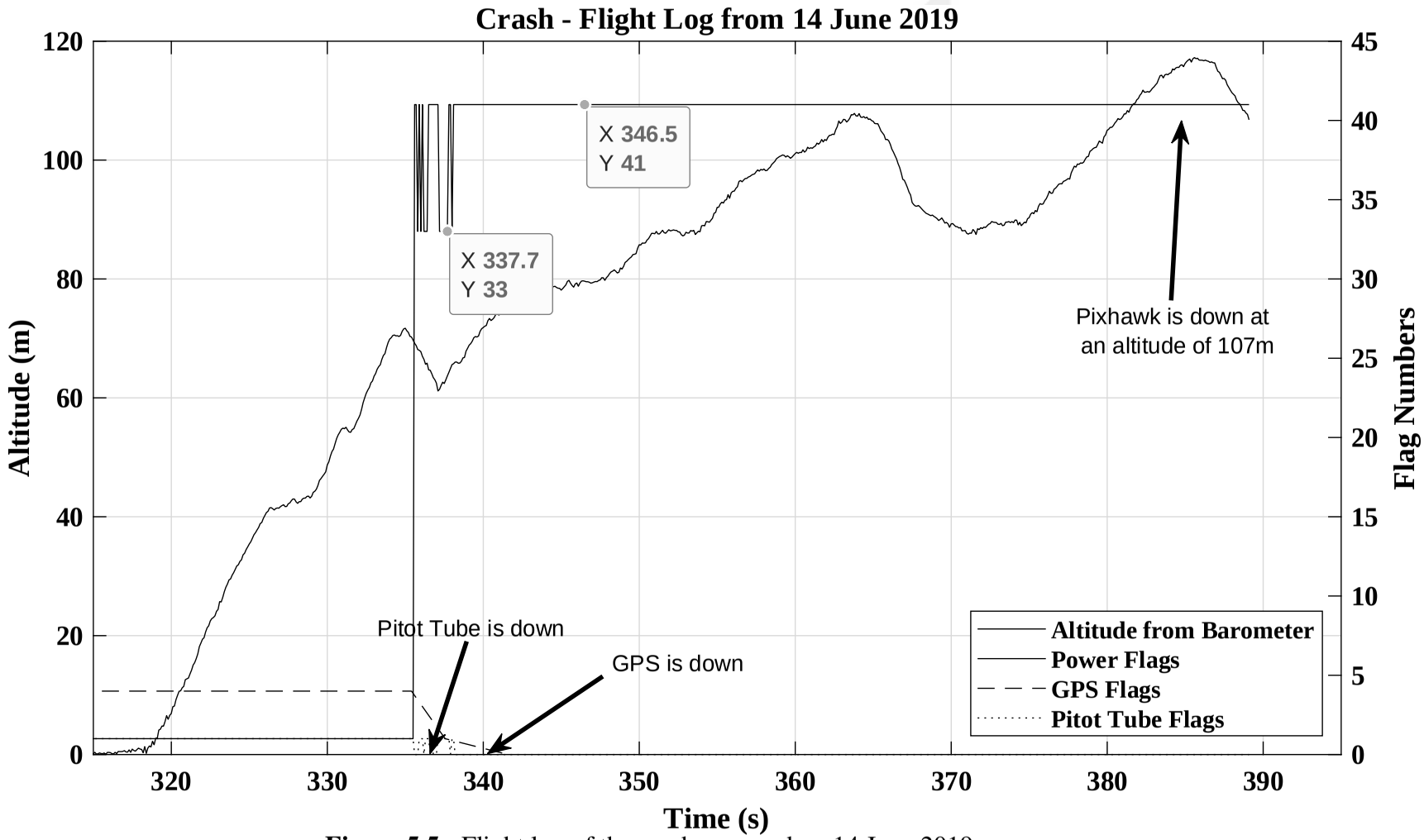


Figure 5.5 : Flight log of the crash occurred on 14 June 2019.

