İSTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEVELOPMENT OF INERTIAL NAVIGATION SYSTEM WITH APPLICATIONS TO AIRBORNE COLLISION AVOIDANCE

M.Sc. THESIS

Mehmet HASANZADE

Department of Control and Automation Engineering

Control and Automation Engineering Programme

AUGUST 2016

İSTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEVELOPMENT OF INERTIAL NAVIGATION SYSTEM WITH APPLICATIONS TO AIRBORNE COLLISION AVOIDANCE

M.Sc. THESIS

Mehmet HASANZADE
(504141120)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Asst. Prof. Ali Fuat ERGENÇ

AUGUST 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ATALETSEL SEYRÜSEFER SİSTEMİ GELİŞTİRİLMESİ VE HAVA ARACI ÇARPIŞMA ÖNLEME UYGULAMALARINDA KULLANIMI

YÜKSEK LİSANS TEZİ

Mehmet HASANZADE
(504141120)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Yüksek Lisans Programı

Tez Danışmanı: Yrd. Doç. Dr. Ali Fuat ERGENÇ

Ağustos 2016

Mehmet Hasanzade, a M.Sc. student of ITU Graduate School of Science Engineering and Technology student 504141120, successfully defended the thesis entitled "DEVELOPMENT OF INERTIAL NAVIGATION SYSTEM WITH APPLICATIONS TO AIRBORNE COLLISION AVOIDANCE", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Asst. Prof. Dr. Ali Fuat ERGENÇ** ....
Istanbul Technical University

Jury Members : **Asst. Prof. Dr. Nazım Kemal ÜRE** ....
Istanbul Technical University

Asst. Prof. Dr. Türker TÜRKER ....
Yildiz Technical University

Date of Submission : 2 May 2016
Date of Defense : 26 August 2016

To my family,

FOREWORD

First of all, I would like to express my sincere appreciation to my advisor Asst. Prof. Dr. Ali Fuat ERGENÇ for his guidance, support and the opportunity.

I would like to express my gratitude to Prof. Dr. Gokhan Inalhan and Asst. Prof. Dr. Nazım Kemal Üre for their guidance, support and insight throughout the research.

Also, I would like to thank my unmanned aerial vehicle team Farabi Ahmed Tarhan and Aykut Çetin for their best support.

I would like to thank Control and Avionics Laboratory members for their full support.

Finally, I am very grateful to my parents and sister for their endless and invaluable supports during all of my education years.

August 2016

Mehmet HASANZADE
(Telecommunication Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

INS	**:** Inertial navigation system
AHRS	**:** Attitude heading reference system
UAV	**:** Unmanned aerial vehicle
UAS	**:** Unmanned Aerial System
KF	**:** Kalman filter
EKF	**:** Extended kalman filter
IMU	**:** İnertial measurement unit
GPS	**:** Global positioning system
FMS	**:** Flight management system
ATC	**:** Air traffic controller
FIDL	**:** Flight Intent Description Language
AIDL	**:** Aircraft Intent Description Language
AI	**:** Aircraft Intent
FI	**:** Flight Intent
FCC	**:** Flight Controller Computer
TCI	**:** Trajectory Computation Infrastructure
ADS-B	**:** Automatic dependence survaillance - broadcast
GUI	**:** Graphical user interface
W	**:** Art izi
TP	: Predicted trajectories
UPM	: User preferences Model
OCM	: operational context model

LIST OF TABLES

LIST OF FIGURES

Page

DEVELOPMENT OF INS WITH APPLICATIONS TO AIRBORNE COLLISION AVOIDANCE

SUMMARY

Last years, the market growth of UAV is increasing day by day. This market growth is not just for some typical applications, but also application areas are increasing, too. This demand also increases the market value of the UAV. For competition in the market, UAV companies try to develop UAVs more efficient, cost effective and adding different capabilities. However, this growth generates some dangerous situations, moreover, because of the growth in application area, common UAVs are become not enough for applications or missions.

In this thesis, I present and demostrate INS-AHRS Design and also Flight Management System with Collision Avoidance for UAV. These algorithms and demonstrations are made by the funding of ITU Control and Avionics Laboratory. In Laboratory, we already have autopilot system for multi-copter platforms and fixed-wing platforms. Before development of this INS-AHRS, we used other products from industry. But these products do not let you manage all system. But with the growth on the UAV applications, in the world also even in our laboratory, many projects required to solve specific problems with UAV. Industry products are designed for just one specific platform which may not be work on another platform. That is the main reason of necessity to develop new INS-AHRS, which can be used for multi-copter platforms.

To develop INS-AHRS, filtering techniques and other conversation equations are studied. In this study, it is decided to use one IMU and one GPS. But after encounter with different problems, external magnetometer is added to the system. Then, as datasheet recommended, scaling and also alignment and offset shifting is studied.

Before developing the all system, for inner loop, controller all need is attitude and attitude rate feed back. So first, with complimentary filter, gyroscope and accelerometer filtering is developed and tried to test at outside. In simulation, decision of coefficient of complimentary filter is easy to find. But these coefficients do not work at the outside. This shows the most important challenge that simulation platform can never be the same with outside real flight.

For INS design, inertial frame to NWU frame conversation is developed. Accelerometers gravity vector and Coriolis vector is removed. Gyroscope outputs are also converted to the NWU frame. At least, all sensor outputs become the type of navigation frame. Whenever all datas gathered are become the type of the same frame, kalman filter is designed for INS.

AS a result of INS-AHRS design, after 6 months of testing with other industrial INS, final coefficient of both INS and AHRS is decided. After few more development, test videos are recorded.

For the growth of the UAV problem, this thesis presents Flight Management System (FMS) with multi-level autonomy modes that meet the requirements of future flight operations for unmanned aerial systems (UAS). It is envisioned that the future of airspace will become highly heterogeneous and integrate non-standardized aerial systems. In that case, only ground systems will be able to predict future trajectories based on performance models (stored in huge parametric databases). Meanwhile, airborne systems are required to share information. The proposed FMS structure integrates new functionalities such as (1) formal intent and information exchange and collaboration in tactical planning utilizing air-to-air and air-to-ground data links and (2) decentralized, short-term collision detection and avoidance. The air-to-ground data link enables intent sharing and allows field operators (i.e., flight operators or air traffic controllers) to interpret, modify, or re-plan UAS flight intent. The onboard FMS persistently monitors the airspace, tracks potential collisions with the other aircraft and the terrain, and requests re-planning when it detects a possible issue. When an immediate response is needed, the onboard FMS generates a 3D evasive maneuver and executes it autonomously. Flight traffic information is obtained from ADS-B/In transponders and air-to-air data links. ADSB-In/Out implementations make the unmanned systems more visible to the systems in 3D. In addition, the air-to-air data links enable intent sharing between airborne systems and are traceable in four dimensions (i.e., space and time). The experimental FMS was deployed in quadrotor UASs and a ground station and GUI was designed to perform demonstrations and field experiments for the issues introduced in the paper.

ATALETSEL SEYRÜSEFER SİSTEMİ GELİŞTİRİLMESİ VE HAVA ARACI ÇARPIŞMA ÖNLEME UYGULAMALARINDA KULLANIMI

ÖZET

Ülkemizde ve dünyada insansız hava araçlarının kullanımı gün geçtikçe artmaktadır. Sadece insansız hava aracı kullanımı değil, kullanıldığı alanlar da artış göstermektedir. Bu da insansız hava aracı marketini daha cezbedici kılmaktadır. Bu artış sonucu dünyada bir çok insansız hava aracı şirketi kurulmuş ve bazıları bu araçları seri üretim şeklinde üreterek ihtacat yapabilmektedirler.

Dünyadaki bu ekonomik büyümenin bir yansıması olarak dünyadaki insansız hava aracının sayısı da gün geçtikçe artmaktadır. Bu talebin büyüklüğüne bakılarak, 20 yıl sonra meydana gelecek insansız hava aracı çarpışmaları ve trafikleri otoriteleri bu konu ile ilgili çalışmaya sevketmiştir.

Bununla beraber uygulama alanlarının artması ve daha da detaylanması nedeniyle belirli özellikleri ve otonom uçuşu gerçekleştirebilen insansız hava araçları artık yetersiz kalmaktadır. Günümüzde genel olarak DJI, Pixhawk, ardupilot gibi markaların araçları veya otopilotları kullanılmaktadır. Bazıları açık kaynak kodlu olsalar bile kod içerisinde değişiklik yapmak veya farklı bir donanım entegre etmek oldukça zor. Bunun haricinde piyasada baskın olup market değeri de en yüksek olan DJI firmasının ürünleri tamamiyle kapalı kutu şekilde satılmaktadır. Otonom uçui, rota takibi, havada asılı kalma ve video çekme, canlı yayın yapma gibi temel isterleri yapabilmelerine ragmen, genişleyen sektörde endüstrinin istekleri, artık insansız hava aracının sadece canlı yayın yapması için değil, harici eklenecek donanımlar ile beraber çalışabilirliği veya başka sistemlerle entegre çalışabilirliği gibi problemleri ortaya çıkarmıştır. Bu nedenle piyasada ciddi bir şekilde müşteri isteğine göre configure edilebilen otopilot sistemleri ihtiyacı doğmuştur.

Diğer yandan insansız hava aracı trafiğine bile yol açacak kadar büyüyen bu sektör ve sivil havacılığın da benzer bir şekilde büyüdüğü iki sektör ile karşı karşıyayız. Sivil havacılığın artan trafiği ve çarpışma önleme sistemlerinin yetersiz kalması gibi durumlara çözümler aranmaktadır. Yapılan çalışmalar sonucu [1] insansız hava aracı sahası ile sivil havacılık sahasının birleştirilmesi ve bu birleştirmelerin nasıl yapılması gerektiği konusu ortaya çıkmıştır. Bunun üzerine bir çok üniversite, bu konu üzerine çalışmalar yapmış ve yayınlar ortaya çıkmıştır. Genel olarak problem ise elbette eski

teknolojinin hüküm sürdüğü sivil havacılıkta kullanılan ürünlerin, insansız hava araçlarına entegrasyonu imkansızdır. Doğal olarak tüm hava araçlarının kontrolü için tek bir iletişim ağı hepsini kapsayacak şekilde kurulması amaçlanmıştır. Tüm bu hava araçlarının gözlemlenmesi aynı anda yapılabilmeli ki tehlike durumlarında gerekli müdahaleler ve tedbirler önceden veya o an alınabilsin.

Bu tezde iki farklı problemin çözümü önerilmiştir. Önerilerin ilki bahsedilen müşteri odaklı insansız hava aracının tasarlanmasıdır. İnsansız hava aracı tasarımındaki en önemli modüllerden biri de INS-AHRS sistemidir. İstanbul Teknik Üniversitesi Kontrol ve Aviyonik Laboratuvarında yapılan bu çalışma öncesinde, otopilot kontrolcü tasarımı çalışmaları yapılmış ve system oturtulmuştur. Yapılan uçuşlarda piyasadaki pahalı sistemler kullanılmaktaydı. Fakat sistemden sisteme farklılıklar göstermesi gereken bu ürünler, platform değişikliklerinde sıkıntılara yol açabiliyordu. Buna örnek vermek gerekirse sabit kanatlı insansız hava aracında sıkıntısız uçabilirken, multi-copter platformunda sapma açısında uçuş anında düzensizlikler ortaya çıkıyordu. Bunun nedeni ise alınan üründe sapma açısı sadee GPS verilerinden elde ediliyor olmasıydı. Hareketli platformun her zaman bir sapma açısı olacağından sabit kanatlı sistemlerde çalışması gayet normaldi. Fakat multi-copter platformunda havada asılı kaldığı zamanlarda sapma açısında bir hız vektörü olmadığından GPS hesaplayamıyor ve bu yüzden salınımlara neden oluyordu. Bu gibi problemlerin çözümü ve tamamiyle yerli, dışarıda çalışabilen, istenilen tüm platformlara tasarım değişiklikleriyle entegre edilebilecek bir INS-AHRS tasarımı yapılmaya çalışılmıştır.

Bu tasarım yapılırken literatürde yapılan çalışmalar referans alınmış, ve filtreleme tekniklerinden navigasyon koordinat sistemlerine kadar çalışmalar yapılmıştır. Sensor çıkışlarının gürültülerini bastırmak için alçak geçiren filtrelerden geçirildikten sonra gerekli dönüşümler yapılarak filter seviyesine kadar getirilmiştir. Filtre kısmında iki farklı filter testi yapılmıştır. Biri tamamlayıcı filter ve diğeri kalman filtresidir. Bu filtrelerin her bir INS-AHRS üzerinde testleri yapılmış ve nihai olarak AHRS'de tamamlayıcı filter, INS'de ise kalman filtresinin kullanımı kararlaştırılmıştır.

Yapılan çalışmalar İstanbul Teknik Üniversitesi Stadyumunda ve İstanbul Teknik Üniversitesi Havacılık Araştırma Merkezinde test edilmiştir. Yapılan testler 6 aydan fazla sürmesine ragmen nihai sonuca ulaşılabilmiştir. Bu süre zarfında tecrübe edilen en önemli nokta ise gerçek hayatta karşılaşılan problemler ile simulasyon ortamının farklı olmasıdır. Gerçek hayatta en küçük problemde bile aracınız yere çakılabilir ve her çakılmada 200-1000 TL zarar alabilirsiniz. Test yaptığımız süre içerisinde bizden kaynaklı olmayan, fakat üretim hatası olan pervanelerin kopması nedeni ile de kırımlar yaşanmıştır. Bu nedenle sistemin argesinin yapılması pahalıya mal olmuştur. Yapılan test sonuçlarının videoları çekilmiş ve sosyal mecralarda paylaşılmıştır.

Bir diğer problem ise insansız hava araçlarının sivil hava sahasına entegrasonudur. Bu entegrasyonun yapılması için gereken teknolojik gelişmeler ve algoritmik çalışmalar gerekmektedir. Önerilen sistemde araç bazlı ve uçuş bazlı haberleşme verileri belirlenip, hangi sistemler üzerinden bu haberleşmenin gereçekleşmesi gerektiği gösterilmiştir. Daha sonra tüm bu sistemler hem hava araçlarında, yer istasyonlarında ve hava trafik kontrolcülerinde olacağından tüm haberleşme ortak bir platform için toplanmış oldu. Bu nedenle uçuş kontrollerinin yapılması daha da kolaylaşacaktır. Bununla beraber çarpışma önleme sistemi için günümüzde kullanılan 2B system değil, zamanın da içine dahil olduğu 4B istem önerilmiştir. Bu algoritmaının adı RRT-Star olup, olasılıksal yaklaşarak çarpışmadan kaçmayı hedefler. Bu kaçışı hedeflerken de en optimal yolu bulmaya çalışır ve o yoldan rotasına devam eder. Olasılıksal yaklaşımların savunduğu argüman sonsuz sayıda örnek sayısında bulunacak yol limitte en optimal yola doğru gider. Bu nedenle olasılıksal çözüm bulma, deterministic yöntemlere göre çok daha hızlı olmaktadır. Fakat algoritmada optimale ne kadar yaklaşmak istenirse o kadar örnekleme sayısını arttırmak gerekmektedir. Bu artış daha çok araştırma yapması ve sistemin uzun zaman boyunca rota üretmesi demektir. Buradaki dengeyi iyi tutturarak hem uygun yolu bulmaya ve en uygun kısa sürede bulmayı amaçlanması istenmektedir.

Sistemin testi için donanımla benzetim çalışması gerçekleştirilmiştir. Bu tezde donanım benzetimi öncesi otopilot şeması verilmiş, buna bağlı test düzeneklerinin sistemi gösterilmiştir. Simulasyon olarak XPLANE programı kullanılmış ve programdan gelen sensor verilerine göre donanım sistemi uçurmaya çalışmıştır. Daha sonra çarpışma önleme algoritmasının entegrasyonu ile system testleri gerçekeştirilmiş ve sonuçları paylaşılmıştır.

Nihai olarak bu tez, insansız hava aracı sektöründeki günümüzde ve gelecekte meydana gelecek problemleri öngörüp bunlara çözüm bulmak amaçlanmıştır. INS-AHRS tasarımları gerçekleştirilip, gerçek ortamda dışarıda testleri gerçekleştirilmiştir. Çarpışma önleme algoritması üzerine çalışmalar yapılarak da bu sistemin entegrasyonu yapılmış ve donanımsal benzetim ile testleri gerçekleştirilmiştir.

1. INTRODUCTION

Over the last 20 years, unmanned aerial systems (UAS) for çivil and defence applications have become more operationally efficient, cost effective, and added high-end capabilities. According to industry forecasts, projections for 2010 to 2019 predict more than 20,000 UASs will be produced in the US, with a total of more than 35,000 produced worldwide [2]. The massive scale of development activities associated with UASs is perceived as bringing aviation into the realm of the third industrial revolution. The new autonomous capabilities that come into the UAS flight management systems allow ground operators to focus on higher level tasks and go beyond operate a single vehicle to manage an entire fleet of UASs.

Small unmanned aerial vehicles are now being used prominently in the various commercial applications. The main reason behind the growth of UAV drones market in the commercial sector is the relaxacation in the regulatory policies for the use of drones [3]. Though the regulatory environment is not completely eradicated, the relaxations offered are driving the UAV drones market to a certain extent. The various commercial sectors such as media and entertainment, inspection, surveying, and precision agriculture are contributing prominently to the demand of UAV drones.

With the growth of UAV and also for civil aviation, old designs of flight management systems become unable to manage all traffic or conflicts that may happen. Therefore, in [18], it is explained that UAV airspace and civil aviation airspae should be integrated each other. But there are a lot of management tools on civil aviation that small UAVs cannot even lift. So, there should be new designed low weighted devices for all vehicles that can use. But the challenge starts with the communication technology decision and communication type decision.

Nowadays, there is Air traffic controllers are responsible to handle all traffic on airspace. Traffic collision avoidance system (TCAS) is used for collision avoidance which is just has to different action such as decline and incline. But after 20 years, that growth on civil aviation and UAV are explained. Therefore, just 2D collision avoidance solutions cannot handle future situations that consist 4 vehicle conflicts or more. For solution, in this thesis proposed collision avoidance algorithm in 4D which consist of three dimention and in time dimention. Also for algorithm, RRT-Star is proposed which is probabilistic method which is very popular recently.

In market, there are a lot of UAVs such as DJI, pixhawk and ardupilot that can autonomously fly and perform a mission. And these companies especially DJI have some products that specificly response to specific application like agricultural drone which can be used for disinfestation or irrigation. On the other hand, without these specific UAVs, the common UAVs are doing almost the same missions like navigation, waypoint tracking, hover mode and survaillance. But the main problem in market is there are no company that can make spesific missions through requirements from industry. Sometimes, these common missions that mentioned are not enough to solve the problem. It might be because of the complexity of mission or not available for another hardware integration or cannot make any software changes.

For spesific example can be given in our university's requirement which is about security. In universities, if there are any safety issues appears to students or stuff members, they should easily reach to the security members. In emergency situations, security members should observe the situation before act. Therefore, usually in this situations, security send a car with two security member to the emergency call location which can be obtained with university's application in smart phones which is called ITUMOBİL. In this application, person who faced danger or any emergency situations, there is red button for emergency calls that allows phones GPS send to the security to get help immidiately. If the region where emergency call came from can be observed by cameras that located seperatly in campus, then before sending cars, situation can be observed and analysed. But in some situations, in that region, there is no survaillance cameras and also that place might be far away from security center. Therefore, the emergency call and security reach to the place might take more than 10 minutes. This amount of time is very important and could be very danger in some dangerous situations like dog attack, robbery or any other situtations. For this problem, as Control and Avionics Laboratory proposed Campus UAV Project that in this emergency situations, if security accepts, one UAV will take-off and follows path to reach that region. On the UAV, there will be dog repeller, light and survaillance camera that can boradcasts live to observe the situation and that region. Also for all these hardwares, there will be gimbal to stabilize the camera and also dog repeller and light. Mentioned autopilot systems on market cannot solve this problem, because they are closed systems. In this system, UAV need to communicate with ITUMOBIL application and also security center. Also there will be a lot of no-fly-zones. So UAV should include an algorithm that can plan path for specific location. These features cannot be added to autopilots. Therefore, there should be an UAV, that developers know exactly all control loops and also sensor sets and algorithms to develop and change system to be available for communicate or add any different path planning algorithm. This is one of the reasons that shows the need of INS and AHRS design.

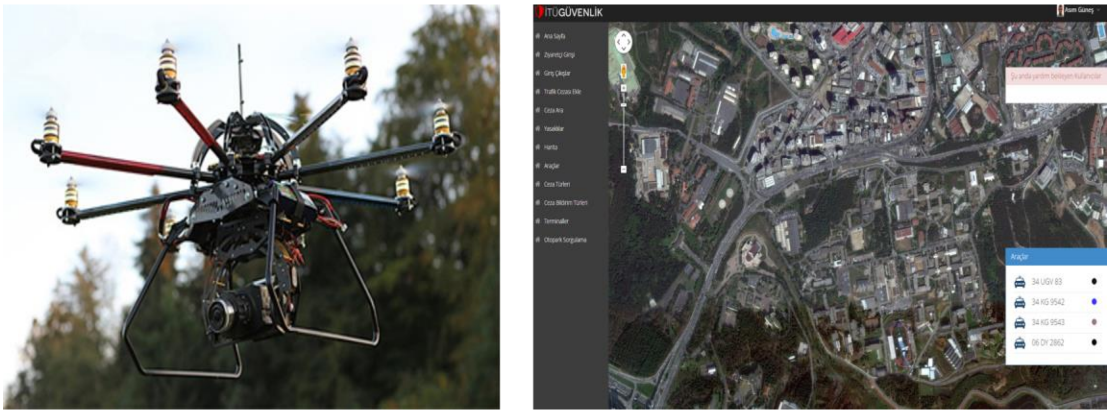

Figure 1.1: Campus UAV Project

Another example can be given which is agricultural UAV. There are multispectral cameras can be used for taking pictures with in spesific wave lenght. Taking pictures in different wave lenght allows different kind of informations to get about humidity around field, productivity level, chlorophyll ratio and nitrogen ratio. These informations can be gathered for each region on field which allows very spesific information for small ares. Before harvest, if farmer learns the problems of field, farmer can solve these problems and end of the year, field's productivity can be increased. But the problem starts when application begins. For demonstrate this mission, all field is divided to smallest regions which can be photographed from specific altitude with multispectral camera on UAV. So the requirement is, there will be waypoints that UAV whenever gets there, UAV should switch to the hover mode and take a picture of the region. And then UAV should do this for all regions to cover all field. In autopilots, there is no intervention that can be done by user to formulate this mission. Also there is no output or input on these autopilots to integrate multispectral camera and take pictures with it. This is the another reason that shows the need of INS and AHRS design.

There can be a lot of example that shows the reason of these systems development. Main idea of this thesis is design INS and AHRS that working outdoor. There is a lot of studies about this INS and AHRS design which is demonstrated in matlab simulation or hardware in the loop. But in industry, main issue is the challenges of the outside of the world. Every disturbances or errors cannot be modeled on Matlab or in other programs. Therefore, hardware and software challenges starts whenever system starts to run outside. There is a lot of disturbances like motors, wind gust, measurement errors, GPS accuracy etc. When these errors add together, it is very difficult to design system to overcome all erorrs. Moreover, UAV is not cheap vehicles. For all testbeds, UAV is customly designed and for all accidents, UAV should be repaired or new UAV should be made. So, another challenge of designing INS and AHRS for UAV is being very expensive.

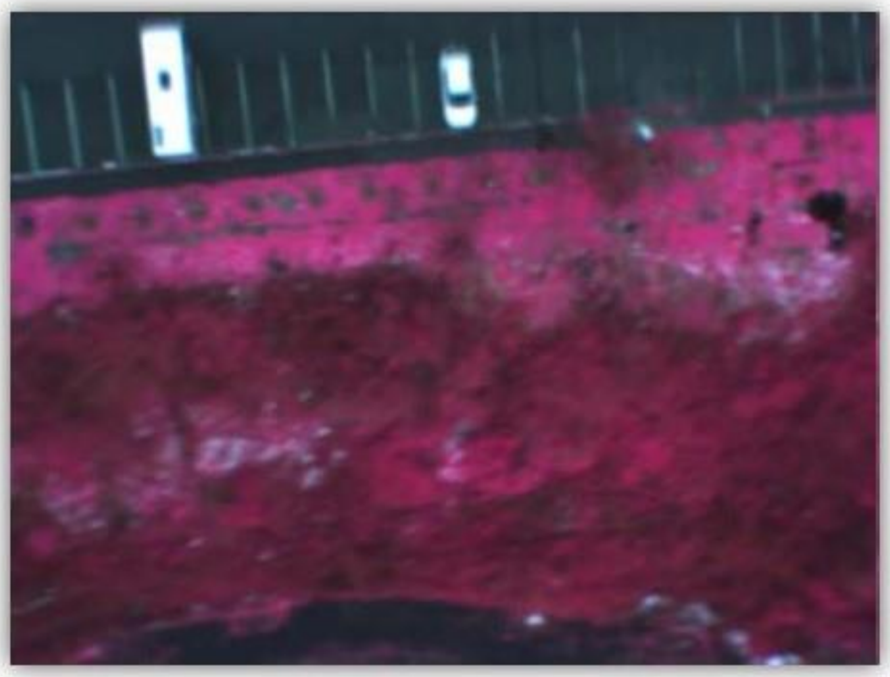

Figure 1.2: Photograph from Istanbul Technical University with multispectral camera

2. INERTIAL NAVIGATION SYSTEM (INS) AND ATTITUDE – HEADING REFERENCE SYSTEM DESIGN

2.1 Introduction

This part of thesis explains about how to develop inertial navigation systems (INS) algorithms. This algorithm is designed for multi-copter type unmanned aerial vehicles (UAV). In literature, INS designs are more likely based on fixed-wing UAV's. But because of the difference of kinematic behavior, the filter coefficients and also filtering methods changes. For example, GPS heading value is very accurate for fixed-wing UAV's, because UAV is always have a motion in certain direction. But instead of fixed-wing UAV's, multi-copter type UAV's can hover in one position. In this state, UAV has its own yaw angle, but do not has any movement on its own heading direction. That's why GPS heading calculation is randomly changes. For this problem, different types of UAV's cannot always use the same INS. That's why this represented algorithm can only be used for multi-copter type UAV's.

2.2 Sensor Errors and Considerations

The main idea of INS is control loops of controller for UAV needs usable data from sensors. But sensor measurement has a lot of errors and noises which makes control of UAV almost impossible. These errors and considerations are listed below [4,5,6]:

- Stability: The ability of the sensor to deliver the same output, over time, for the same constant input.
- Repeatability: The ability of the sensor to deliver the same output for the same repeated input, assuming all other conditions are the same (see Turn-on to Turn-on Bias).
- Drift: The change of the output over time (zero drift is the change over time with no input)

- Input Range: The input range is the maximum angular rate or acceleration the IMU can meaningfully measure.
- Bias: The difference between the real value and the output is the bias.
- Bias Stability (In-run Bias): This change in bias is often related to temperature, time and/or mechanical stress on the system.
- Bias Repeatability (Turn-on to Turn-on Bias): The initial bias is different. This is due to a number of effects, including change in the physical properties of the IMU.
- Scale Factor: Scale factor error is the relation between input and output.
- Scale Factor Linearity: Scale factor effects are most apparent in times of high acceleration and rotation.
- Resolution: The resolution of a sensor is the smallest change it can detect in the quantity that it is measuring.
- Accuracy: The accuracy of the sensor is the maximum difference that will exist between the actual value (which must be measured by a primary or good secondary standard) and the indicated value at the output of the sensor.
- Random Walk (Sensor Noise): If a sensor measures a constant signal, a random noise (error) in the measurement is always present. This is described as a stochastic process and is minimized using statistical techniques.
- Sensor Non-orthogonality (Misalignment): The three gyroscopes and three accelerometers are mounted orthogonal to each other. The mountings, however, have errors and so are not perfectly 90 degrees. This leads to a correlation between sensors.
- G Dependency (Acceleration Effect): Some gyroscopes and accelerometers are subject to a change in the bias depending on how the sensor experiences acceleration.
- Timing Errors (Latency): The difference between the time the IMU measures motion and the time that external sources like GNSS measure the same motion, is a very important factor in the quality of the resulting, combined solution.

2.3 Hardwares

Basically, this INS design consist of two hardwares which are global positioning system (GPS) and inertial measurement unit (IMU). For GPS, U-blox NEO 6M product is used and for IMU, Analog Devices ADIS16407 10 DOF is used. IMU includes accelerometer, magnetometer, gyroscope and barometer sensors. As a result, these two hardwares is enough to design INS. But in practice, these two is not enough, because of magnetic interferences.

Figure 2.1: Multi-copter platform UAV

As it can be seen from Figure 2.1: Multi-copter platform UAV, in multi-copter platforms, motor mounts and body frame which is used for place all electronic components such as controller, receiver and INS is in the same height. When motors start working, motors propagate a magnetic field which have an effect on electronic devices. Actually, not all devices are effected, but as a magnetic interference, magnetometer sensor measurement become unstable and also not reliable. That is a main problem about magnetometer sensors because not just magnetic interference from motors but also any metal stuff such as street light poles and cars are effecting the measurement very significantly. For this problem, there are some solutions. First one is using magnetic absorbent material like mu metal sheet. But the problem for this solution is, when this sheet is used, if it is not properly placed around magnetometer, it can absorb the earth magnetic field. Therefore, magnetometer can not measure

UAV's heading. Second solution which is used in this INS design is place magnetometer in higher place from motors.

In this INS design, it is mentioned that for IMU Analog Devices ADIS16407 is used which is all IMU sensors are inside this IMU module. For accelerometer and gyroscope, these sensors should be placed at the center of gravity for measure accurate accelerations and angular velocities. But this rule conflicts with magnetic interference problem from motors for magnetometer. Therefore, magnetometer inside of IMU is canceled and another magnetometer which is called HMC5883L is used.

Also, vibration of multi-copter is very important because it effects directly to the IMU. Because of vibration of motors, without any damping, usable value cannot be obtained from IMU. Therefore, with 3d printing technique, IMU cradle is designed. There are a lot of damping techniques like using foem based material or jelly material to absorb the vibrations. After some tests, jelly type materials seems proper for that weight of IMU.

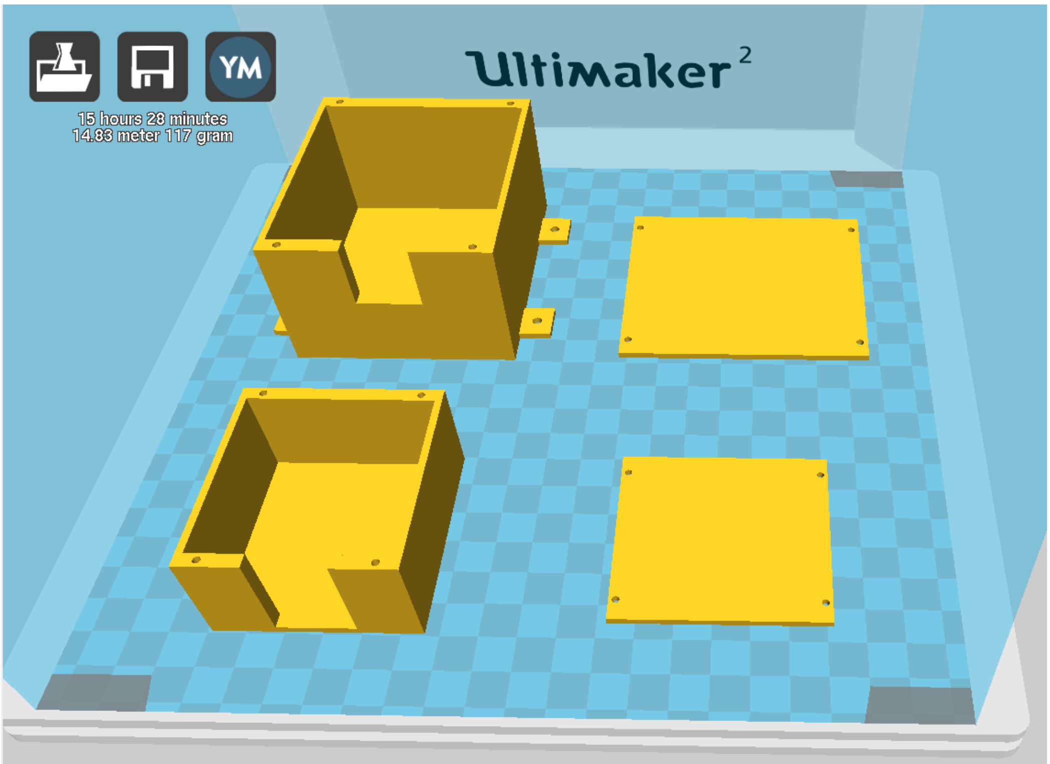

Figure 2.2: IMU cradle design

These sensors features are listed below:

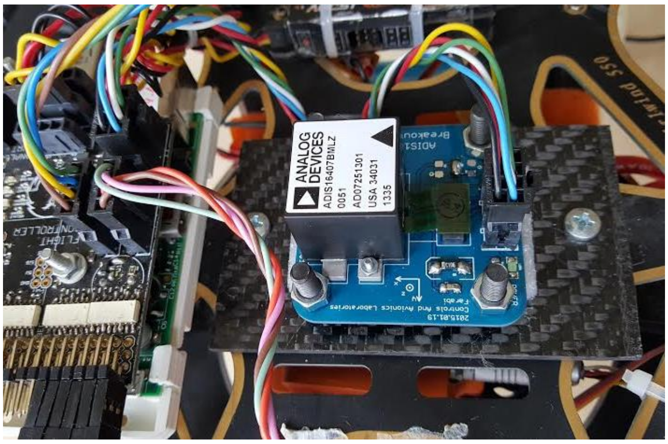

Figure 2.3: ADIS16407 10 DOF IMU

IMU: Analog Device ADIS16407

- Factory calibration,
- Three axis Accelerometer,
- Three axis Gyroscope,
- There axis Magnetometer,
- Barometer
- Temperature sensor
- 800 Hz sensor measurement update
- From SPI, 1 Mbps communication speed

Figure 2.4: GPS NEO 6M

GPS: U-blox Neo-6M Module

- 5 Hz position and velocity measurement

- 50 channel position motor
- 230 Kbps communication speed
- Signal receiver precision: -162 dbm
- 8x18x2 mm ceramic patch antenna

Figure 2.5: HMC5883L and GPS

Magnetometer: Honeywell HMC5883L

- Three axis Magnetic Field sensor
- 160 Hz measurement update
- I2C communication protocol
- 12-bit ADC
- 2 mili gauss resolution
- 1°-2° Heading Accuracy

2.4 Sensors

2.4.1 Accelerometer

Accelerometer contains a small plate attached to torsion levers. In acceleration, this plate rotates and changes the capacitance. This capacitance change is measured and acceleration value is found. The output of the accelerometers is given by:

$$y_{acc} = k_{acc}A + \beta_{acc} + \eta_{acc} \tag{2.1}$$

In this equation, y_{acc} is in Volt as output of measurement, k_{acc} is a gain, A is accelerometer in meters per second square, β_{acc} is bias term and η_{acc} is zero mean white noise [4].

In calculation, k_{acc} is given in datasheet of module and β_{acc} should be calculated before flight. Actually, this β_{acc} is significantly depend on temperature. Because of that, if the sensor is not factory calibrated, measurements always includes some bias term negatively or positively. Because before flight, β_{acc} term can be calculated, but when UAV start to flight operation, in air, temperature of enviroment can change which is also effects accelerometer β_{acc} itself. Therefore, this is the main idea for using factory calibrated IMU.

The equation given above is basic accelerometer measurements mathmatical expression. But in detail, because of gravity vector, the equation will be [7]:

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \frac{dv}{dt_b} + \omega_{b/i} \times v - R_v^b \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (2.2)$$

So;

$$a_x = \dot{u} + qw - rv + g \sin\theta + \eta_{accel,x} \quad (2.3)$$

$$a_y = \dot{v} + ru - pw - g \cos\theta \sin\Phi + \eta_{accel,y} \quad (2.4)$$

$$a_z = \dot{w} + pv - qu - g \cos\theta \cos\Phi + \eta_{accel,z} \quad (2.5)$$

As it can be seen that each accelerometer measures linear acceleration, Coriolis acceleration and gravitational acceleration.

These three axis acceleration can be measured, if sensor axis and UAV axis are aligned.

As a result of measurement acceleration, before using this acceleration values, gravitational acceleration and Coriolis acceleration should be removed from equation and in result, the actual acceleration values can be found. Removing these accelerations is discussed in next section.

2.4.2 Gyroscope

Gyroscope measures angular velocities in three axiss. These rate gyro contains a small vibrationg lever. When lever undergoes an angular rotation, Coriolis effects change the frequency of the vibration, so with this frequency change, angular rotation can be detected.

The output of the rate gyro is given by:

$$y_{gyro} = k_{gyro}A + \beta_{gyro} + \eta_{gyro} \tag{2.6}$$

In this equation, y_{gyro} correspondes to the measurement of rate of rotation in Volts, k_{gyro} is a gain, β_{gyro} is a bias term, and η_{gyro} is zero mean white noise.

In calculation, k_{gyro} is given in datasheet of module and β_{gyro} should be calculated before flight. As it is explained about accelerometer, this bias term also significantly depends on temperature.

These three axis rate gyro can be measured, if sensor axis and UAV axis are aligned.

2.4.3 Magnetometer

Magnetometer is used as a compass which is measures earth's magnetic field and finds heading. Earth's magnetic field vectors behaves similarly to that of a common magnetic dipole with the magnetic field running normal to the earth's poles and vectors are parallel at equator. Magnetometer measures the direction of magnetic field locally and provides indication of heading. Measured heading relative to magnetic North. At magnetic North it is 0° and it positively changes during clock wise heading change from magnetic North.

Mathematically, the measurement of magnetometer can be shown as,

$$y_{mag} = \psi + \beta_{mag} + \eta_{mag} \tag{2.7}$$

Where η_{mag} is zero mean gaussian nouse and β_{mag} is bias term. In magnetometer measurement, the difference from accelerometer and gyroscope, it should be calibrated hard and soft iron effects which is directly has an effect on sensor. These calibrations are actually made for ADIS16407 because of being factory calibrated. But as mentioned before, in this INS design, HMC5882L magnetometer is used which is not factory calibrated. Therefore, these calibrations should be done. These calibrations is discussed in next sections.

2.4.4 Barometer

Barometer measures atmospheric pressure. As it is known that the greater the altitude, the lower the pressure. It is also called barometric altimeter. So, the equation of pressure change which is based on equation of hydrostatics is:

$$P - P_{ground} = -\rho g(h - h_{ground}) \tag{2.8}$$

In this equation, h is the absolute altitude of the aircraft, h_{ground} is the absolute altitude of the ground. As it can be seen from equation that the absolute sea level ground pressure and also height of sea level ground should be known. But in UAV applications, this value can not be known because there is not such sensor that can measure these values. Therefore, absolute right altitude values can not be found. But still barometer is needed in this INS design, because of lack of high frequency data for altitude.

Mathematicaly, basic barometer output equation is shown as,

$$y_{abs\,pres} = (P_{ground} - P) + \beta_{abs\,pres} + \eta_{abs\,pres} \tag{2.9}$$

Where $\beta_{abs\,pres}$ is a tempreture related bias drift and $\eta_{abs\,pres}$ is zero mean gaussian noise. As it is explaned, from output of barometer, there is still unknown value which is pressure of ground P_{ground}. But this is not going to be problem for this INS design.

2.4.5 Global positioning system (GPS)

Gps is a satallite-based navigation system which provides 3-D position and velocity information. Gps system is the constellation of 24 satallites that continuously orbit the earth at an altitude of 20,180 km [8]. The configuration of satallite orbits are specificly designed, so at any point on earth's surface is observable by at least four satallites at all times. At least with this four satallites, measurement of the times of flight of signals from these satallites can determine three dimention position information. The time of flight of the radio wave signal is used to determine the range from each satallite to the receiver. Because of synchronization error exist between the satallite clocks and receiver clock, the range estimate determined from the time of flight measurement is called pseudorange to distinguish it from the true range.

Because of syncronization errors between satallites and the receiver, acurracy of this position datas is not precise. To bypass this problem, inside of GPS, there are some models of behaviours depends on real behaviours such as pedastrian, stationary, air borne etc.

For specific accuracy problem reasons are [9, 10]:

- Ephemeris Data:

  The satallite epheremis is a mathematical description of its orbit. Epheremis errors in the psuedorange calculation depends on uncertainity of the transmitted location of the satallite. This error range can be maximum 5 meters.

- Satallite Clock:

GPS satallites use cesium and rubidium atomic clocks, which drifts about 10 ns in one day. This causes an error about 3.5 meters. For solution of this problem, Satallite clocks are updated every 12 hours. So in this time interval, position error can be 1 to 2 m on avarage.

- Ionosphere:

The ionosphere is the uppermost layet of earth's atmosphere which includes free electrones that delays the transmission of GPS signals. Therefore, this delay couse positioning errors between 2 to 5 m.

- Troposphere:

The troposphere is the lowest layer of the earth's atmosphere which is almost consist all weather activities such as temperature, pressure and humidity changes. These changes effect the speed of light that couses about 1 m position error.

- Multipath Reception:

This error is happening when gps receives reflected signals that mask the true signal. This error is cause 1 m maximum position error.

- Receiver Measurement:

Latency of measurement can cause less than 0.5 m errors.

These errors are explained for understanding about GPS, position and velocity calculations and problems.

As a INS designer, there are two main problems. First problem is there is not measurement unit or any other sensor to measure and calculate these all errors. So these errors should be eleminated. Second problem is the data rate of GPS. In all common GPS modules, data rate is 5 Hz. Because of this INS is used for UAV which is running control algorithm for navigation, Control algorithm is used in 100 Hz. This means that control algorithm also needs 100 Hz precise sensor values that can be used. Therefore, these two main problems are solved by using some state estimation and dynamic filtering methods which is directly main part of INS algorithm.

It is mentioned that for INS design, U-blox Neo 6-M is used for GPS. It also has 5 Hz data rate. From defined errors, it may sound like when these errors add together, the uncertainity of GPS datas will be very unprecise. Because of that, minimum 8 satallites that GPS is actively communicate situation is suggested for user. With minimum 8 satallites, the uncertainity of GPS data is around 1.8 m to 1 m. This result can be seen from GPS itself. Because GPS data contains its own uncertainity data that user can see all uncertainity datas from GPS.