6. CONCLUSIONS AND RECOMMENDATIONS

After conducting a detailed literature survey, it has been seen that there are a lot of papers and works in the modeling of UAS. However, modeling a custom design UAS with limited facilities and budget and testing the model with test-flight was rare. The main purpose of this work was to obtain a high-fidelity MATLAB/Simulink model which will be a validated model to be used for the future works of the MoDeL Lab. Team Hedef's Grand Champion UAS Tulpar was modeled, and the developed model is aimed to be integrated into undergraduate courses to allow students gain hands-on experience and understanding of flight dynamics and Guidance, Navigation, and Control.

Wind tunnel test or CFD analysis are required for more accurate stability derivatives. Since we obtained them via using VLM in XFLR5, the results are not accurate enough. To eliminate this, DATCOM analysis would be helpful to have a better idea on drag analysis, which would also enable the user to have an idea on the performance, too.

ArduPilot lacks the implementation in MATLAB/Simulink environment. Creating block diagrams of this open-source autopilot software is useful for the future works as well. After this implementation, a code generation toolbox would be useful to develop algorithms and to test them quickly on the aerial vehicle. Also, ROS implementation would be very beneficial as well.

The developed model lacks the guidance model. ArduPilot's guidance algorithm implementation could be done in MATLAB/Simulink to work on its performance. According to the experience from test flights, there could be an improvement especially about the waypoint accuracy in this guidance algorithm, and testing the novel methods on simulation would save a lot of time and money.

In the model, the estimator is not included so the measurement data is taken directly as input to the controllers. To examine the sensor effects such as bias and noise, it would be beneficial to include an estimator in the model.

After all these implementations, mission planning tasks could be achieved using this high-fidelity simulation environment. Performance estimation and optimization tool can be developed afterward to create an all-in-one software package on UAS development.

In this study, it is also aimed to share the experiences during this project work with others, so the reader can find additional information in the appendices about SUAS operations, such as checklists, lessons learned, additional resources. It is very crucial not to reduce the safety level because SUAS with low-cost equipment is always prone to malfunctioning. Therefore, conducting flights at the registered fields is a must for any research team.

If you reported an issue/error in the thesis to the author and would like to check the updated document, please check the following link: bit.ly/ondes-thesis-msc-pdf

Also, if the research group decides to share the files used in this thesis, the reader can find them in the following link: bit.ly/ondes-thesis-msc-files

REFERENCES

[1] **Misra, P. and Enge, P.** (2006). *Global Positioning System: Signals, Measurements and Performance*, Ganga-Jamuna Press, Massachusetts, 2 edition.

[2] **Langelaan, J.W.**, **Alley, N. and Neidhoefer, J.** (2010). Wind Field Estimation for Small Unmanned Aerial Vehicles.

[3] **Chabot, D.** (2018). Trends in drone research and applications as the Journal of Unmanned Vehicle Systems turns five, *Journal of Unmanned Vehicle Systems*, *6*(1), vi–xv, http://www.nrcresearchpress.com/doi/10.1139/juvs-2018-0005.

[4] **Leutenegger, S.**, **Hürzeler, C.**, **Stowers, A.K.**, **Alexis, K.**, **Achtelik, M.W.**, **Lentink, D.**, **Oh, P.Y. and Siegwart, R.**, (2016). Flying Robots, Springer International Publishing, Cham, pp.623–670, https://doi.org/10.1007/978-3-319-32552-1{_}26.

[5] **Karaağaç, C.**, (2018), Türkiye İHA Sektörü 2018, https://twitter.com/myDRONEland/status/1021333547418824705.

[6] **Noth, A.** (2007). Design of Solar Powered Airplanes for Continuous Flight, *ETH University*, (December), 18.

[7] **Unmanned Dynamics Inc.**, (2006), AeroSIM Aeronautical Simulation Blockset, Version 1.2., http://www.u-dynamics.com/.

[8] **ArduPilotDevTeam**, (2019), ArduPilot Project - Source Code, https://github.com/ArduPilot/ardupilot.