2.5 First Phase of INS

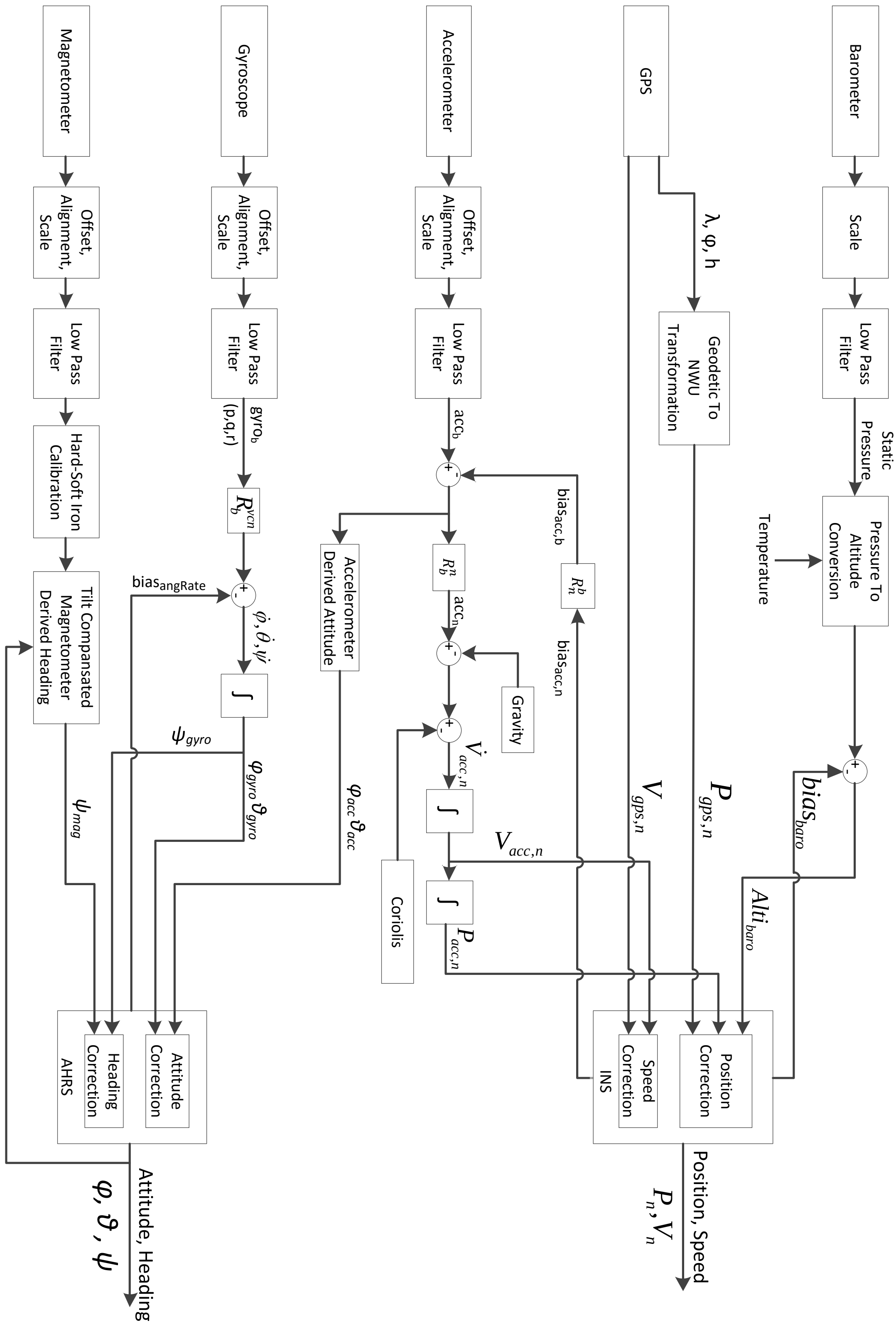

Figure 2.6: INS System

As it can be seen from Figure 2.6: INS System INS design algorithm is represented. Before this section, output of the used sensors are mantioned which means that barometer, GPS, accelerometer, gyroscope and magnetometer blocks are explained.

2.5.1 Offset, Alignment, Scale

This blocks input are outputs of sensors which are barometer, accelerometer, gyroscope and magnetometer.

If the output signal is not zero when the measured property is zero, the sensor has an offset or bias. This is defined as the output of the sensor at zero input. But for different sensors, the measured zero point tests are different.

Sensor offset characteristic is not the same as noises. This offset can be thinkable as an output which has a positive or negative value at zero point. That's why for practice, there is common algorithm for solving this issue, but some conditions should be verified.

For accelerometer example, let's say there is a offset value at zero point. For idle position which is at landing position of multi-copter platform, because of this offset value, controller of multi-copter will think this situation as there is some acceleration at that offset axis. For further blocks, there will be integrals that gives velocity and position estimations from accelerometer. And also because of this offset, controller will think there is acceleration and also some value velocity and drift from current position. This is the main issue about offset and this problem occur for each sensor as the same.

For accelerometer, there is ofcourse some offset but there is also gravity vector which is seems like offset but it is not. That's why gravity vector should be considered for another part of block definitions.

For offset calibration, multi-copter platform is placed stable position. There should be no vibration on the multi-copter, which will cause wrong offset calibration. Than for that idle position, all sensor outputs are gathered 1000 times. And than, for each sensor outputs, the avarage value of outputs are calculated and this avarage values are removed from all sensor outputs.

For scale block, at datasheet of sensors, there is some scale value for all output of sensors for make that sensor value usable. That's why for all different sensor, this value will be different. This should be checked from sensor's datasheet.

For barometer, there is no offset or alignment algorithm. That's because the output of barometer is direcly related to preasure of ground P_{ground} which is explained before sensors definition section. Also barometer sensor output is used very differently from other sensors.

2.5.2 Low pass filter

A low pass filter allows signals below a cutoff frequency and attenuates signals above the cutoff frequency. By canceling some frequencies, this filter makes sensor output much more smooth. The filter produces slow changes in output values to make it easier to see trends and boost the overall signal-to-noise ratio with minimal signal degradation.

Low pass filter algorithm can be shown as [7],

$$y[n+1] = \alpha_{LPF} y[n] + (1 - \alpha_{LPF}) u[n] \tag{2.10}$$

which α_{LPF} is represented as low pass filter coefficient which is directly depend on sensor output characteristics. $y[n]$ is the state value of that time and $u[n]$ is the sensor output of that time.

α_{LPF} value decision is very significant because of having smooth data input for system. But also in practise, if α_{LPF} is chosen very close to 1, then next state depend too much to the current state and this situation may cause disadvantages for more agile situations. If this α_{LPF} is chosen very high, if multi-copter system accelerate very fast to some way, in filtering, next state still depend on current state too much. And as a result of this situation, this high α_{LPF} low pass filter cause lag and controller miss the exact sensor output. For this situations, system cannot be stable, therefore, INS system output values and also control outputs will decieve platform to unstable situations. On the other hand, different problem may occur if α_{LPF} value is chosen very close to 0 which means that next state is directly depend on sensor output. So with this α_{LPF}, there is no meaning to use low pass filter. Because of that, α_{LPF} should be chosen carefully. As an advice, before flight or real test, Matlab test should be done with real time data and this lag size and cutoff frequency should be considered.

2.5.3 Pressure to Altitude Conversation for Barometer

As it can be seen from Figure 2.6: INS System, for altitude value, barometer sensor output is first converted to static pressure through scale and low pass filter blocks. Than before state estimation, INS needs altitude value from barometer. This pressure to altitude convertasion block is needed for this reason.

The calculation of this conversation is shown as [11],

$$P = P_b \cdot \left[\frac{T_b}{T_b + L_b.(h - h_b)}\right]^{\frac{g_0.M}{R^*.T_b}} \quad (2.11)$$

Where P_b is static pressure, T_b is standard temperature in Kelvin, L_b is standard temperature lapse rate (K/m) in ISA, h is height above sea level (m), h_b is height at bottom of layer b (meters), R^* is universal gas constant in N.m/(mol.K) , g_0 is gravitational acceleration which is changes for every different location, but for general 9.80665 m/s^2 can be used and M is molar mass of Earth's air which is 0.0289644 kg/mol.

As this issue is mentioned before, P_{ground} cannot be known before flight, because there is no such sensor or information that can be gathered. Therefore, exact altitude value cannot be known. On the other hand, because of this IMU, barometer outputs can be gathered in high frequency. Without barometer, just with GPS altitude value, altitude control cannot work properly because of 5 Hz frequency data output from GPS. But with barometer, not exact altitude value but altitude change dynamics can be estimated. So for further filters, exact value will be corrected by GPS, but prediction in high frequency will be done by barometer.

2.5.4 Geodetic to NWU transformation

The World Geodetic System is a standard use for GPS. It comprises a standard coordinate system for the Earth which uses standard spheroidal reference surface. NWU (North West Up) is also called NWU (North Wesr Up) is a geographical coordinate system for representing state vectors used in aviation. It includes three vectors which are northern axis, eastern axis and vertical position [12].

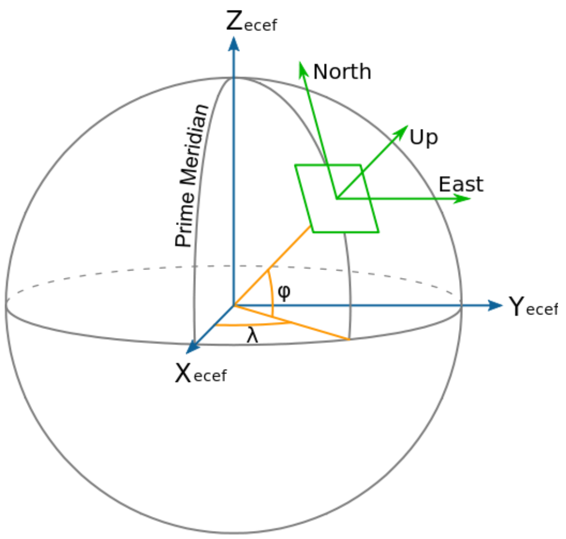

Figure 2.7: NWU and Geodetic Coordinate System [12]

With this transformation, the elliptical surface effects is removed. Also for controller, error values always in meter unit. Therefore, distances between UAV and waypoint should be converted to NWU coordinates to find values in meter.

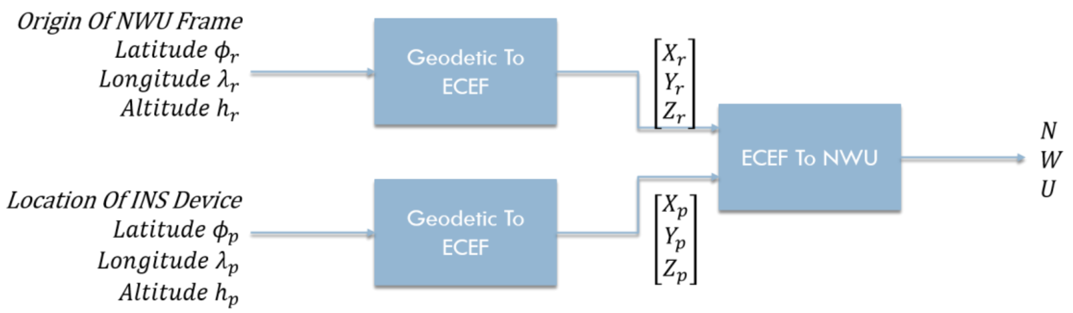

Figure 2.8: Geodetic to NWU transformation Scheme

First of all, geodetic coordinate system should be converted to ECEF coordinate which is earth centered, earth fixed coordinate system represents positions as an X, Y and Z. (0,0,0) is defined as the center of mass of the earth. Conversation equations are [12];

$$X = (N(\Phi) + h)\cos\Phi\cos\lambda \tag{2.12}$$

$$Y = (N(\Phi) + h)\cos\Phi\sin\lambda \tag{2.13}$$

$$Z = (N(\Phi)(1 - e^2) + h)\sin\Phi \tag{2.14}$$

$$N(\Phi) = a/\sqrt{(1 - e^2\sin\Phi\sin\Phi)} \tag{2.15}$$

For these equations, Φ is latitude, λ is longitude, h is altitude and a is radius of Earth. Secondly, after these conversations are calculated, location of INS device and origin of NWU frame is obtained. So, for now, location of INS device should be calculated raleted to NWU frame origin. So ECEF to NWU transformation calculations are;

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\Phi\cos\lambda & -\sin\Phi\sin\lambda & \cos\Phi \\ \cos\Phi\cos\lambda & \cos\Phi\sin\lambda & \sin\Phi \end{bmatrix} \begin{bmatrix} X_p - X_r \\ Y_p - Y_r \\ Z_p - Z_r \end{bmatrix} \tag{2.16}$$

This calculation can be done for both position and velocity values which are gathered from GPS. With this calculations, the output values are directly used for correction of position and speed state estimation filters.

2.5.5 Magnetometer hard and soft iron calibration

HMC5883L magnetometer module is used in this INS and this kind of cheap magnetometers cannot be used without calibration. These magnetometers are not factory calibrated, so these modules are subjected to distortion which can be categorized in two different topic such as hard iron distortion and soft iron distortion.

The hard iron distortion refers to the presence of magnetic fields around the sensor such as magnets and power supply wires and this distortion causes measurement offset errors. The soft iron error refers to the presence of ferromagnetic materials around the sensor, which skew the density of the Earth's magnetic field locally and this distortion causes scaling offset errors.

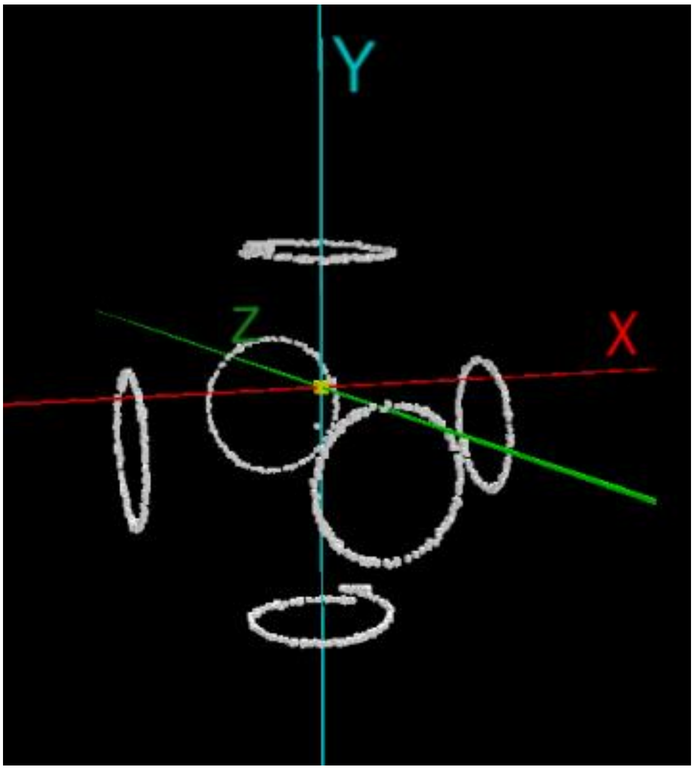

Figure 2.9: Offset errors for magnetometer output values around each axis

As it can be seen from Figure 2.9: Offset errors for magnetometer output values around each axis, the offset difference between the center of circles and axis shows hard iron distortion and the each circles similarity to ellipsoid shows the soft iron distortion. Each circle shoud be a circle and the center of each circle should be on its own axis.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} . \left(\begin{bmatrix} X_{nc} \\ Y_{nc} \\ Z_{nc} \end{bmatrix} - \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \right) \tag{2.17}$$

In this equation, X_c, Y_c, Z_c are calibrated magnetometer data, M matrix is transformation matrix, X_{nc}, Y_{nc}, Z_{nc} are non calibrated magnetometer data, B_x, B_y, B_z are bias. To find this transformation matrix and also bias values, ellipsoid fit algorithm can be used.

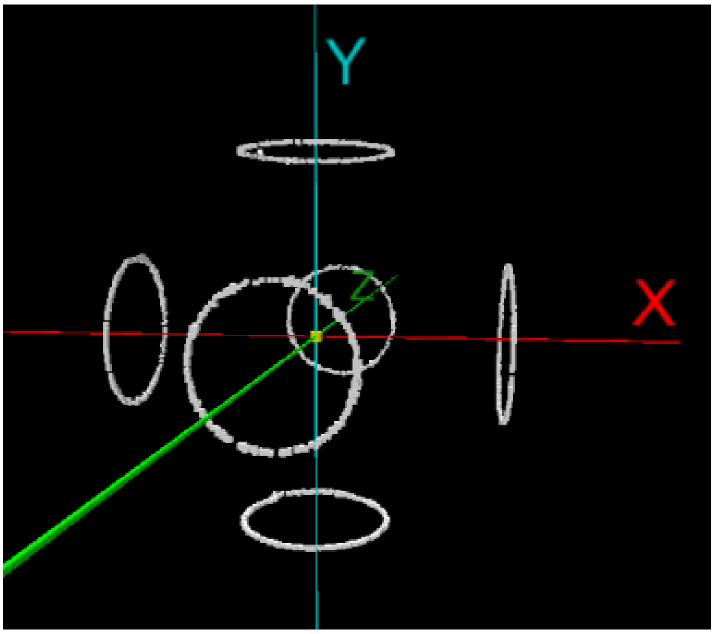

Figure 2.10: After calibration magnetometer output view

After calibration, these mentioned hard and soft iron distortions effects should be disappeared.

Now, magnetometer outputs for all three axiss should be correct. Therefore, heading of multi-copter can be calculated. But there is another challenge to calculate heading value which is multi-copter in tilt action.

2.5.6 Tilt-compansated magnetometer derived heading

After hard and soft iron calibration, heading value can be found. Heading value is the angle relative to the North that shows the direction of multi-copter's front axis. Basicly, when the device is at leveled position, pitch and roll angles are 0°, heading angle can be determined by,

$$\psi = tan^{-1}\left[\frac{y_{mag,y}}{y_{mag,x}}\right] \tag{2.18}$$

Which is not tilt compensated. If the multi-copter is tilted, then the pitch and roll angles are not equal to 0°. So pitch and roll should be measured by accelerometer. Therefore, the magnetometer sensor measurements need to be compensated to obtain tilt compansated $y_{magcalib,y}$ and $y_{magcalib,x}$ angles. So, tilt compensated angles can be determined by [13,14],

$$y_{magcalib,x} = y_{mag,x} \cos\varphi + y_{mag,z} \sin\varphi \quad (2.19)$$

$$y_{magcalib,y} = y_{mag,x} \sin\vartheta \sin\varphi + y_{mag,y} \cos\vartheta - y_{mag,z} \sin\vartheta \cos\varphi \quad (2.20)$$

Which φ is pitch angle, ϑ is roll angle. After tilt compensated values are found, the same equation that used for heading angle calculation is used.

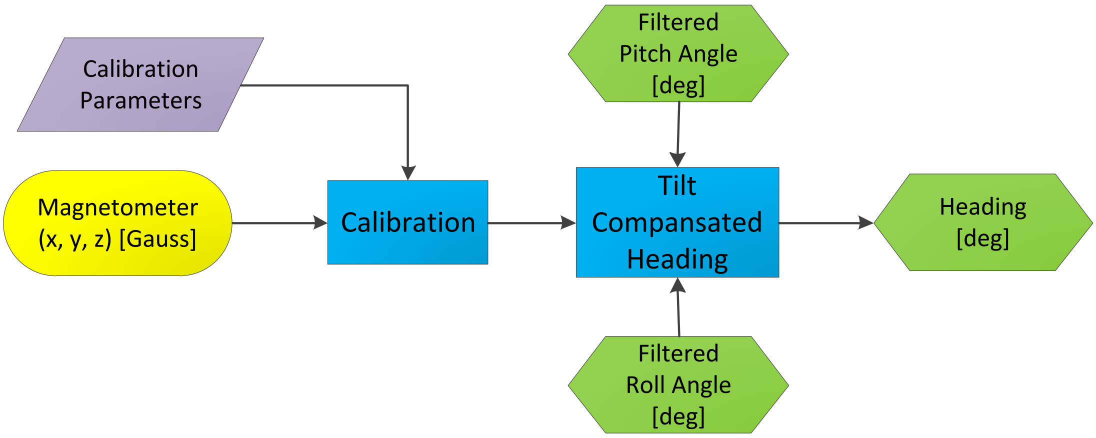

Figure 2.11: Magnetometer Calibration Schematics

After these calculations are done, this heading value is used for AHRS (altitude and heading reference system). This heading angle value will be used for correction phase of state estimation filter.

2.5.7 GPS Mode decision

There are some smoothing techniques for GPS outputs but more important thing is, there are some dynamic models inside GPS which have a very significant effect on velocity and position informations.

Modes inside GPS are “stationary”, “portable”, “pedestrian”, “automotive”, “sea”, “airborne <1g”, “airborne <2g”, and “airborne <4g”. After GPS obtain information, these models are used for minimize errors. For multi-copter platform, stationary mode might be usefull for position hold mode. On the other hand, automative, pedestrian and sea modes do not exactly similar to multi-copter platform. Therefore, stationary, portable and airborne modes are tested for altitude and also velocity for all 3 axes. Before using new GPS, decided mode should be configurated.

The test was made at faculty roof which is shown by orange color. One person walks through that orange path.

Figure 2.12: GPS Test area

The same test was applied to all modes of GPS which is shown in Figure 2.13, Figure 2.14, Figure 2.15 and Figure 2.16.

The velocity comparisons which is shown in Figure 2.14 and Figure 2.16 will be the same with other axis. Therefore, only Velocity North graphic is shown. Another important thing is the accuracy when muli-copter is in hover mode. Ofcourse these graphics are also significant for decision of which mode should be used, but in auto-pilot design, hover mode is most challenging one to design. So, for decision of this mode, no motion situation should be considered, too. Graphics for altitude change and Velocity North change is also shown below.

For comparison, there is two different situations which is in motion and in no motion. When these are compared, it is important to think that which one of the graphics are most suitable for multi-copter platform. Because these models are design for other different dynamics, so these are should be considered.

For altitude comparison in no motion in Figure 2.15, all graphics are look like the same. So, there is no enough information to choose. But for in Velocity North comparison, stationary mode is tried to be stay in constant velocity. Even if there is big change in velocity and even there is no velocity in north for that situation, it changes its value for velocity north little bit and try to stay in that value. This situation is not suitable enough for multi-copter.

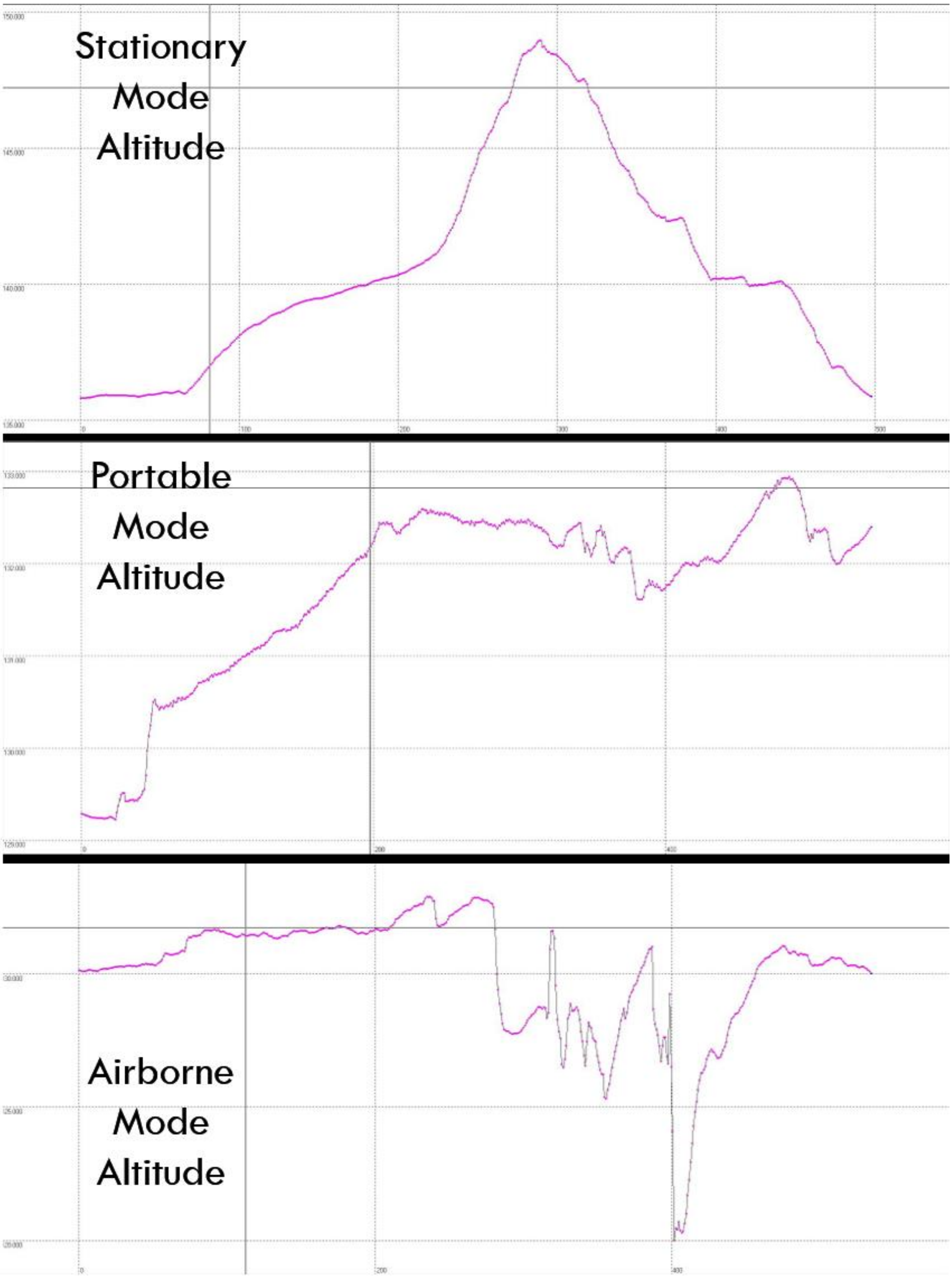

Figure 2.13: Altitude comparison in motion

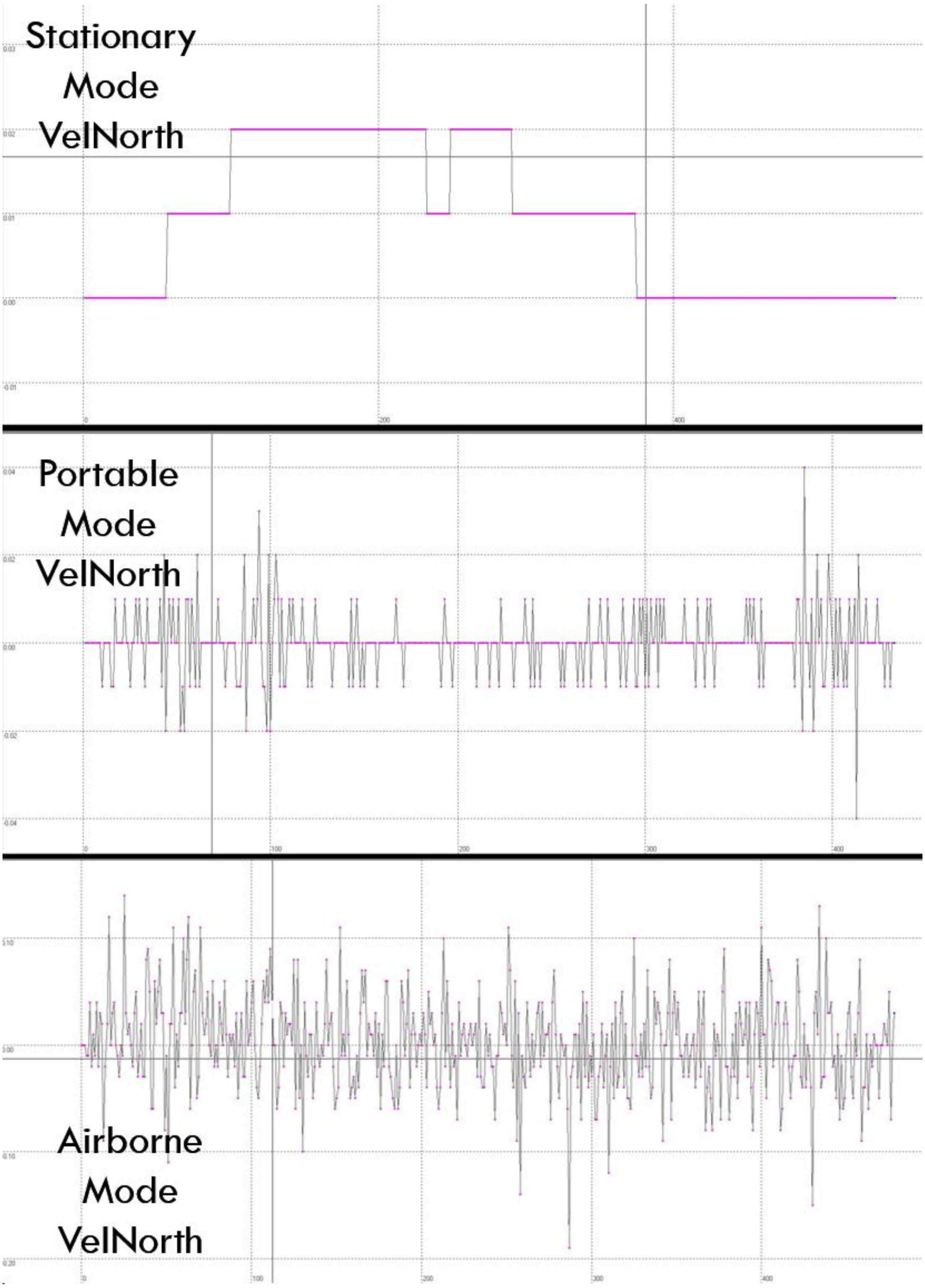

Figure 2.14: Velocity North changes in motion

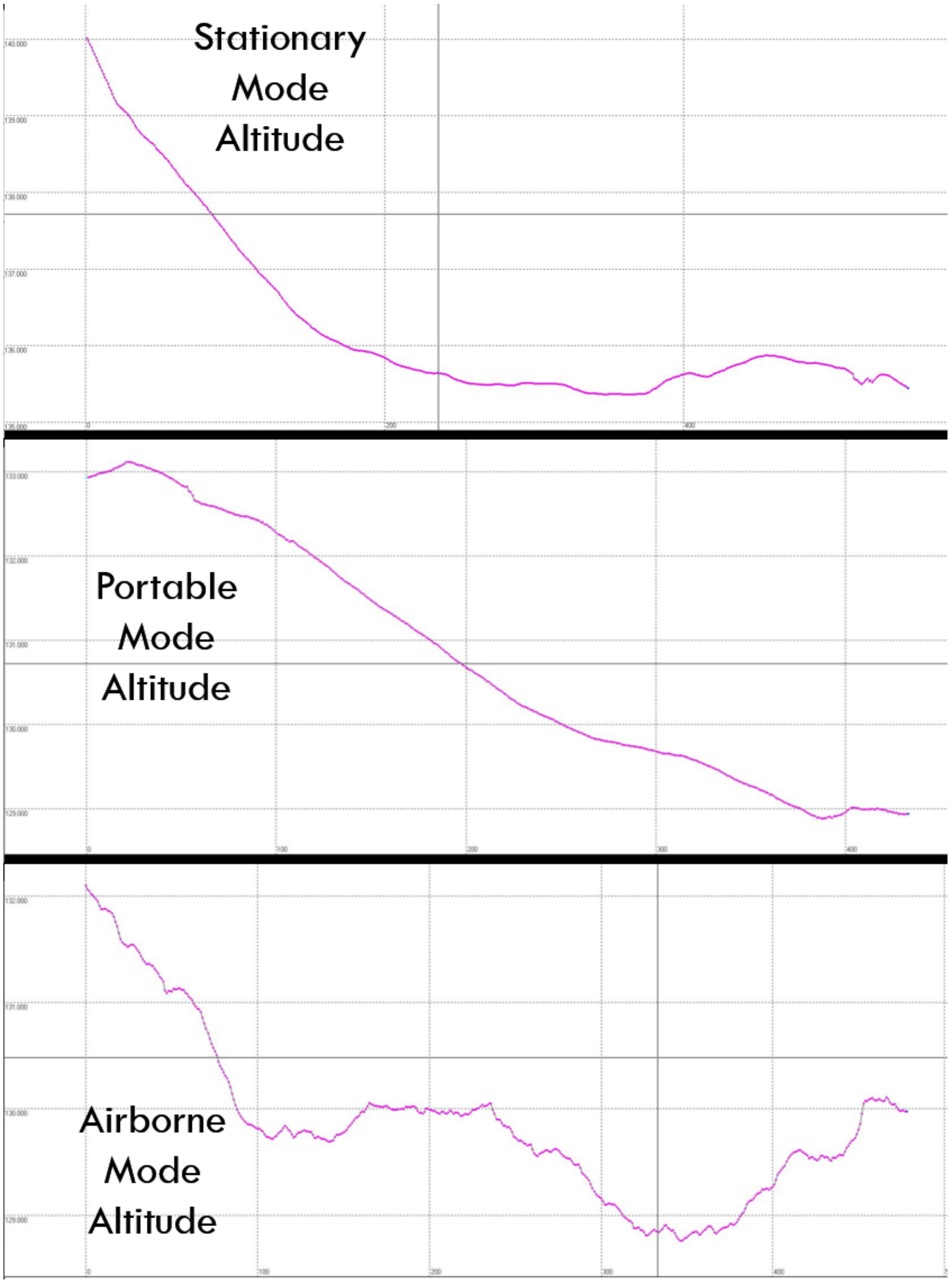

Figure 2.15: Altitude comparison in no motion

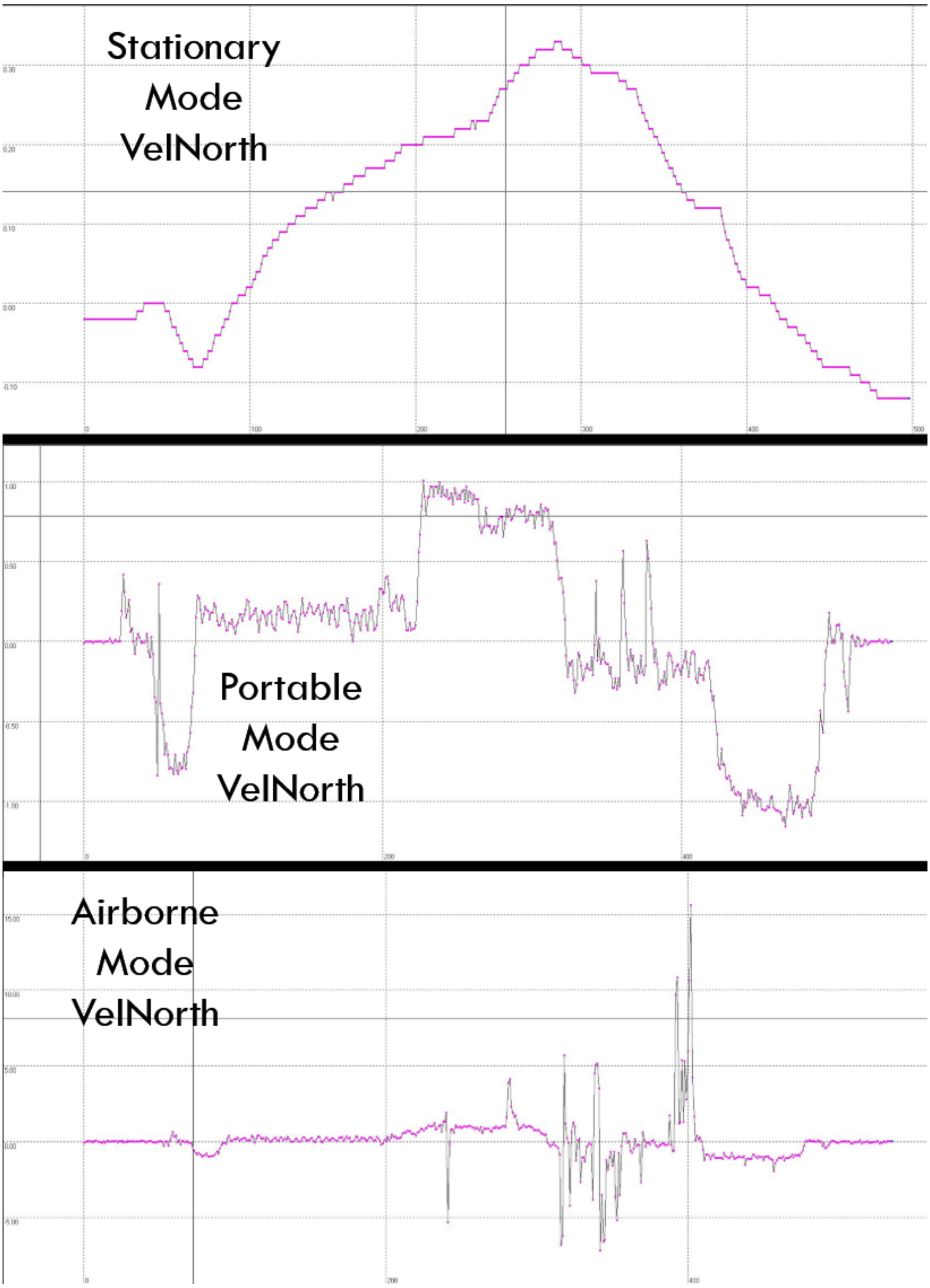

Figure 2.16: Velocity North comparison in no motion

For comparison of other two modes which is portable and airborne, it is also can be seen from velocity north comparison in no motion. Not like stationary mode, but close to it that try to stay constant on that value. On the other hand, we can see very sensitive outputs from airborne mode. For being sensitive enough in hover mode, controller can sense all velocity changes. Moreover, it is not just velocity that caused by multi-copter itself, but also in wind situation. Therefore, to be this much sensitive, it would be right decision to choose airborne mode. As a result, for system design, “Airborne <2g” GPS mode is choosen.

2.5.8 Accelerometer for INS and AHRS

Accelerometer can sense acceleration for all three axiss. It can be seen from Figure 2.6: INS System, after offset, aligment, scale block and low pass filter block, system gather acceleration values for three axis of multi-copter body frame.

First of all, this acceleration values includes bias, too. This bias should be removed from accelerometer outputs. And this bias is calculated inside INS block which is aligned for navigation frame. Therefore, this bias values should be converted to body frame bias values, and then should be removed from accelerometer outputs. Also for INS and AHRS design, body frame to navigation frame should be known. So, the body frame to navigation frame conversation can be determined by [7];

$$R_b^n = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \tag{2.21}$$

For navigation frame to body frame which is R_n^b, can be calculated by inverse of this matrix.

After accelerometer bias values converted to body frame values, these bias should be removed from accelerometer values.

For AHRS design, from accelerometer, φ_{acc} and ϑ_{acc} values can be calculated. For this calculation, actually multi-copter is assumed to be there is no acceleration from any axis which also means that there is also one acceleration which is gravity. Under normal curcimstances, when multi-copter is standing on ground, accelerometer output should give -1g value for z vector which shows gravity vector. Because of skew ground and also measurement errors, absolute -1g can not be measured. But on the other hand, if we use this stable situation and divide axis each other and take arc tangent of this value, we can find angles of multi-copter. This concept is explained in Figure 2.17: Accelerometer angle calculation example.

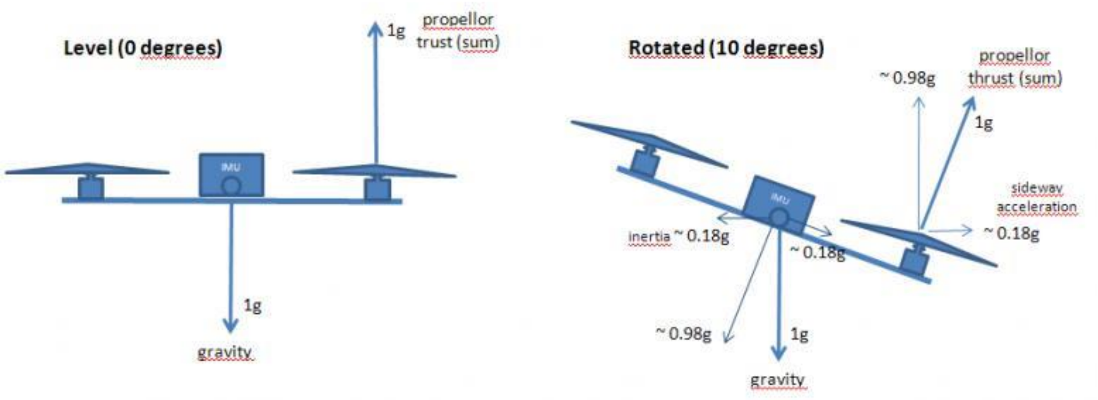

Figure 2.17: Accelerometer angle calculation example

And calculation is determined by

$$\text{rotation angle} = \operatorname{atan}\left(\frac{accx_{bias-free}}{accz_{bias-free}}\right) \tag{2.22}$$

which will be used for AHRS filtering. This rotation angles can be obtained from accelerometer which allows AHRS design an angle measurement source. This measured values will be used for AHRS design.

After calculating angles, INS need to obtain position and velocity information from accelerometer. Therefore, accelerometer values should be converted to navigation frame. The navigation frame type acceleration values can be obtained by;

$$\begin{bmatrix} acc_{k,nN} \\ acc_{k,nW} \\ acc_{k,nU} \end{bmatrix} = R_b^n \begin{bmatrix} acc_{k,bX} \\ acc_{k,bY} \\ acc_{k,bZ} \end{bmatrix} \tag{2.23}$$

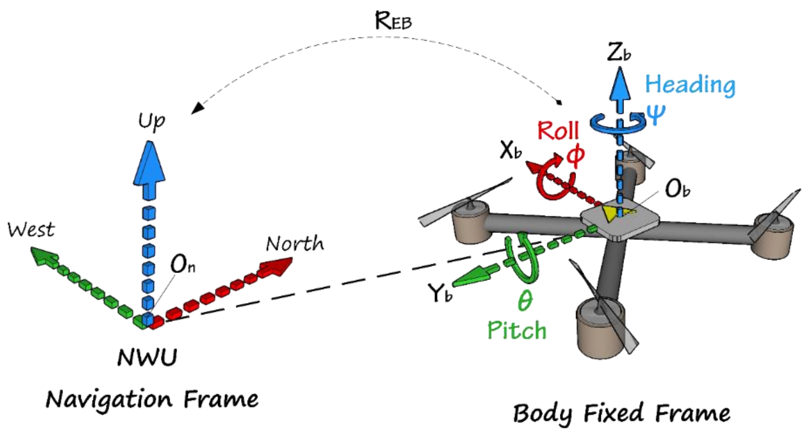

Figure 2.18: Body frame to navigation frame conversation [32]

Where $acc_{k,bX}$, $acc_{k,bY}$ and $acc_{k,bZ}$ are body frame acceleration values, $acc_{k,nN}$, $acc_{k,nW}$ and $acc_{k,nU}$ are navigation frame acceleration values.