[9] **Benson, T.**, Pitot Tube, https://www.grc.nasa.gov/WWW/K-12/VirtualAero/BottleRocket/airplane/pitot.html.

[10] **MRobotics**, (2019), mRo Next-Gen MS5525 Airspeed Sensor, https://store.mrobotics.io/mRo-Next-Gen-MS5525-Airspeed-Sensor-NEW-p/mro-ms5525v2-mr.htm.

[11] **U-blox**, (2017), NEO-M8P u-blox M8 High Precision GNSS Modules Data Sheet, https://www.u-blox.com/sites/default/files/NEO-M8P{_}DataSheet{_}{%}28UBX-15016656{%}29.pdf.

[12] **AltiGator**, Drone, UAV, UAS, RPA or RPAS ..., https://altigator.com/drone-uav-uas-rpa-or-rpas/.

[13] **DefenceNews**, (2019), Top 100 for 2018, https://people.defensenews.com/top-100/.

[14] **Institution of Mechanical Engineers**, (2018), Istanbul Technical University wins UAS Challenge drone competition, http://www.imeche.org/news/news-article/istanbul-technical-university-wins-student-drone-design-competit

[15] **Shen, J.**, **Su, Y.**, **Liang, Q. and Zhu, X.** (2018). Calculation and Identification of the Aerodynamic Parameters for Small-Scaled Fixed-Wing UAVs, *Sensors*, *18*(2), 206, http://www.mdpi.com/1424-8220/18/1/206.

[16] **Ostler, J.N.**, **Bowman, W.J.**, **Snyder, D.O. and McLain, T.W.** (2010). Performance Flight Testing of Small, Electric Powered Unmanned Aerial Vehicles, *International Journal of Micro Air Vehicles*.

[17] **Foster, T. and Bowman, J.** (2005). Dynamic Stability and Handling Qualities of Small Unmanned- Aerial Vehicles, ***Intergovernmental Panel on Climate Change**, editor, 43rd AIAA Aerospace Sciences Meeting and Exhibit*, volume 53, American Institute of Aeronautics and Astronautics, Reston, Virigina, pp.1–30, https://arc.aiaa.org/doi/abs/10.2514/6.2005-1023http://arc.aiaa.org/doi/10.2514/6.2005-1023.

[18] **Brandt, J.B.** (2005). Small-Scale Propeller Performance At Low Speeds, *Ph.D. thesis*.

[19] **APCPropellers**, Performance Data, https://www.apcprop.com/technical-information/performance-data/.

[20] **Verstraete, D. and Macneill, R.** (2016). The Effects of Blockage on the Performance of Small Propellers, *Proceedings of the 27th ISABE Conference*, (December), 5–8.

[21] **Huria, T.**, **Ceraolo, M.**, **Gazzarri, J. and Jackey, R.** (2012). High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells, *2012 IEEE International Electric Vehicle Conference*, IEEE, pp.1–8, http://ieeexplore.ieee.org/document/6183271/.

[22] **Paw, Y.C.** (2009). Synthesis and validation of flight control for UAV, *Ph.D. thesis*.

[23] **Jung, D.** (2007). Hierarchical path planning and control of a small fixed-wing UAV: Theory and experimental validation, *Ph.D. thesis*, https://smartech.gatech.edu/handle/1853/19781.

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

[25] **Cotting, M.C.** (2010). Evolution of Flying Qualities Analysis: Problems for a New Generation of Aircraft, *Ph.D. thesis*.

[26] **Kabbabe, K.P.** (2011). Development of Procedures for Flight Testing Uavs Using the Ardupilot System, *Ph.D. thesis*, http://uavs.mace.manchester.ac.uk/files/2513/7971/0471/MScKabbabe2011.pdf.

[27] **Cook, M.** (2012). *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*, Elsevier.

[28] **Stevens, B.L.**, **Lewis, F.L. and Johnson, E.N.** (2015). *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systemstle*, Wiley, 3rd edition.