After this conversations, system obtain navigation frame type acceleration values. Gravity vector is constant for navigation frame and it is direcly -1g for Up direction. Before all calculations, we cannot remove gravity from acceleration values because for different euler angles, accelerometer's gravity measurement always changes. The gravity is not always remaining at Z direction for body frame, but for navigation frame, gravity always at Up direction. Therefore, gravity value can be removed from navigation frame type acceleration values.

Before removing this vector, system need to know exact magnitude of gravity, because accelerometer measurement includes measurement errors, therefore accelerometer cannot measure gravity always in -1g. So, before flight, multi-copter is stanting still, gravity magnitude should be calculated by find intersection vector. Ofcourse, this value also includes some errors. For solve this problem, system can measure gravity not once but many times and take mean value. That's an advantage for thinking these errors as zero-mean gaussian noises. After this calculation, gravity magnitude should be removed from Up axis acceleration value.

As it is shown at equation accelerometer output equations, what we need to measure is $\dot{u}$, $\dot{v}$ and $\dot{w}$ which is actually the acceleration values that removed from all other errors and effects. Thus far, noise part is tried to be minimized with low pass filter and also with converting acceleration values to navigation frame, gravity vector also removed. Therefore, now system obtained acceleration values are:

$$a_x = \dot{u} + qw - rv + \eta_{accel,x} \tag{2.24}$$

$$a_y = \dot{v} + ru - pw + \eta_{accel,y} \tag{2.25}$$

$$a_z = \dot{w} + pv - qu + \eta_{accel,z} \tag{2.26}$$

In equations, noise value did not remove because with low pass filter, this noise cannot absolutely have removed. But on the other hand, there is still some other values that acceleration values should not include this which is red highlighted. This part of equation are coriolis and centrifugal effects.

Coriolis force [15] is an inertial force which acts on objects in motion relative to a rotating referance frame. When Newton's laws are transformed to a rotating frame of reference, the coriolis force and centrifugal force appear and these forces are proportional to the mass of the object. For aviation part, this coriolis effect is caused by the rotation of the Earth. For instance, a multi-copter travels in a straight line through space. Because of the rotation of the Earth, the object will appear to veer to the right in the northern hemisphere or back to the left in the southern hemisphere. Coriolis acts on the air as well as an object flying through it. Air drawn towards an area of low pressure will actually move along the Isobars, the gradient force created by the pressure difference being balanced by the coriolis effect. Thus wind travels clockwise around a area of high pressure in the northern hemisphere and anticlockwise around an area of low pressure. The coriolis effect varies with ground speed (or wind speed) and is greatest at the Poles and zero at the Equator.

Centrifugal force is used to refer to an inertial force directed away from the axis of rotation that appears to act on all objects when viewed in a rotating reference frame.

These both forces have an effect on each axis. Therefore, these parts should be removed from measurement. Because these values always manipulate to different acceleration value and this situation will lead some drift effects on INS and AHRS.

$$\text{Euler Acceleration: } -\dot{\Omega} \times r \tag{2.27}$$

$$\text{Coriolis Acceleration: } -2\Omega \times V \tag{2.28}$$

$$\text{Centrifugal Acceleration: } -\Omega \times (\Omega \times r) \tag{2.29}$$

Table 2.1: Coriolis and centrifugal forces on each axis

	North Axis	East Axis	Down Axis
Acceleration component			
Coriolis	$2V_E\Omega\sin\lambda$	$-2V_N\Omega\sin\lambda - 2V_D\Omega\cos\lambda$	$-2V_E\Omega\cos\lambda$
Centrifugal	$(V_E^2\tan\lambda - V_D V_N)/R$	$-(V_N V_E\tan\lambda + V_D V_E)/R$	$(V_N^2 + V_E^2)/R$
Gravitational			$\frac{R_0^2}{(R_0+H)^2}g_0$

Table 2.2: Content of coriolis and centrifugal equations

Symbol	*Default Value*	*Unit*	*Expression*

Ω	7.2717×10^{-5}	$rads/s$	
R_0	6,378,137	m	Earth's Radius
R	depends on the vehicle position	m	$R_0 + H$
H	depends on the vehicle position	m	Altitude
λ	depends on the vehicle position	$degree$	Latitude
g_{equ}	9.7803267714	m/s^2	
k	0.00193185138639		
e^2	0.00669437999013		e: the first eccentricity
g_0	depends on the vehicle position	m/s^2	local gravity

After removing this part, system finally obtains navigation frame reference acceleration values also consist of noise. For noise problem, filtering techniques is used. So, with these acceleration values, now system can obtain velocity and also position values integrated from acceleration values.

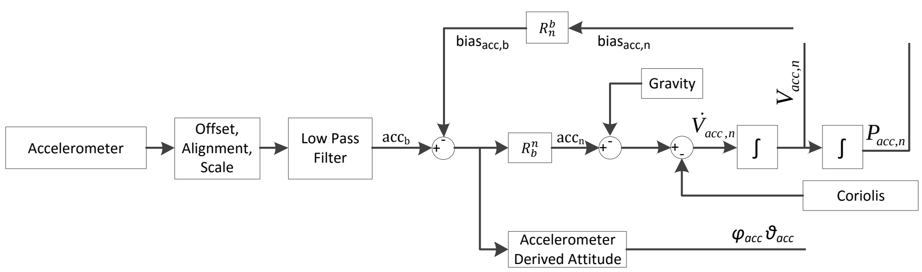

Figure 2.19: Accelerometer Schematics

So as a result of these part, system can obtain $V_{acc,n}$ and $P_{acc,n}$ which will be used for INS. Also φ_{acc} and ϑ_{acc} obtained and will be used for AHRS.

2.5.9 Body frame angular rate to euler angles

As it is mentioned before for body frame type acceleration values, conversation from body frame to navigation frame should be done for angular rate values, too. Therefore, conversation is determined by [7];

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & (\sin\phi \tan\theta) & (\cos\phi \tan\theta) \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Figure 2.20: Body frame to navigation frame gyroscope conversation

2.5.10 Obtained information for INS and AHRS

With all these block calculation, system obtain usable data for filtering. So, from Figure 2.6: INS System, it can be seen which output become input for INS and also AHRS. To be more precise;

For INS System:

- Position value
  - GPS, $P_{gps,n}$
  - Accelerometer, $P_{acc,n}$
- Velocity value
  - GPS, $V_{gps,n}$
  - Accelerometer, $V_{acc,n}$
- Altitude value
  - GPS, $Alti_{gps}$
  - Accelerometer, $Alti_{acc}$
  - Barometer, $Alti_{baro}$

For AHRS System:

- Euler Angles
  - Gyroscope, ϑ_{gyro}, φ_{gyro}
  - Accelerometer, ϑ_{acc}, φ_{acc}
- Heading Angle

- Gyroscope, ψ_{gyro}
- Magnetometer, ψ_{mag}

For fixed-wing INS design, for heading angle calculation, system directly use GPS heading value because of being accurate enough to use. But for multi-copter platforms, it can hover in mid-air which do not have any velocity through heading direction. So, GPS heading value is always changes because GPS cannot measure without velocity. That's why for heading angle, system should use gyroscope and magnetometer sensors.

2.6 State Estimation Filters

After gathering these all usable sensor output values through navigation frame, system obtain more than one source for estimate multi-copter's state. Therefore, for more accurate state estimation, system need to use these all sensor informations. As it is mentioned before, in literature, there are some filtering methods such as complimentary filter, kalman filter, extended kalman filter, information filter and so. In this design, system uses complimentary filter and kalman filter. For next sections these two filtering technique is explained.

2.6.1 Complimentary filter

Complimentary filter is basic filtering technique that can be used for any sensor fusion algorithm. Main idea is use two different source of information and combine these informations with some low or high pass filter and find more accurate and usable information. [A Comparison of Complementary and Kalman Filtering]

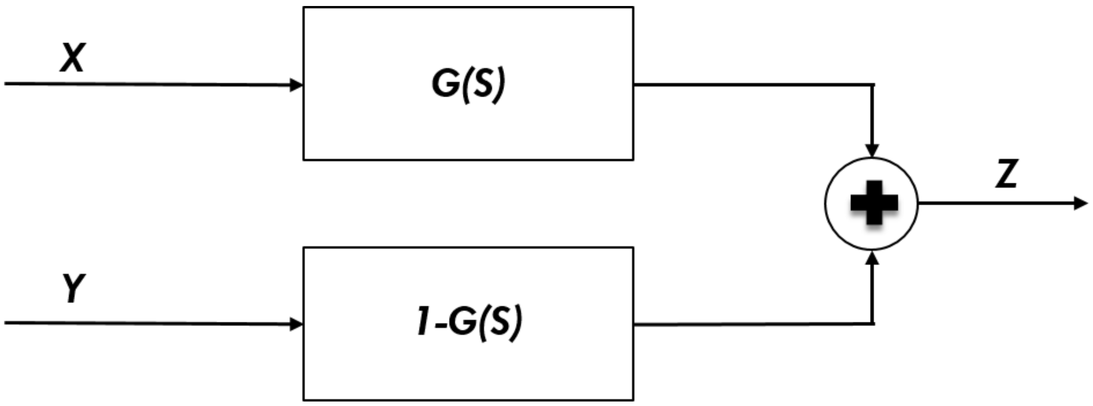

Figure 2.21: Complimentary Filter

The basic complimentary filter is shown in Figure 2.21: Complimentary Filter, where X and Y are noisy measurements of some signal Z. Assume that the noise in Y is mostly high frequency, and the noise in X is mostly low frequency. Therefore, G(s) can be made a low-pass, so [1-G(s)] is the complement. There is no detailed description of the noise processes are considered in complimentary filtering.

Also basic calculation can be determined by [16],

$$Z = \alpha_{comp}X + (1 - \alpha_{comp})Y \tag{2.30}$$

Where α_{comp} can be constant value between 0 and 1.

2.6.2 Kalman Filter

The kalman filter can be summed up as an optimal recursive computation of the least-square algorithm. It is a subset of a bayes Filter where the assumptions of a Gaussian distribution and that the current state is linearly dependant on the previous state are imposed. The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance when the condition of a linear gaussian system is met. If the Gaussian assumption is relaxed, the Kalman filter is still best (minimum error variance) filter out of the class of linear unbiased filters. [Rudolph van der Merwe, Alex T. Nelson, Eric Wan. "An Introduction to Kalman Filtering." OGI School of Science & Engineering lecture. Oregon Health & Sciences University. November 2004.]

The Kalman filter recursively estimates the state of a dynamic system based on a sequence of noisy measurements and starting from an initial condition which is also can assumed random variable or specific value. The current state depends on previous state and also current sensor measurement.

There are two parts for the kalman filter which are prediction part and correction part [16].

Time Update:

- Project The State Ahead

$$x_{k+1}^{-} = A_k x_k + B u_k \tag{2.31}$$

- Project the error covariance ahead

$$P_{k+1}^{-} = A_k P_k A_k^T + Q_K \tag{2.32}$$

Measurement Update:

- Comute Kalman Gain

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{2.33}$$

- Update estimate with measurement z_k

$$x_k = x_k^- + K(z_k - H_k x_k^-) \tag{2.34}$$

- Update the error covariance

$$P_k = (I - K_k H_k) P_k^- \tag{2.35}$$

Where;

- k: *Time Step*
- x_k: *State Vector at k*
- u_k: *Control Vector at k*
- $w_k \approx N(0, Q_k)$: *Process Noise at k*
- $v_k \approx N(0, R_k)$: *Measurement Noise at k*
- Q_k: *Process Covariance*
- R_k: *Observation Covariance*

2.6.3 AHRS Dynamic Model

Actually for AHRS design, kalman filter and complimentary filter is used to compare each other. As a rasult of this, complimentary filter is choosen. It is mentioned before that for AHRS design we have:

- Euler Angles
  - Gyroscope, ϑ_{gyro}, φ_{gyro}
  - Accelerometer, ϑ_{acc}, φ_{acc}
- Heading Angle
  - Gyroscope, ψ_{gyro}
  - Magnetometer, ψ_{mag}

Whenever system try to find angle from gyroscope, because of output of gyroscope is angular rate value, system takes integral and find angle values. But before integral, system should know the initial value which is obtained from accelerometer when multi-copter is standing ground. After that the calculation for state estimation starts.

Accelerometer gives a good indicator of orientation in static conditions. Gyroscope gives a good indicator of tilt in dynamic conditions. So the idea is to pass the accelerometer signals through a low-pass filter and the gyroscope signals through a high-pass filter and combine them to give the final rate. The key-point here is that the frequency response of the low-pass and high-pass filters add up to 1 at all frequencies. This means that at any given time the complete signal is subject to either low pass or high pass.

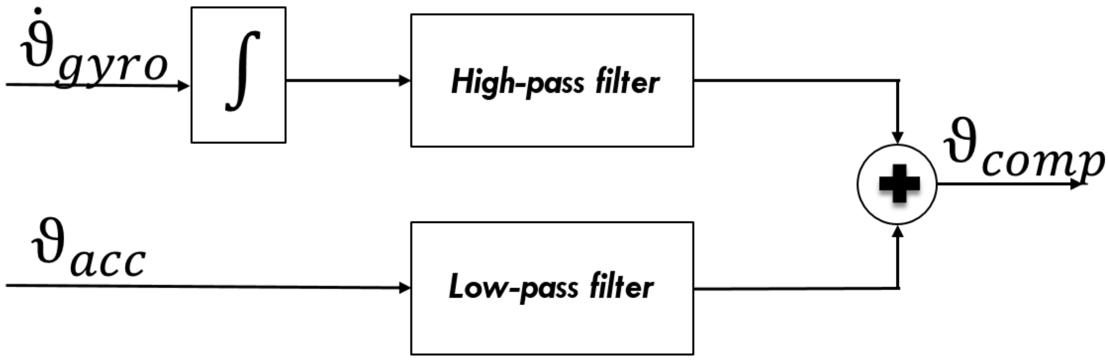

Figure 2.22: Complimentary filter for pitch angle

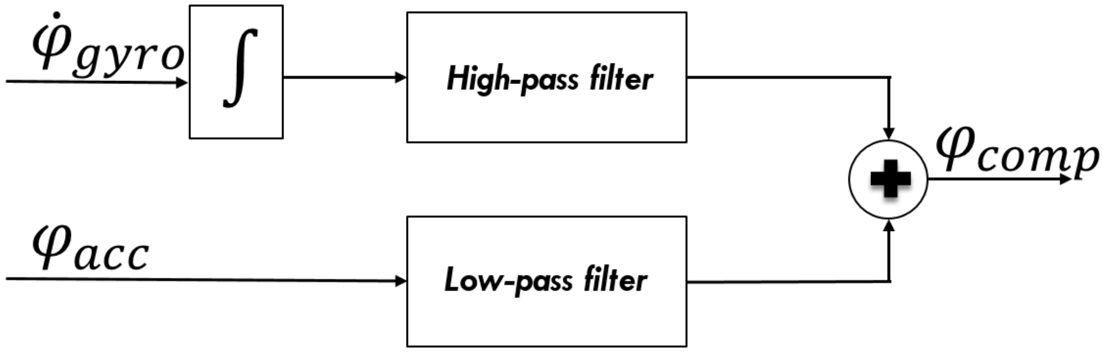

Figure 2.23: Complimentary filter for roll angle

In Figure 2.22: Complimentary filter for pitch angle and Figure 2.23: Complimentary filter for roll angle, there are integral block which is conversation angular rate to angle. For detail conversation is shown by,

$$\varphi_k = \varphi_{k-1} + dt * (\dot{\varphi}_k) \tag{2.36}$$

$$\vartheta_k = \vartheta_{k-1} + dt * (\dot{\vartheta}_k) \tag{2.37}$$

$$\psi_k = \psi_{k-1} + dt * (\dot{\psi}_k) \tag{2.38}$$

As it is mentioned before, gathered usable gyroscope values should be integrated for converting to the angle. Then, these two values are used for input of complimentary filter for both pitch and roll angle calculation. Because of drift effect on gyroscope, even after flight, it can be seen directly that calculated angles are effected by drift. But with some time, because of angle obtained from acceleration has no drift effect, the angle estimation is corrected. It can be thought as a kalman filter. High-pass filter is prediction part and low-pass filter is correction part.

$$\text{High-pass filter : } \alpha_{comp} = 0.95 \tag{2.39}$$

$$\text{Low-pass filter: } (1 - \alpha_{comp}) = 0.05 \tag{2.40}$$

There is always a drift effect on gyroscope. But on the other hand, because of acceleration for all directions, angle obtained from accelerometer is not always accurate enough for estimate angle values. Although gyroscope has drift effect, gyroscope is very dependable for estimating angle. Therefore, gyroscope values are used with high-pass filter in complimentary filter.

For now, with complimentary filters, we calculated pitch and roll angles that can be used accurately for AHRS. For another information we need is heading value. For find heading value, system has gyroscope and magnetometer sensor datas.

It is mentioned before that gyroscope has drift effect on measurement. But it is always more reliable than other sensors. Therefore, gyroscope will be used in high-pass filter part of complimentary filter. On the other hand, magnetometer is directly measures the Earth's magnetic field and works like a compass. It measures magnetic field and gives heading output which is reference to North. But is not always reliable sensor for measure heading. Because if there is some another magnetic field, it cannot measaure just Earth's magnetic field but also other fields if exist in that area. So the calculation of heading would be wrong. For other magnetic fields, multi-copter's motor's magnetic interferences can be a good example for this. But generally, there is no another magnetic field if magnetometer is placed right which do not be affected from motor's magnetic interferences.

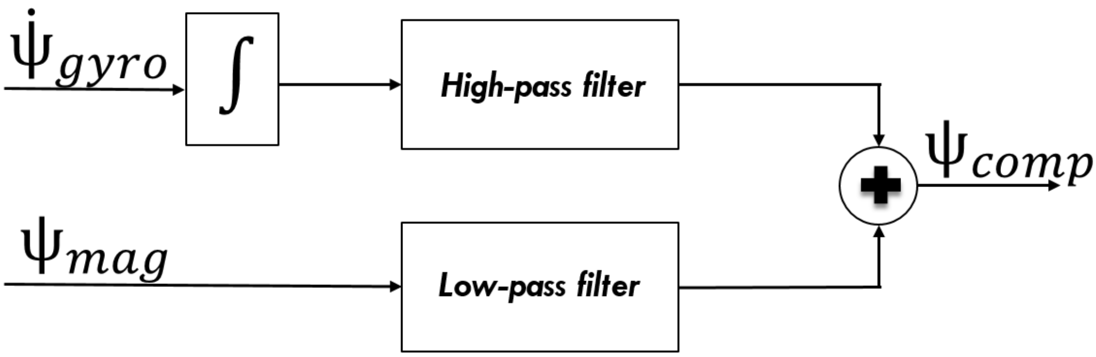

Figure 2.24: complimentary filter for heading

$$\text{High-pass filter : } \alpha_{comp} = 0.90 \tag{2.41}$$

$$\text{Low-pass filter: } (1 - \alpha_{comp}) = 0.1 \tag{2.42}$$

For AHRS, these complimentary filter's coefficients should be decided. For this system design, these coefficients are received by hand tuning.

2.6.4 INS design

INS compose usable velocity and position information. These informations are very important for multi-copter controller system. AHRS is used for low level control that multi-copter can be used with remote control. But when multi-copter is needed to do navigation such as waypoint tracking and position hold, controller needs position and velocity information which is gathered from INS.

Data Source	Unit	Rate [hz]
Acceleration North	m/sn^2	800
GPS Speed North	m/sn	5
GPS Position North	m	5
Acceleration West	m/sn^2	800
GPS Speed West	m/sn	5
GPS Position West	m	5
Acceleration Up	m/sn^2	800
GPS Speed Up	m/sn	5
GPS Altitude	m	5
Barometer Altitude	m	800

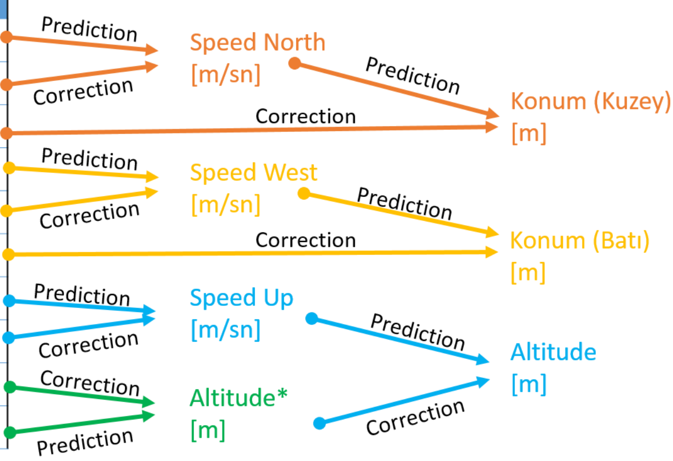

Figure 2.25: General picture of INS filtering system

In Figure 2.25, general expression of filtering is shown. GPS has 5 Hz output rate and other sensors has 800 Hz. But for INS design, these sensors rate will be used at 200 Hz.

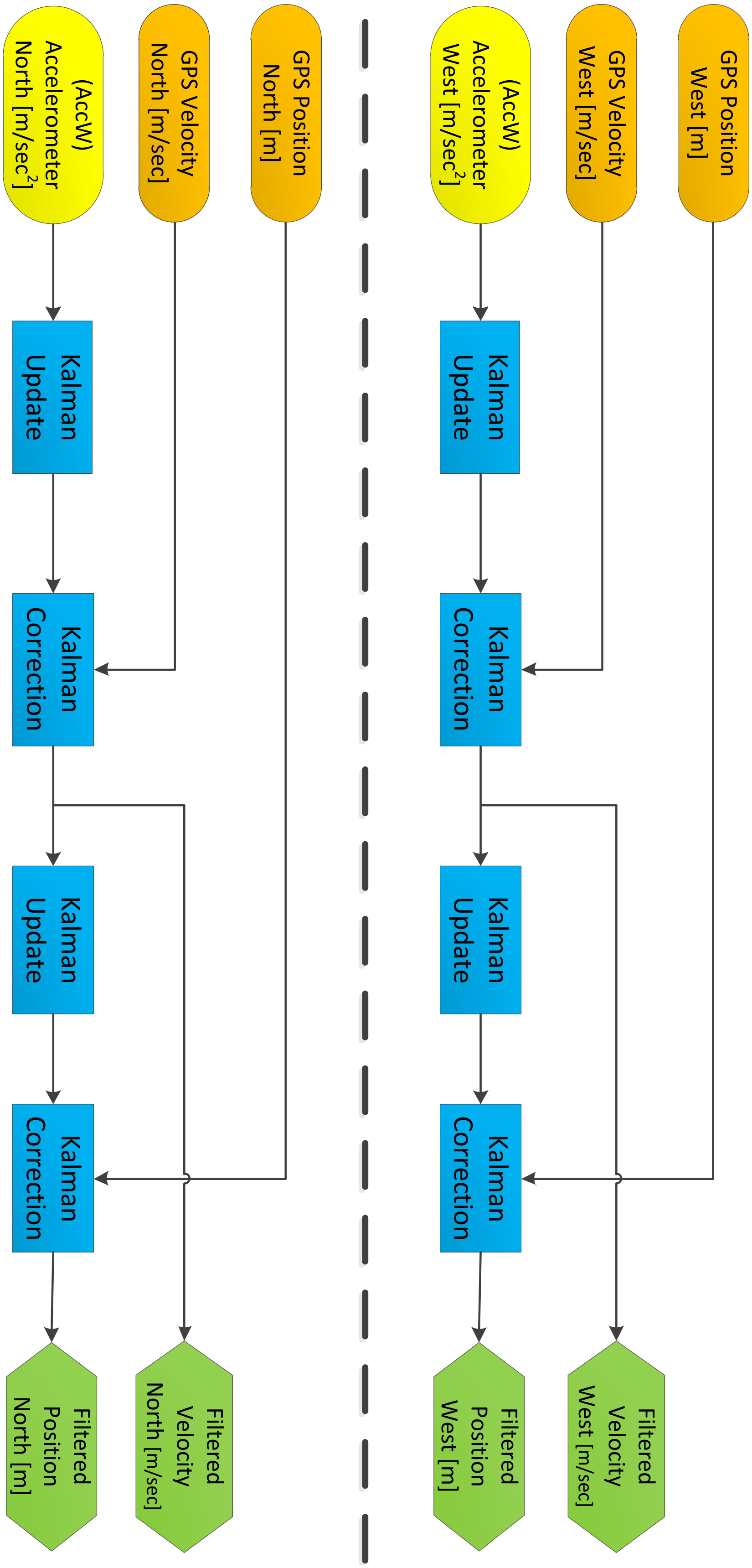

Figure 2.26: vertical speed and position values from INS

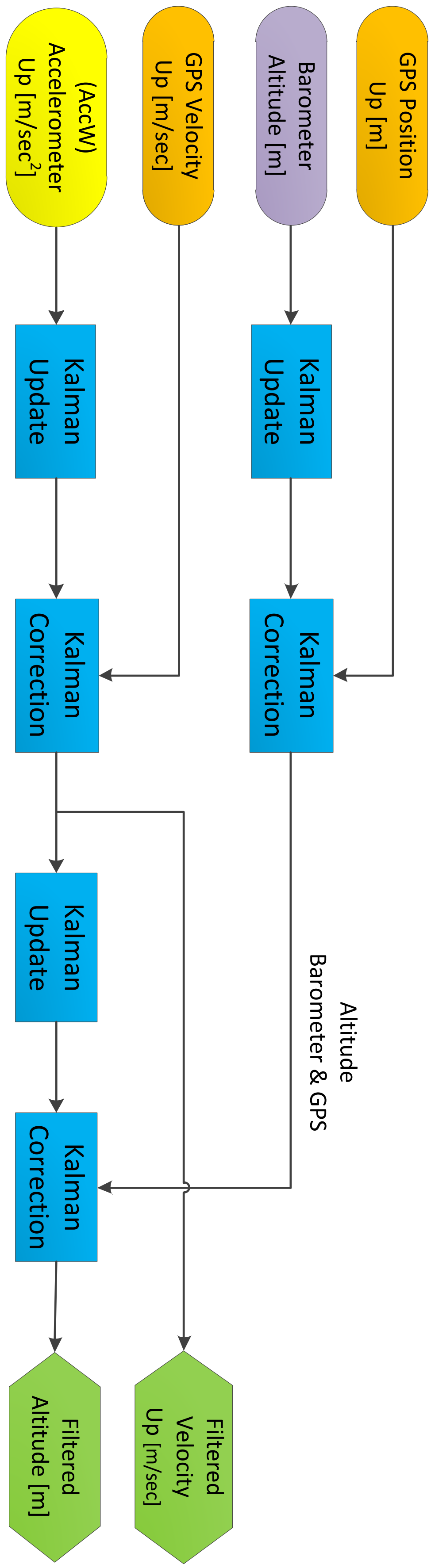

Figure 2.27: Altitude from INS

INS design is shown in Figure 2.26 and Figure 2.27. For Figure 2.26, position and velocity estimation are made for North and West axis. There are two specific important issue to use this filter in this way. First one is GPS is used in correction phase. As it is mentioned before, sensors except GPS are all have some drift or bias kind errors that can not be solved. Because of that, GPS is the only most reliable sensor that can be used for this system. On the other hand, second important issue that GPS has 5 Hz frequency for transmitting output values. 5 Hz frequency is not enough for multi-copter autopilot system which is working on 100 Hz sampling time. Because of this, for correction part GPS is used and for prediction part, other sensors are used.

For Figure 2.26, kalman filter is used. Kalman filter calculations are common for all other kalman filter calculation except the rigid body dynamic model. In this model,

$$x_{k+1}^{-} = A_k x_k + B u_k \tag{2.43}$$

Common rigid body dynami model should be given. Therefore, for position and velocity estimation, to find these values, used equations are;

$$p = p + vt + \frac{1}{2}at^2 \tag{2.44}$$

$$v = v + at \tag{2.45}$$

So, for conversion as a dynamic model,

$$A = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} dt^2/2 \\ dt \end{bmatrix} \tag{2.46}$$

$$x_k = \begin{bmatrix} p \\ v \end{bmatrix}, u_k = Acc_{accel}, z_k = P_{gps,n} \tag{2.47}$$

where dt is relative to frequency of sensor output, p is position, v is velocity, a is acceleration and $P_{gps,n}$ is position information from GPS. As it is explained before, GPS has 5 Hz, and sensor outputs has 200 Hz. Therefore, 40-times prediction and 1-time correction is done by kalman filter.

For altitude estimation, there is also another sensor which is barometer. So system has barometer, GPS and accelerometer for estimating exact altitude.

In altitude estimation, first barometer altitude and GPS altitude values are used in kalman filter to find more accurate altitude value. But both sensor output has position value. Therefore, there is no such dynamic model for both the same inputs. But, for acceleration and position integration, there is simple equation which is already used for position North and West estimations. Therefore, system need to use different type of kalman filter which find bias differences and compensate these difference.

The kalman filter of barometer altitude and GPS altitude is given by;

$$x_{k+1}^{-} = Alti_{baro} - \beta_{baro} \quad (2.48)$$

$$\beta_{baro} = \beta_{baro} \quad (2.49)$$

So, model matrixs are become;

$$A = \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2.50)$$

$$x_k = \begin{bmatrix} x \\ \beta_{baro} \end{bmatrix}, u_k = Alti_{baro}, z_k = Alti_{gps} \quad (2.51)$$

By this model, filter finds bias of barometer altitude and GPS altitude. As it is explained before that GPS has 5 Hz low frequency rate, but has accuracy in 5 meters. On the other hand, barometer altitude output cannot give exact altitude value. Because in altitude equation of barometer, system should know exact pressure of ground. However, system do not know this pressure, therefore barometer altitude output cannot be exact altitude value. However, barometer has 200 Hz output value and very sensitive for sensing pressure. But because of GPS has 5 Hz output, GPS do not have sensitive output values and also cannot keep up with autopilot controller. So the main idea is remove this bias which has between barometer altitude and GPS altitude, and take dynamic behavior of barometer and put it on GPS altitude. In short time barometer, but in long time GPS altitude value will be effective.

2.7 Test Results

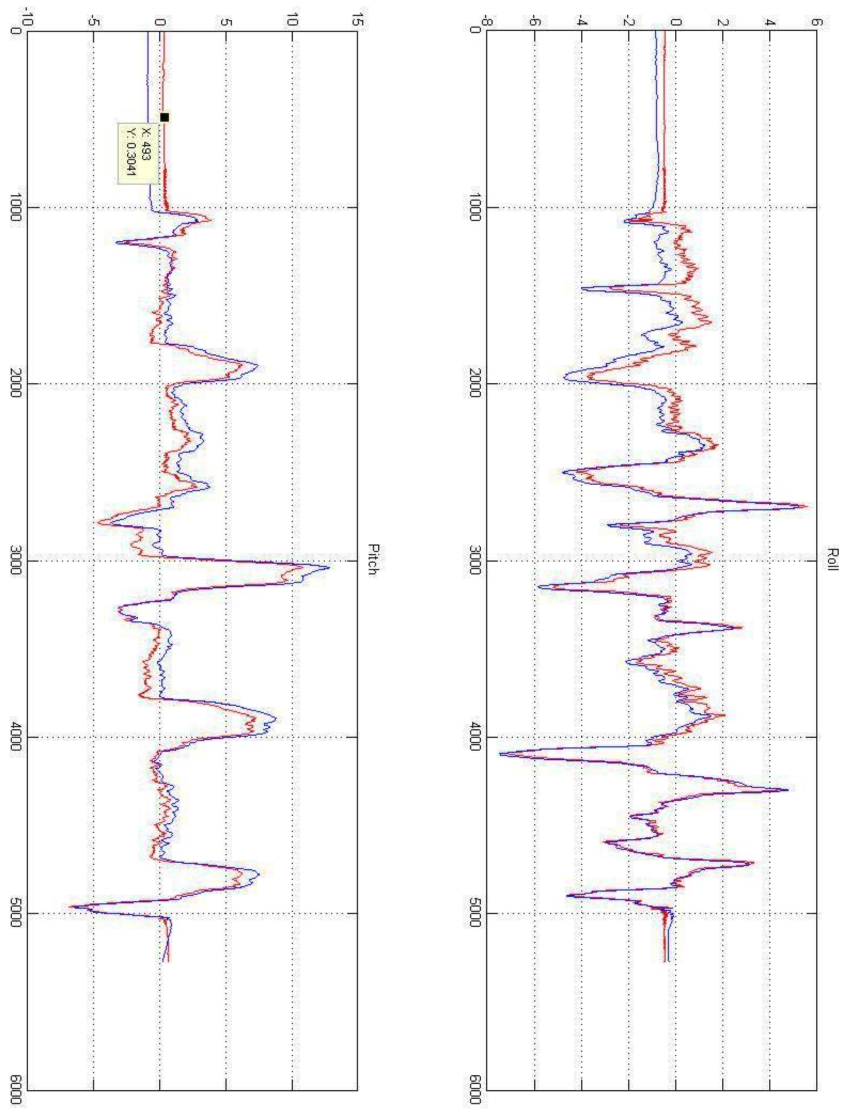

Figure 2.28: AHRS design roll and pitch outputs compared with XSENS INS (red – Xsens, blue- designed AHRS)