[29] **Etkin, B.** (2012). *Dynamics of atmospheric flight*, Courier Corporation.

[30] **Nelson, R.** (1989). *Flight stability and automatic control*, McGraw-Hill series in aeronautical and aerospace engineering, McGraw-Hill Ryerson, Limited, https://books.google.gr/books?id=1YhTAAAAMAAJ.

[31] **Napolitano, M.** (2011). *Aircraft Dynamics: From Modeling to Simulation*, CourseSmart Series, Wiley, https://books.google.gr/books?id=VMij4rR-DzoC.

[32] **Vogeltanz, T.** (2016). A Survey of Free Software for the Design, Analysis, Modelling, and Simulation of an Unmanned Aerial Vehicle, *Archives of Computational Methods in Engineering*, *23*(3), 449–514, http://link.springer.com/10.1007/s11831-015-9147-y.

[33] **Rubio, S.R.** (2013). Aerodynamic analysis of a sustainable-energy sailplane using CEASIOM, *Ph.D. thesis*.

[34] **DassaultSystems**, CATIA Documentation - Displaying the Assembly Mass Properties, http://catiadoc.free.fr/online/cfyugasm{_}C2/cfyugasmut1000.htm.

[35] **Inman, D.** (2001). *Engineering Vibration*, Prentice Hall, https://books.google.gr/books?id=bHIeAQAAIAAJ.

[36] **Lundström, D.**, **Amadori, K. and Krus, P.** (2010). Validation of Models for Small Scale Electric Propulsion Systems, *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Reston, Virigina, http://arc.aiaa.org/doi/10.2514/6.2010-483.

[37] **Verstraete, D. and Macneill, R.** The Effects of Blockage on the Performance of Small Propellers, **Technical Report**.

[38] **NASA** (1976). U.S. Standard Atmosphere, **Technical Report**, https://ntrs.nasa.gov/search.jsp?R=19770009539.

[39] **ArduPilotDevTeam**, (2019), List of Suggested Projects for GSoC 2019, http://ardupilot.org/dev/docs/gsoc-ideas-list.html.

[40] **Faleiro, L.F. and Lambregts, A.A.** (1999). Analysis and tuning of a 'total energy control system' control law using eigenstructure assignment, *Aerospace Science and Technology*, *3*(3), 127–140, http://dx.doi.org/10.1016/S1270-9638(99)80037-6.

[41] **Fari, S.** (2017). Guidance and control for a fixed-wing UAV, *Ph.D. thesis*.

[42] **Connectivity, T.**, (2018), MS5525DSO Specifications, https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc{&}DocId=Data+Sheet{%}7FMS5525DSO{%}7FD2{%}7Fpdf{%}7FEnglish{%}7FENG{_}DS{_}MS5
pdf.

[43] **Lidbom, A. and Kiniklis, E.** (2015). Providence - UAV system to support search and rescue.

[44] **DGCA**, (2016), İNSANSIZ HAVA ARACI SİSTEMLERİ TALİMATI (SHT-İHA), http://web.shgm.gov.tr/documents/sivilhavacilik/files/mevzuat/sektorel/talimatlar/SHT-IHA{_}WEB.pdf.

[45] **Keane, A.J.**, **Sóbester, A. and Scanlan, J.P.** (2017). *Small Unmanned Fixed-wing Aircraft Design: A Practical Approach*, John Wiley & Sons.