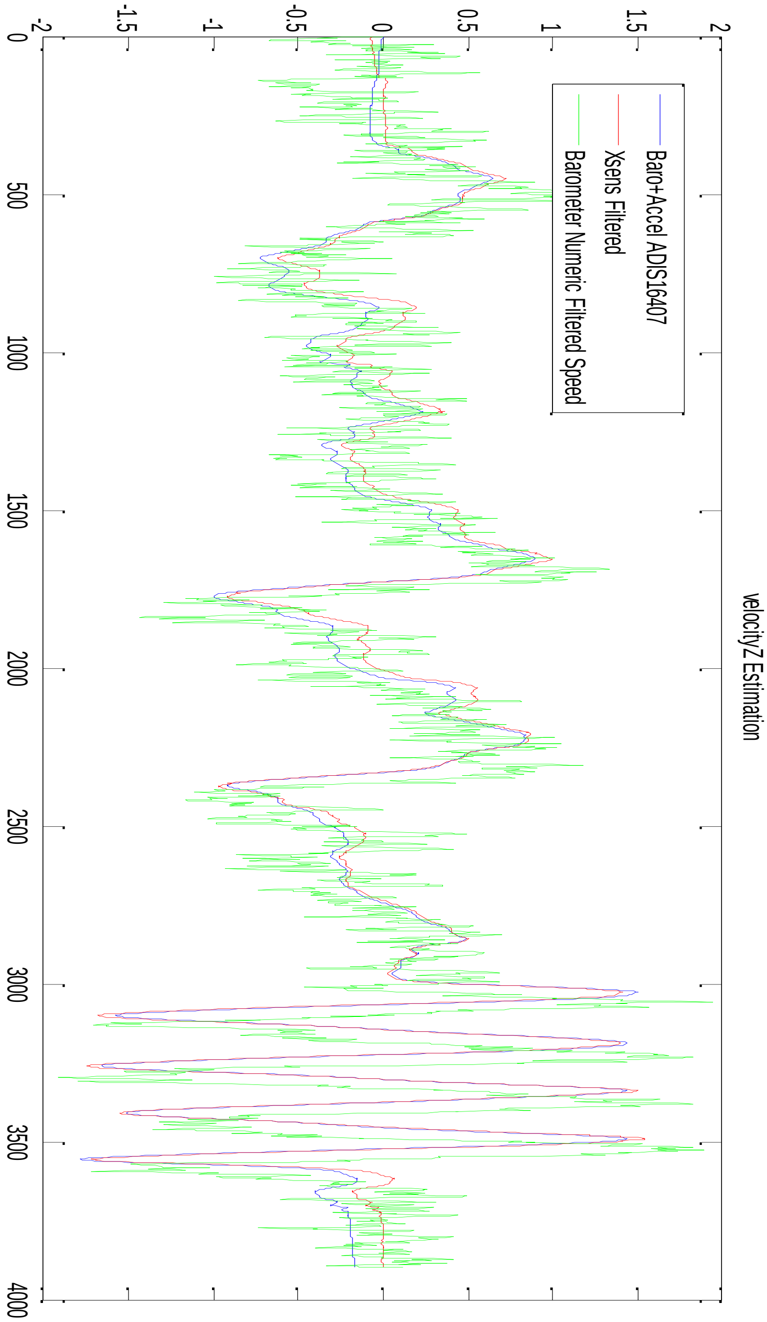

Figure 2.29: Velocity Z estimation comparison

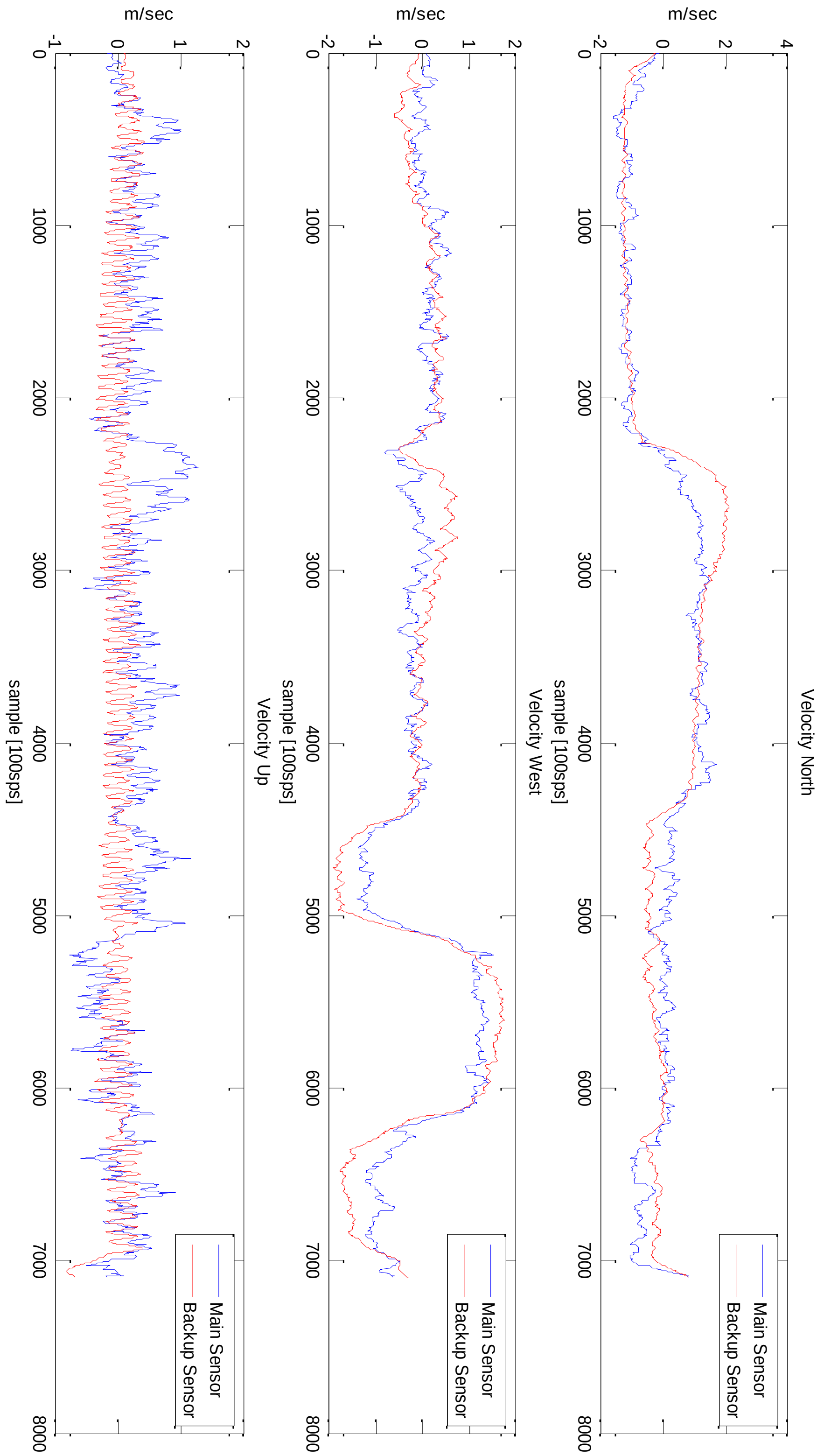

Figure 2.30: Velocity Comparison

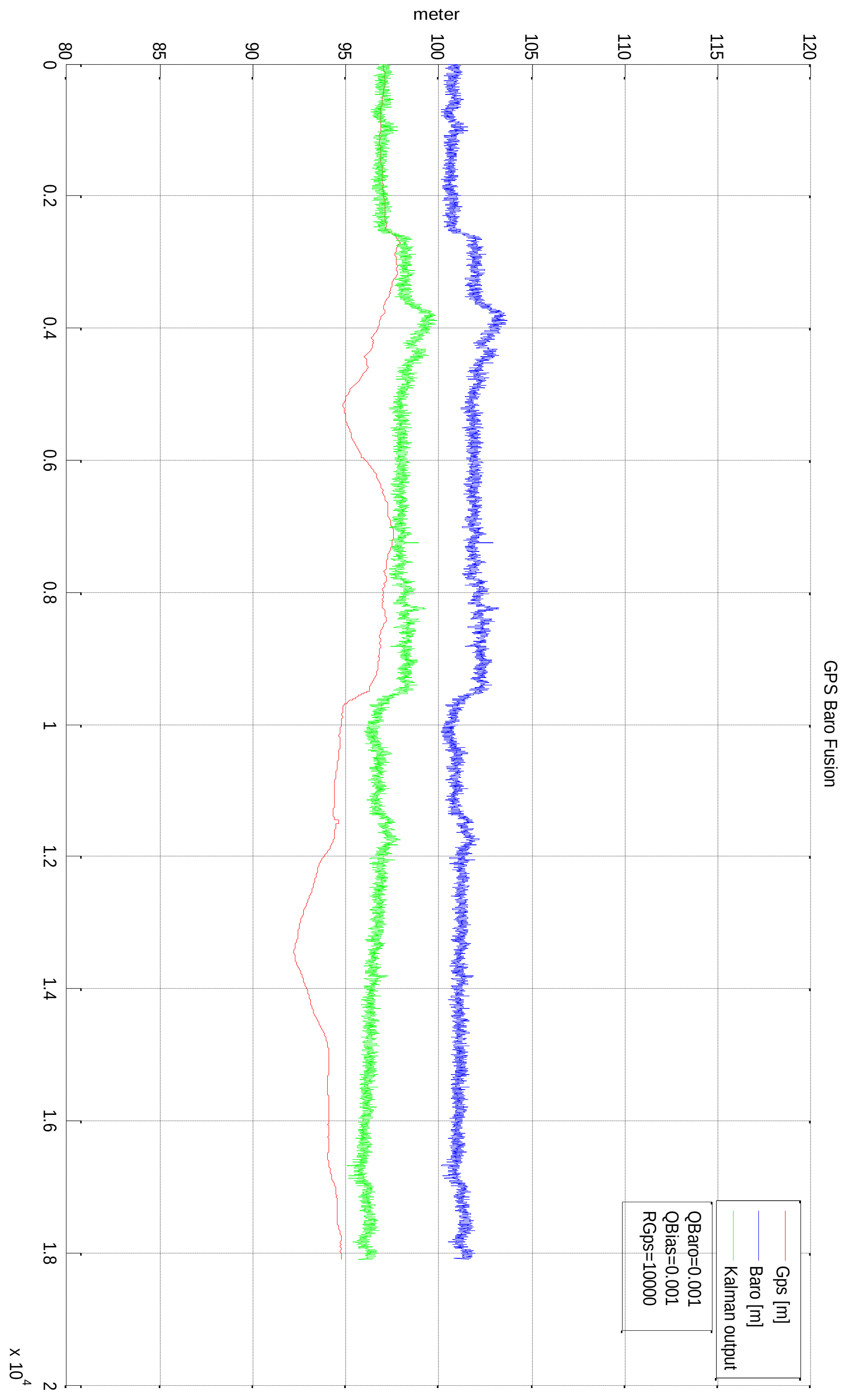

Figure 2.31: Barometer-GPS fusion result

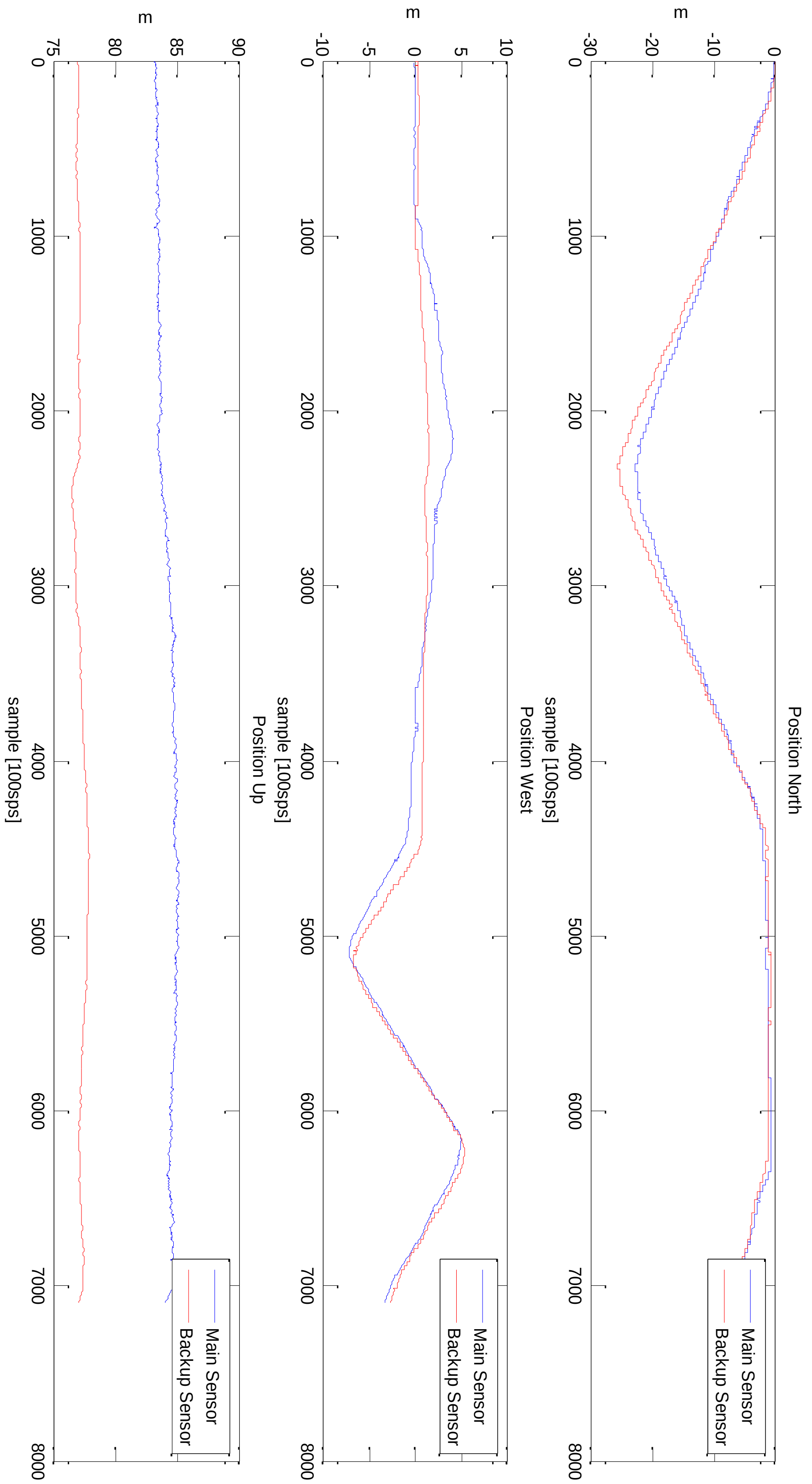

Figure 2.32: Position estimation results and comparison with XSENS

3. FLIGHT MANAGEMENT SYSTEM WITH COLLISION AVIODANCE SYSTEM

3.1 Introduction

Over the last 20 years, unmanned aerial systems for civil applications are being operationally more efficient, cost effective, and having high-end capabilities. The new functionalities come into UAV flight management systems allow ground operators to focus on higher level tasks including not only operating single vehicle but also managing the entire operation with large scale UAV fleet. The current practice of the UAV operations is to segregate certain areas of the airspace for their use. However, growing demand in both use of UAVs and commercial air traffic will make unfeasible existing procedures and will require to build joint airspace structure. Future intensive use of UAVs for civil applications will require appropriate integration into general aviation.

The FAA Modernization and Reform Act of 2012 [17] mandates for full integration of UAS into the NAS by 2015, but according to [18], issues of lack of UAS interaction links with Air Traffic Management System, and non-standardised performance/behavioural characteristics still continues as major challenges in integration of UAVs into the national airspaces. Therefore, next generation UAV flight management systems should contain additional data links in order to make themselves visible in 4-dimension for both ground systems and other aircrafts. In addition to this, they should have also own sense-and-avoid system operates independent from both ground and air systems for safety redundancy.

In this thesis, the authors propose a novel UAV Flight Management System (FMS) structure that integrates two-level autonomy modes in order to meet the operational requirements of the future UAV operations. In a nominal flight operation, the FMS operates in a collaborative manner where UAVs exchange their intent with ground systems and other aerial vehicles using a formal intent description language through air-to-air and air-to-ground (e.g. ATC and C2 segments) data links. These data links using formal intent languages enable ground segment (human) to ”talk” with aircraft (machine), and convey their diverting command at different levels. Whenever the

ADS-B equipped FMS detects a potential collision (immediate-term) with other aircraft and terrain obstacles generates and executes an 3D evasive maneuver in order to solve the issue. The FMS handles switching these safety modes considering the required response time. For technological demonstrations and operational validations, an experimental prototype of the FMS has been deployed on a Quadrotor testbed, and a command and control (C2) station has been built. Due to page limitation, the paper mostly covers hardware and software design considerations of the FMS, and leaves the algorithmic and implementation details to the future publications.

Figure 3.1: On-board camera capture from flight tests of the quadrotor UAV testbed

In order to enhance the predictability of the aircrafts's future path, trajectory planners have begun to utilize a wide range of information including reference intention. In algorithmic side, a formal methodology has been introduced in [19] and applied into small UAV. In [20], intent information prediction by observing aircraft motion has been studied, and in addition to modal estimation, [21] has also utilised flight plan information. In the similar fashion, [22] has presented a probabilistic myopic intent estimation method for an intruder UAV with uncertain goals and motivations. The intent based probabilistic trajectory planning method using a hybrid model has been proposed in [23]. In application side, Intent based approaches have begun to appear in air traffic management, for example, in ground based systems. These tools require almost complete knowledge of the aircraft intents and assume that aircraft follow the advisories of air traffic controller (ATC) and standard flight rules. However, the picture of the future airspace with various type of aircraft (e.g. private aircrafts, commercial planes, UAVs etc.) is envisioned that flight plan and reference trajectory continue to evolve over time in order to meet the dynamic constraints and achieve changed objectives [24], [25]. To address this issue, both the aircraft and ground systems will be handling shared flight data to build a" shared situational awareness" on trajectory evolution. Through these considerations, Boeing Research and Technology Europe has studied on the trajectory synchronization problem between the different trajectory planners and proposed three-level formal description languages (AIDL, ICDL and FIDL). These languages have enabled to efficiently define an action sequence of the aircraft dynamics or the flight plan with different levels of detail, fully or partially specifying some aspects of the aircraft motion. [26].

Figure 3.2: Graphical User Interface screen on the ground station during flight test

In the flight operations, the multi-layer safety structure plays a major role in ensuring safety especially in the high level autonomous systems such as UAVs. Through new concepts of the future use of airspace that redefines responsibility, aircraft must also be equipped with multilayer safety automation where at least one must work independently from the ground or air [25]. This structure will reduce dependence to the ground and isolate the system from common mode failures such that single data error would invalidate the entire system. By considering these facts, nonintent-based collision avoidance (i.e. Airborne Collision Avoidance or Sense-and-Avoid Systems), which does not require any knowledge on the aircraft intent, will still be crucial when the collaborative separation assurance process fails. The limitation of this method is that the prediction error tends to grow quadratically with time; therefore, these types of tools will still remain in the domain of the immediate to short-term collision avoidance. The method in [27] which is a modal-based probabilistic short-term collision avoidance has been integrated in this system.

3.2 Onboard Flight Management System for UAVs: Experiment Quadrotor Testbed

The integrated system is envisioned to integrate two layers of safety mode into the onboard flight management system (FMS) of the aircraft in order to meet the requirements of the future UAV flight operations. The FMS handles switching these safety modes considering the required response time. The ”required response time” term is defined as the min time for creating an appropriate response (includes comprehending, evaluating and reacting) to solve the occurring and evolving situation. These two process cycles at different autonomy levels are represented with *Intent Based Planning* and *Sense-and-Avoid* modules where both are involving different procedures and algorithms. Figure 3.6 demonstrates this entire integrated functionalities and its add-on modules.

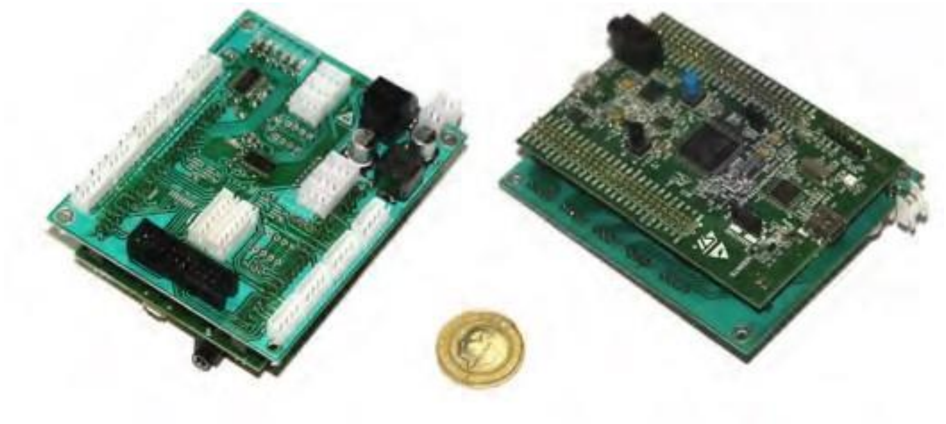

Figure 3.3: Custom flight control computer (FCC) hardware

In the mid-term horizon (in couples of minutes) processes are operated in a collaborative manner. In this domain, it is expected that the aircraft cooperates with the ground command and control systems. This module incorporates all tactical level information (i.e. weather data, intent data, user preferences data and traffic data) obtained from both on-board sensing (including air-to-air data link) and air-toground data exchange. In this mode, *Replanning Request (ReP)* can be initiated by either the UAV or the ground system. The ground based ReP request may emerge in some circumstances such as drastic change in operational constraints, conflict detection, emergency situations or detection of an aircraft does not conform to the anticipated behaviour. Similarly, the UAV may also create an ReP request cycle when the on-board *Conflict Monitoring* detects a potential conflict. *Trajectory Computation Infrastructure* (TCI) which its details will be given in the following section, automatically validates the feasibility of the given intent data, and *Conflict Monitoring* block checks potential conflicts between the predicted trajectories. This structure allows lowlevel intent sharing between the aircraft in the surrounding traffic through the air-to-air data link (see the Figure 3.6). This low-level intent sharing is the” machine-to-machine” communication, which inherently makes unmanned systems visible for also commercial aircraft, envisions to integrate the UAVs into the national airspace. This function provides the aircraft to have more accurate information about near airspace than the ground systems. Furthermore, in an efficient manner, aircraft can monitor the conflicts more frequently than the ground systems for the interested local region. Specifically, potential conflict may be detected on-board while ground based system can not yet detect the same situation due to the lack of accurate local information.

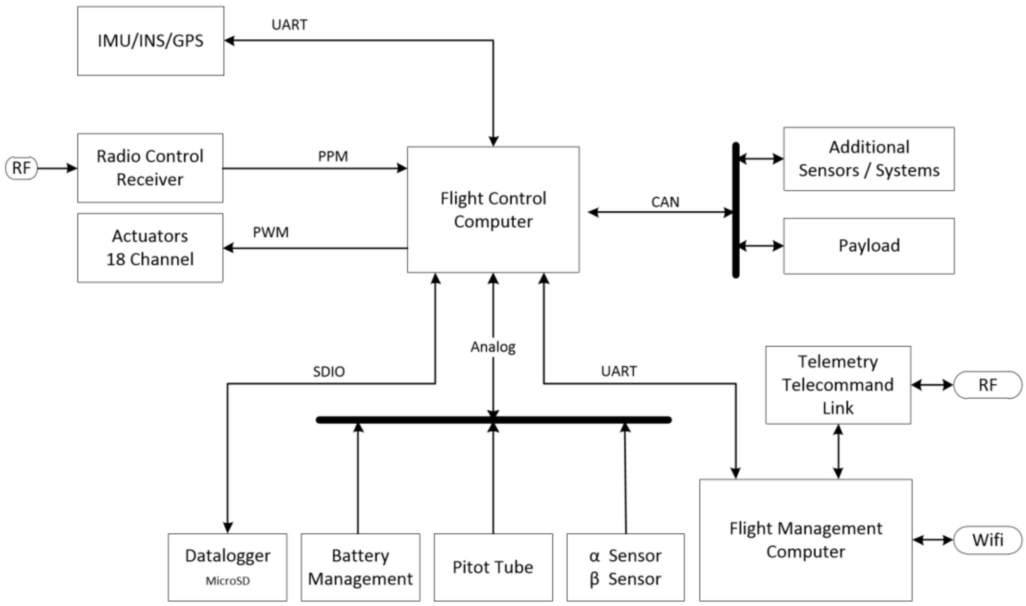

Figure 3.4: Hardware architecture of Flight Management System

The Sense-and-Avoid module (seen in the Figure 3.6) is an isolated system from the intent data exchange and works independently. Thus, it provides redundancy in the aircraft. This module only uses position data of the aircraft in the surrounding traffic obtained via ADS-B based onboard sensing. The Sense-and-Avoidance block persistently monitors occurrence probabilities of potential collisions with other aircraft and terrain obstacles for bounded local region. The conflict detection algorithm uses worst case approach and takes into account both uncertainties in position measurement and aircraft actions (e.g failure in control). Whenever the immediate threat(s) is/are detected (i.e. immediate response is required), the autopilot system executes evasive behaviour to solve the issue with required 3D avoidance maneuvers which is generated by *Sense-and-Avoidance* block. These avoidance maneuvers including recovery generate small deviations where their impacts on the entire flight route is minimized.

General architecture of the FMS is illustrated in Figure 3.4. The Flight Management System (FMS) includes flight control computer (FCC), flight management computer

(FMC) and sensor package. While flight control computer is executing low level control processes with on-board sensors, flight management computer handles high level navigation processes. The Flight Control Computer (FCC) is a custom board (seen in Figure 3.3) with STM32F4 microcontroller which is based on 32bit ARM Cortex M4 floating point processor (168 MHz clock rate and has 192kB Ram).

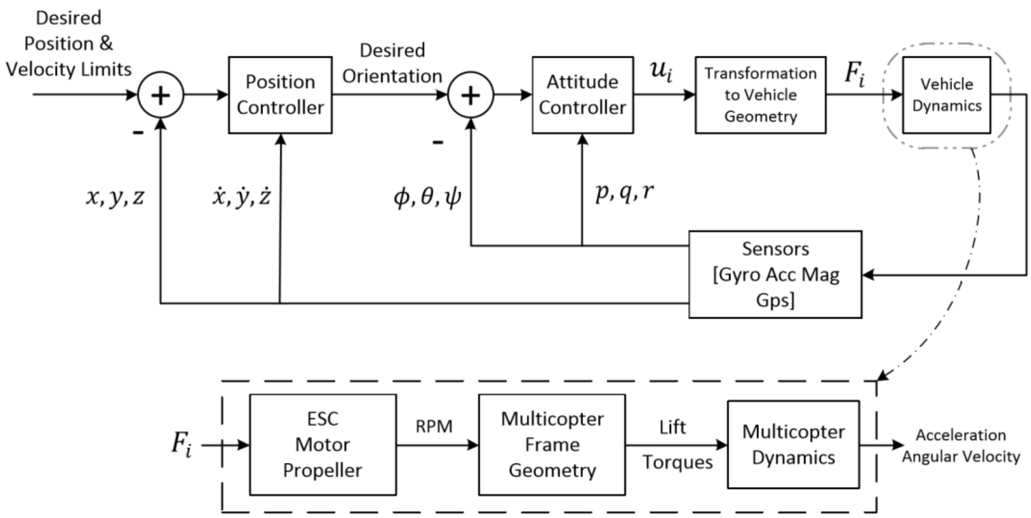

Figure 3.5: Position and attitude control loops in the Flight Control Computer

The FCC includes different analog and digital interfaces such as; serial inputs/outputs, PWM outputs, PPM encoders, CAN and ADC inputs/outputs. The serial interfaces such as RS232, UART, SPI and SDIO are used to communicate with IMU/INS/GPS, data logger and Flight Management Computer (FMC). Large number of PWM outputs enables to build a generic autopilot that suitable for most UAV concepts such as helicopter, plane, multicopter or even hybrid systems. Two channel PPM encoders are reserved to get inputs from traditional RC controller, where one of them is designed for master operator while the other is for trainee function. Moreover, the FCC board includes many ADC channels to interface with analog sensors such as pitot tube pressure sensor and alpha-beta sensors. The CAN interface, which is based on standard automotive communication protocol ensuring high electrical and electromagnetic noise immunity, has been reserved for an expansion port needs of additional sensors or payloads.

Flight Management Computer (FMC) executes high level guidance and navigation algorithms including aircraft intent language transitions. This computer is a Linux based Raspberry Pi which has 720MHz clock speed and 512MB ram. The aircraft intent (AI) output of the flight management computer transferred to the flight control computer through a serial interface. Telemetry and telecommand communication module is also linked to the flight management computer. The Xtend RF transponders (seen in Figure 3.7) operating at 900Mhz emulates ADS-B transponders for sense-and-avoid application. In addition to RF link, a wireless network with higherspeed but short range communication is utilised for remote debugging of the flight control computer through on-chip debugger abilities of the flight management computer.

For in-flight inertial measurements, Xsens Mti-G-700 Inertial Measurement Unit (IMU) is used. This IMU combines embedded accelerometer, gyroscope, magnetometer measurements (up to 100 Hz rate) for flight control purposes. In addition to this package, concept-specific sensors such as pitot tube, alpha-beta sensors has been equipped into FCC.

The Figure 3.5 demonstrates the control cycles of the autopilot system. As seen in the Figure 3.6: Flight/Aircraft intent planning data handling with command and control, and intent exchange procedures, the desired position and maximum velocity limits are generated by FMC, while other low level control loops run in flight control computer. The FCC evaluates required actuator signals to steer the aircraft. Position and Attitude Controllers are mainly based on cascaded Proportional and Integral (PI) controllers with washout filter. For position control, the outer PI controller loop derives the desired velocity for the Position Controller, while the washout filtered inner PI holds the aircraft at desired velocity. The same applies to attitude controller. The outer PI controller derives desired angular rates while the second PI with washout filter keeps the vehicle in the desired angular rate to achieve controlled turns. Washout filtered PI controller limits the velocity control signals within the desirable region. Otherwise, growing velocity generating errors as time passes cause undesirable control signal biases. On the other words, washout filter provides more smooth and stable flights. The control signals generated by the Attitude Controller drives another block consisting transformation matrixs associated with vehicle actuator geometry. The transformed signals are then conveyed to the actuators which are each driven by electronic speed controllers.

A portable ground station with graphical user interfaces (GUI) enables the operator to manage and monitor highlevel flight operations through 900 MHz RF and wireless network links. RF modems are pre-programmed for working in broadcast mode to communicate with all transponders in the field. For test and validation purposes, wireless link is used for data-link experiments based on intent language; and

RF links are integrated as enabling ADS-B implementations.