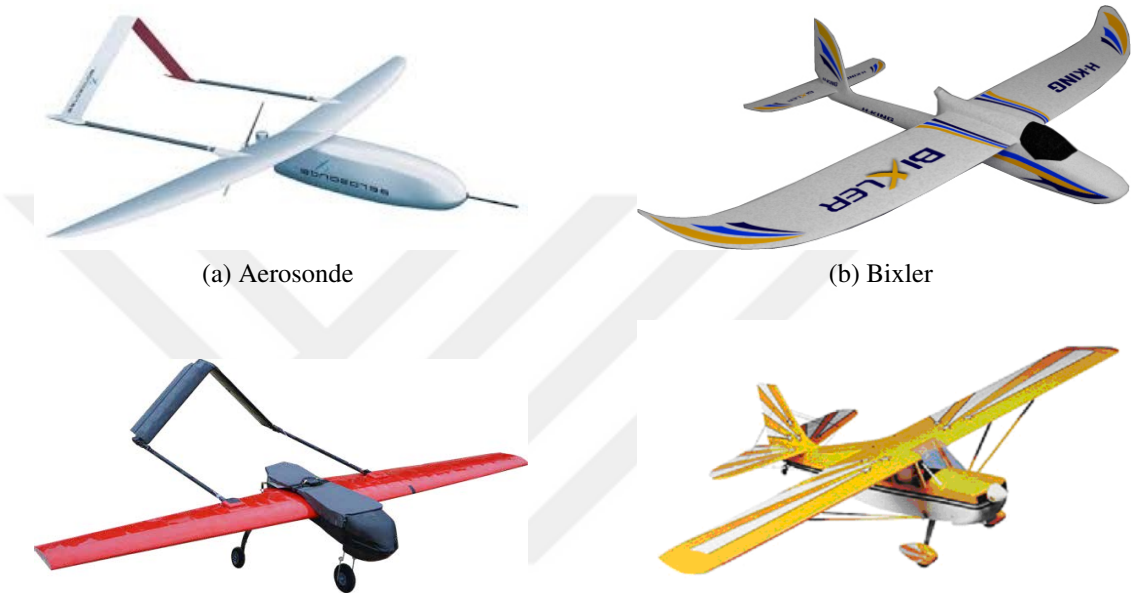
[46] **Zogopoulos-Papaliakos, G.**, (2019), Ardupilog, https://github.com/Georacer/ardupilog.

APPENDICES

APPENDIX A: UAS Parameters - Tulpar
APPENDIX B: ArduPilot Parameters
APPENDIX C: Flight Checklists

APPENDIX A: UAS Airframe Parameters

This appendix includes the dimensional, aerodynamic, mass, propulsion parameters of Tulpar, and also the other three different UAS shown in Fig. A.1 can be found using the references [23, 24, 41].



(a) Aerosonde (b) Bixler

(c) Tulpar (d) Decathlon

Figure A.1 : UAS similar to Tulpar

All the tables will be given in the same format later on, and also the link to .mat files of the tables will be provided as a GitHub repository or alternative way if the project goes open-source.

The aerodynamic sizing of Tulpar is done by ITU Team Hedef.

Table A.1 : Aerodynamic Sizing for Tulpar UAV

Wing		**Ruddervator**	
Airfoil	USA 35B	Airfoil	NACA 0012
Span [m]	2.00	Tail Arm [m]	0.96
Aspect Ratio	7.00	Aspect Ratio	4.36
Platform Area [m2]	0.57	Platform Area [m2]	0.19
MAC [m]	0.292	Tail Volume Ratio	0.60
Dihedral [deg]	0.00	Anhedral [degree]	44.00
RootChord [m]	0.36	Chord [m]	0.20
Tip Chord [m]	0.21	Wing LE- UAS CG [m]	0.13
Incidence Angle [deg]	2.00	UAS CG - Tail AC [m]	0.95
Control Surfaces			
	Flap	**Aileron**	**Ruddervator**
Span [%]	0%-60%	60%-100%	0%-100%
Chord [%]	33%	33%	30%
Deflection [deg]	30°&60°	± 45°	± 45°

Table A.2 : Performance Values for Tulpar UAV

Empty Weight [kg]	4.65	Cruise Speed [m/s]	17.00
Avionics Weight [kg]	2.72	Take-off Speed [m/s]	14.00
Payload Weight [kg]	2.15	Max Speed [m/s]	25.00
Max. Takeoff Weight [kg]	7.50	Stall Speed, Clean Conf [m/s]	11.00
Wing Loading [kg/m2]	13.16	Flapped Landing Speed [m/s]	10.80
Drag with Payload [N]	8.40	Climb Rate [m/s]	2.50
ESC	Hobbywing Platinium 100 A v3	Endurance [min]	12.00
		Operational Range [m]	9700.00
Motor	SunnySky X3525	Operational Ceiling [m]	60.00
Power Consumption [A]	22.00	Take-off Distance [m]	14.50
Propeller	13x8 Wooden	Landing Distance [m]	25.00
Batteries	6S2P - 5,200 mAh 10C Li-Po	Minimum Turn Radius @2g [m]	13.00

APPENDIX B: ArduPilot Parameters

About ArduPilot Tuning Process, the documentation states that the parameters with "*" is for advanced users.

Table B.1 : Roll control loop.

Parameter	Meaning	Symbol	Value	Range
RLL2SRV_TCONST*	Roll Time Constant	τ_ϕ	0.45	$[0.4, 1.0]$
RLL2SRV_P	Roll Proportional Gain	K_{P_ϕ}	0.7	$[0.1, 3.0]$
RLL2SRV_D	Roll Damping Gain	K_{D_ϕ}	0.01	$[0, 0.5]$
RLL2SRV_I	Roll Integrator Gain	K_{I_ϕ}	0.1	$[0, 0.2]$
RLL2SRV_RMAX_UP	Maximum roll rate	-	75	$[0, 180]$
RLL2SRV_IMAX	Roll integrator gain limit	-	30	$[0, 4500]$
RLL2SRV_FF	Feed forward Gain	K_{FF_ϕ}	30	$[0.1, 4.0]$

Table B.2 : Pitch control loop.

Parameter	Meaning	Symbol	Value	Range
PTCH2SRV_TCONST*	Pitch Time Constant	τ_θ	0.45	$[0.4, 1.0]$
PTCH2SRV_P	Pitch Proportional Gain	K_{P_θ}	1.055	$[0.1, 3.0]$
PTCH2SRV_D	Pitch Damping Gain	K_{D_θ}	0.08	$[0, 0.2]$
PTCH2SRV_I	Pitch Integrator Gain	K_{I_θ}	0.15	$[0, 0.5]$
PTCH2SRV_RMAX_UP	Pitch up max rate	-	75	$[0, 100]$
PTCH2SRV_RMAX_DN	Pitch down max rate	-	75	$[0, 100]$
PTCH2SRV_RLL	Roll compensation	-	1	[0.71.5]
PTCH2SRV_IMAX	Pitch integrator gain limit	-	30	$[0, 4500]$
PTCH2SRV_FF	Feed forward Gain	-	1	$[0.1, 4.0]$

Table B.3 : Yaw control loop.

Parameter	Meaning	Symbol	Value	Range
YAW2SRV_SLIP*	Sideslip control gain	-	0	$[0,4]$
YAW2SRV_INT*	Sideslip control integrator	-	0	$[0,2]$
YAW2SRV_DAMP*	Yaw damping	-	0	$[0,2]$
YAW2SRV_RLL*	Yaw coordination gain	$K_{r_{coord}}$	0	$[0.8,1.2]$
YAW2SRV_IMAX*	Yaw integrator gain limit	-	15	$[0,4500]$
KFF_RDDRMIX	Rudder Mix	$K_{r_{mix}}$	0.5	$[0,1]$

APPENDIX C: Flight Checklists

Table C.1 : Flight checklist - pilot.

TEAM HEDEF - FLIGHT FORM

Flight No:	Flight Date & Time:	Flight Duration:

Aircraft Conditions					
	Take-off Weight (kg)		Location of Center of Gravity from LE		
	Main Battery Type		Aux Battery Type		
Cell					Cell
mAh					mAh
	Main Battery Level		Auxiliary Battery Level		
	Before Flight	After Flight	Before Flight	After Flight	
Volt					Volt
%					%

No	Structural Controls	Check
1	Fuselage and Nose Sctructral Integrity	
2	Wing Structral Integrity	
3	Wing-Fuselage Connection	
4	Tail Structral Integrity	
5	Wing-Tail Connections & Booms	
6	Landing Gear Sctructral Integrity	
7	Control Surfaces and Servo Connections	
8	Fuselage Hatch Connection	

No	Flight Procedure	Check
1	Turn On RC Transmitter	
2	Select the Correct Profile from RC Transmitter	
3	Perform Range Test for the RC Transmitter	
4	Check RC Transmitter's Battery Level	
5	Check Aircraft's Main and Auxality Battery Level	
6	Connect Main Battery to Power Brick	
7	Connect Auxilary Battery	
8	Check Propeller Direction	
9	Check Control Surfaces' Direction & Angle Limits	
10	Do Trim Adjustments (If Required)	
11	Perform Propulsion Test For ESC Calibration	
12	Perform Taxi Test	
13	Check the Flight Area Clearance	
14	Determine Wind's Direction For Take-off	

Pilot's Flight Comments:

Table C.2 : Flight checklist - autopilot.

Flight No: **Flight Date & Time:**

AUTOPILOT CHECKLIST

Ground Control Station		Check
GCS Computer	Battery level is sufficient	
	Telemetry is connected	
Mission Planner	WPs and altitudes are accurate	
	Flight plan is saved to PC	
	Uplink/Downlink speed is accurate	
	Flight plan is uploaded to UAS	
Telemetry Module Antenna	0 and 90 degree angles are set	
Telemetry Strength	Signal is 100%	

Autopilot		Check
Pixhawk position, direction	Position and direction in plane	
GPS position, direction	Is GPS position and direction true ?	
Telemetry Signal Strength	Is sıgnal power enough ?	
Acceleration, gyro, mag calib.	Is calibration necessary	
EKF	Is the filter faulty	
RC Failsafe	Remote control failsafe behavior	
GCS Failsafe	Telemetry failsafe behavior	
Logging	Was Logging opened ?	
Artificial Horizon	Is autopilot orientation compatible ?	
Manual control surface direction	Control surfaces' direction of movement on manual mode	
FBWA control surface direction	Direction of movement surfaces on FBWA	
ALT_HOLD_RTL, WP_LOITER_RAD	Entered height, and radius values	
Max roll, pitch	Maximum orientation angles	
PID, Parameters	Are PID values or parameters correct ?	
Flight Modes	Remote control flight modes	

CURRICULUM VITAE

Name Surname: Ertuğrul Barış ÖNDEŞ

Place and Date of Birth: Bakırköy, June 1993

Adress: Istanbul Technical University Ayazaga Campus, Faculty of Aeronautics and Astronautics, No: 310, 34469, Sariyer, Istanbul, Turkey

E-Mail: ondes@itu.edu.tr **Website:** bit.ly/ondes-website

EDUCATION:

- **M.Sc.:** 2019, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Aeronautics & Astronautics Engineering
- **B.Sc.:** 2017, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Aeronautical Engineering
- **B.Sc.:** 2017, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Control and Automation Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2018-2019, Teaching & Research Assistant, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Aeronautical Engineering Department.
- 2015-2018, Team Leader, ITU Hedef UAS Team
  - Institution of Mechanical Engineers's UAS Challenge 2018

    Grand Champion Title, Safety & Airworthiness Award, Best Flight Readiness Review.
  - Institution of Mechanical Engineers's UAS Challenge 2017

    5th Place, Manufacturing Award, Best Flight Readiness Review.

PUBLICATIONS:

- Ondes, E. B., Bayezit, I., Poergye, I., Hafsi, A., "Model-Based 2-D Look-Up Table Calibration Tool Development," *IEEE Asian Control Conference (ASCC)*, Gold Coast, Australia, December 17-20, 2017.
- Bayezit, I., Ince, F. F., Ondes, E. B., "Model Based Hybrid Powertrain System Design and Testing," *Proceedings of the Automative Technology Congress*, Bursa, Turkey, May 23-24, 2016.