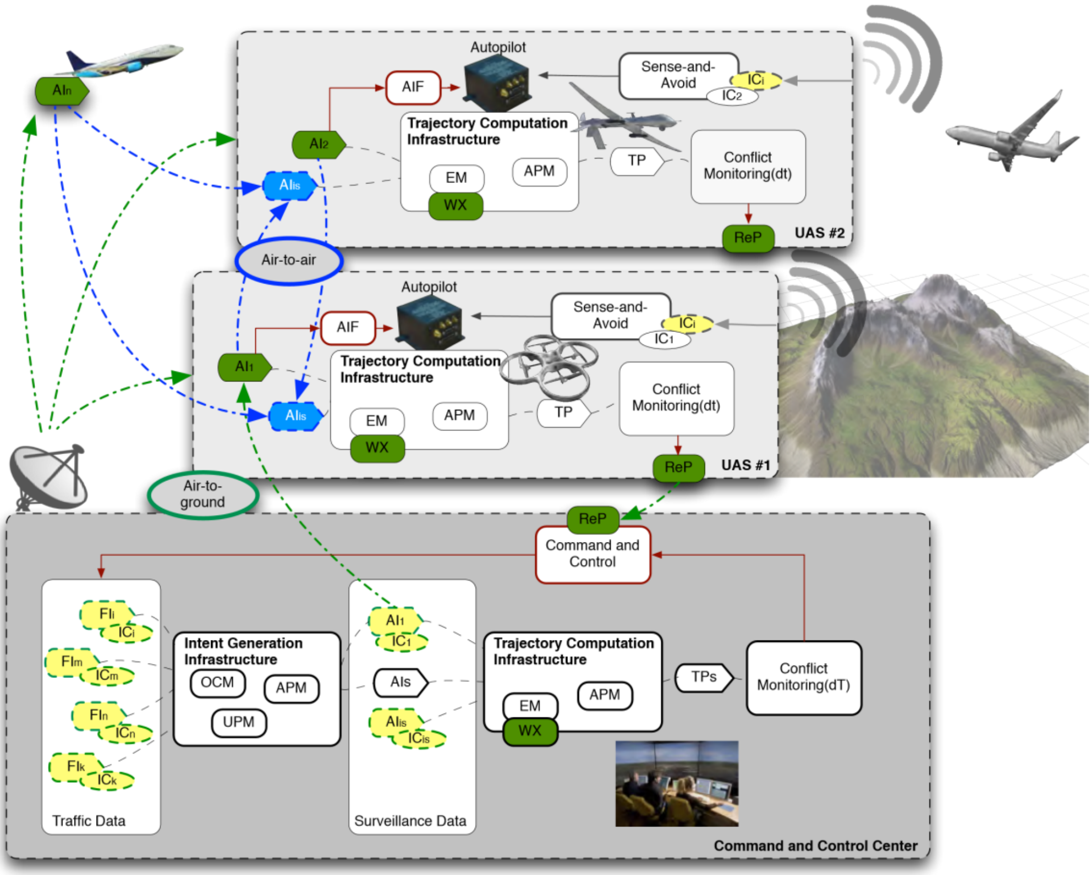

Figure 3.6: Flight/Aircraft intent planning data handling with command and control, and intent exchange procedures

Graphical User interface consists of two separated screens. The primary flight display provides with real-time video and flight data such as orientation, battery status, navigation accuracy, altitude, speed, control inputs, and telemetry status. The operator can choose to monitor any information at the required level. The GUI also includes a second screen demonstrating an operational map overlay which enables to view vehicles' location as shown in Figure 3.2. The vehicles which are equipped with transponders can also be located on the operational map of operator GUI. Using this screen, the operator can create a flight intent sequence (similar to the flight plan) or modify the existing one on-the-fly. Through the data-link, the operator can update flight intent sequence for evolving operational needs. The details of this process are given in the next section. In addition to real-time camera broadcast, the GUI also provides with synthetic vision suite (seen in Figure 3.8) using synthetic 3D map and earth terrain model. This add-on mode enables operating the vehicle in low visibility conditions such as foggy weathers and even night flights.

Figure 3.7: XTend 900MHz transponder for hardware emulator of ADS-B

3.3 Intent Based Nominal Operation and Trajectory Planning

Formal intent based planning module is envisioned to integrate effective command and control functionalities, and efficient intent data sharing capability into the unmanned aerial systems through a standardised intent language. This module utilizes two-level formal description languages such as flight intent and aircraft intent. One of the pretty mature formal intent language set, Flight Intent Description Language (FIDL) and Aircraft Intent Description Language(AIDL) are developed by Boeing Research and Technology Europe in order to efficiently synchronise the trajectories between the trajectory planners [26]. In our experimental testbed, a similar but simplified version has been implemented into command and control structure of unmanned aerial vehicle for technological demonstration purposes. These type of languages enable to define an action sequence of the aircraft dynamics (aircraft intent) or a sequence of the flight plan (fligh/mission intent) with different levels of detail, fully or partially specifying some aspects of the aircraft motion and leaving others open for later optimization/specification/planning considering the constraints and the objectives.

The Aircraft Intent (AI) language is a low level formal description employed to model the basic commands, guidance modes or control strategies for managing the aircraft on autopilot level. The AI instructions basically fill each degree of freedom of the mathematical description model describing the aircraft motion. The instructions set including different primitive modes of operation that an aircraft may employ has been derived from a detailed analysis and simulations. Any valid combination of predefined instructions (e.g. Hold Cruise Speed, Hold Altitude etc.) with their specifiers and execution intervals (bounded with end-triggers) describes the motion control objectives of the aircraft which are accepted by FMC of the UAV. The AI language grammar is subjected to set of lexical and syntactical rules in order to create a valid sequence.

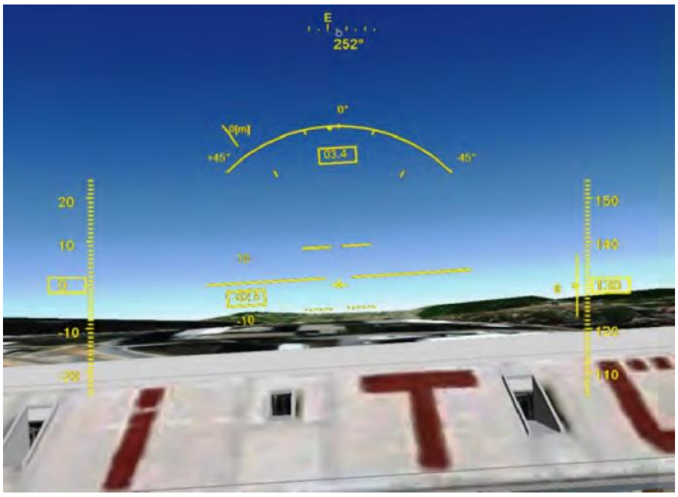

Figure 3.8: Synthetic vision screen capture from a real flight

The higher level language, Flight Intent (FI) language, is seen as a approximate mission plan of the UAV where the details to be satisfied by the limitations and objectives. A FI sentence provides a high level direction (flight segments or composite AI templates) on how a flight will be operated, and includes operationspecific constraints and objectives. In general, the flight intent does not determine a unique trajectory. A basic example for FI instruction is given in Figure 3.6, where flight segment primitives defines certain waypoint sequence tracking with their constraints and objectives associated with airspace rules and operational preferences. Flight Segment (FS) instructions may also include additional details about the lower level operation of flight if some aspects of the aircraft behaviour are defined. These are represented by the composites which are the template representation of a set of AI compositions such as Level Flight, Descent, Level Thrust Deceleration.

The Figure 3.6 demonstrates whole data handling processfor theintent based command and control and intent data sharing through the air-to-ground and air-to-air data links respectively. In this structure, mission interpretation and management procedures are handled through FI language that is the higher level language enabling the human operator (operation/mission manager) to easily manage, interpret and modify. The ground based Conflict Monitoring and Command and Control blocks represent management functions including all autonomous and decision support tools for managing flight operation at tactical level. The Intent Generation Infrastructure is a tool introduced in [28] including Aircraft Performance Model (APM) and a pair of databases, one storing a User Preferences Model (UPM) and one storing an Operational Context Model (OCM). The UPM involves the preferred operational strategies directing the aircraft such as the preferences of a mission manager [28]. The OCM involves standard constraints on the use of airspace. The Intent Generation Infrastructure accepts a FI sentence (including flight segments, constraints and objectives), and Initial State (IC) as inputs; then processes with UPM, OCM, APM in order to translate into a compatible AI sentence.

The peripherals of the aircraft FMS also include functionalities enabling similar capabilities of Trajectory Generation Infrastructure for trajectory planning and intent based control handling. In addition to routine automated data exchange, any intervention (ReP; replanning request) can be initiated through air-to-grounddata-link when it is needed. The *Conflict Monitoring* functions in both air and ground segment monitor potential loss of separation situation within the prescribed time interval through predicted trajectories (TPs). These trajectories may also include uncertainty factors in a set of parameters (e.g. in aircraft performance, position, weather etc.) and their" what-if" extensions (e.g. considering unexpected behaviours) in a probabilistic manner. The *Command and Control* function operates this intervention from the ground by attaching new constraints or objectives to the pre-planned FIsequencewhen it requires. The textitTrajectory Generation Infrastructure translates updated FI sequence into AI sequence and broadcast to the aircraft.

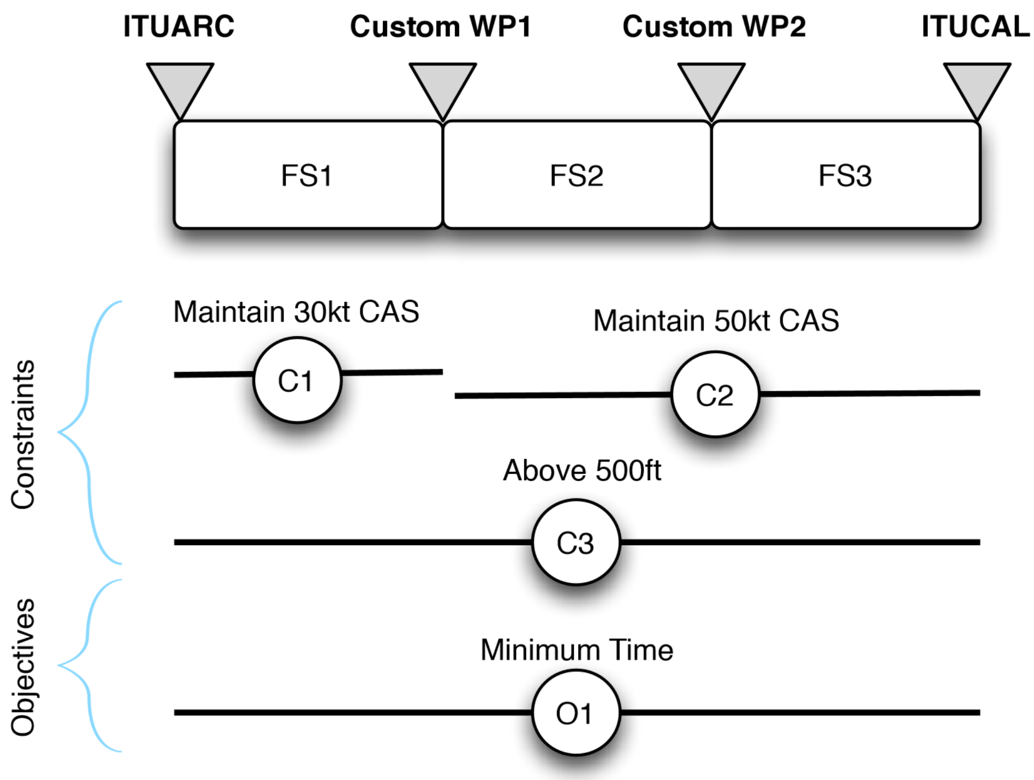

Figure 3.9: Example FIDL instance with flight segments, constrains and objectives

The aircraft can also request re-planning (ReP) when the on-board *Conflict Monitoring* detects apotential conflict. The AI data sharing is the low-level" machine-to-machine" communication where the autopilot of the UAV can fully understand and execute through the *Automated Intent Flight – AIF*. Similarly, air-to-air intent data exchange procedure is also handled through AI language. In this case, the on-board *Conflict Monitoring* blockmonitors the potential conflicts between the predicted trajectories of the aircraft in the surveillance traffic through the *Trajectory Computation Infrastructure*.

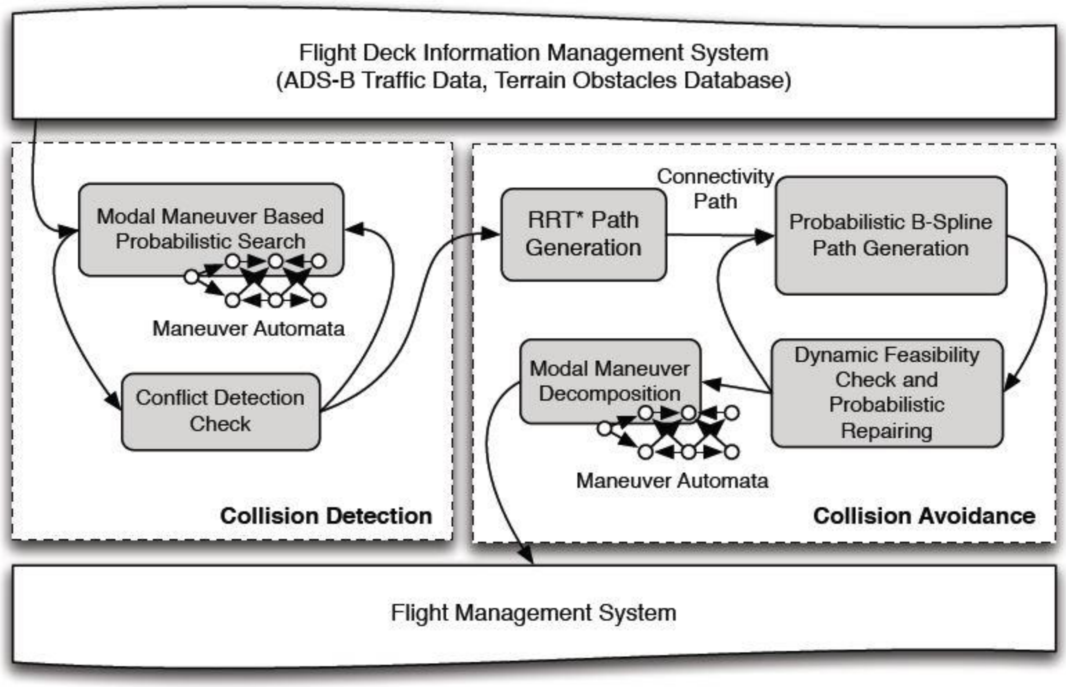

Figure 3.10: Fundamental architecture of Sense-and-Avoid algorithm

The *Trajectory Computation Infrastructure* translates an AIDL sentence into unique predicted trajectory (TP). In this level, it is expected that different trajectory computation tools would result in the same predicted trajectory if they use the same inputs and models such as a) aircraft intent (AIDL) b) Initial conditions (IC) (aircraft state at the initial position and environmental condition at this altitude), c) Aircraft Performance Model (APM), d) Environmental Model (EM), and e) similar trajectory computation algorithms. Event though these premises may not be unattainable in practice, sharing the aircraft intent significantly contributes to achieve partial trajectory synchronization [29].

3.4 Autonomous Sense-And-Avoid

The Sense and Avoid module is a decentralised independent safety assurance system in the unmanned system for immediate to short-term collisions with aircraft and terrain objects. This module does not require intent sharing or time-consuming negotiation process, and immediately intervenes when the mid-term separation assurance process fails. Envisioned approach involves recent algorithmic advances based on probabilistic models of aircraft behaviour and information uncertainty in order to improve existing logics of the collision avoidance system.

The sense and avoid module of the UAV uses two types of information; surrounding traffic information and terrain database. *Terrain Obstacle Database* stores spatial model of the earth objects with their locations and heights in certain resolution. The traffic information is obtained from ADS-B (Automatic Dependent Surveillance-Broadcast) transponders of surrounding aircraft. This module enables both ADS-In and ADS-B Out applications where allow data transmission between aircraft themselves and ground segments respectively. This data including a set of GPS-derived states of the aircraft is automatically broadcasted and received through equipped transponder emulators. The hardware emulators of the ADS-B transponders which allow multi-vehicle data communication can be seen in Figure 3.7. For simplification of the implementation, the experimental ADS-B transponders always use exact same operation mode and simplified data structure which is illustrated in Figure 3.11.

The collision detection algorithm is based on the idea of spatial search phenomena for potential collisions with aircraft and terrain obstacles. This search method relies on the creatingof probabilistic flight trajectory envelopes (for constant time windows) for every aircraft in the surrounding traffic. These envelopes also include uncertainty factors due to the uncertainty in measured position, weather effect and performance models. Trajectory envelope generating process hinges on using multi-modal approach utilising distinct flight maneuver modes can be performed in the short-term domain. This multi-model approach strongly connected with the concept of hybrid systems. The algorithm generates bundle of probabilistic action patterns by sampling finite maneuver mode set and their parameter domains respectively [27]. Through this probabilistic search, this random sampling inherently embeds the stochastic nature of the rational or irrational behaviour of the aircraft (managed by human operator or machine). The Figure 3.12 was captured during *Collision Detection and Resolution* (CD&R) algorithm running. The Conflict Detection system recursively computes and observes the probability of collisions, and delays to issuing alert until the conflict probability exceeds the predefined thresholds.

The collision avoidance method based on closed-loop planning where theprocess generates an action sequence that minimizes cost by accounting future actions, and update likelihoods upon the new information availability. The algorithm hinges on solving first relaxed forms of the problem and then gradually refining it using the previous approximate solutions. This simplification enables the process to obtain a real-time solutionfor required response maneuver in the order of seconds. In the first step, the algorithm rapidly explores the airspace with a modified version of Rapidly Exploring Random Tree (RRT*) algorithm [30], which generates approximate conflict-free route (ensuring asymptotic optimality). In the second step, this approximate path (including recovery to the original track) is enhanced with the Probabilistic B-Spline algorithm [27] in order to create smoother path. The generated path is further iteratively verified for collision and dynamic feasibility considering dynamic performance model of the aircraft. After obtaining the feasible flight trajectory with executable velocity and acceleration sequence; maneuver decomposition algorithm readily decomposes the flight trajectory into a feasible sequence of maneuver modes primitives and evaluates their flight-specific parameters. The overall functional architecture of the algorithm can be seen in Figure 3.10. The details of this algorithmic phenomena had been first introduced in the previous work of the authors [27].

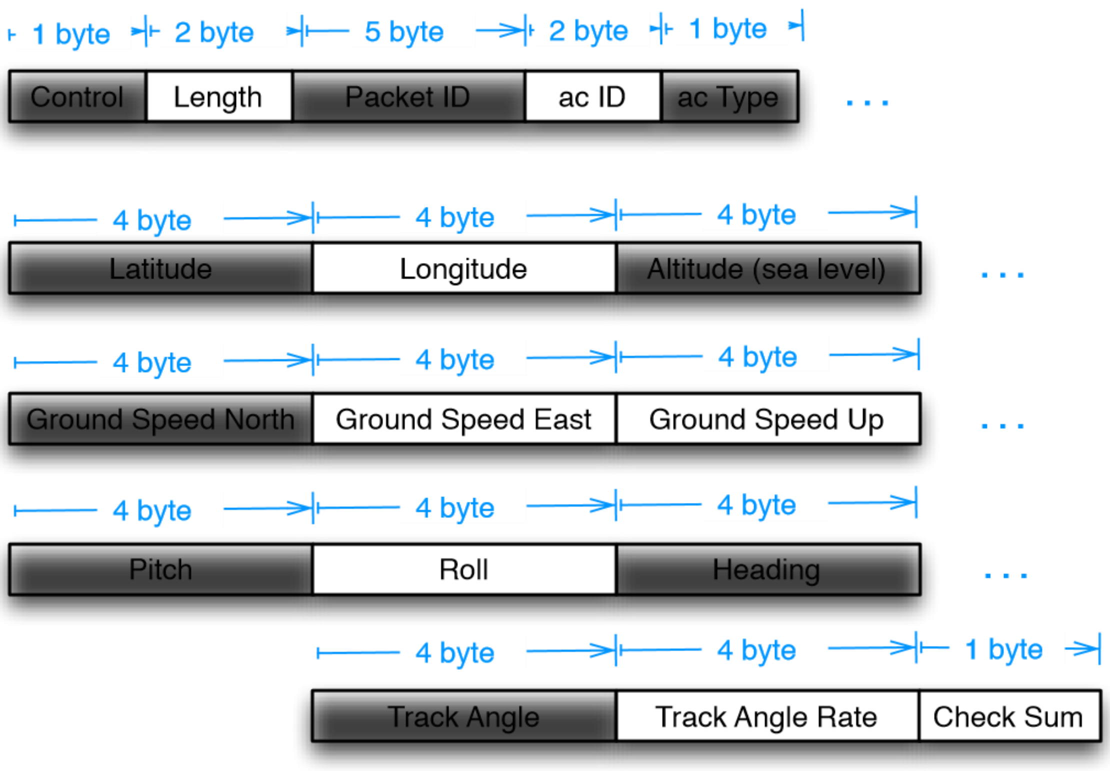

Figure 3.11: Simplified ADS-B data structure for traffic information sharing

The generated modal sequence is then translated into navigation inputs to provide the Flight Management System (Figure 3.6 FMS) for avoidance maneuver (with its recovery) implementation (as seen in the Figure 3.6). These avoidance maneuvers including recovery generate small deviations where their impacts on the entire flight route is minimised.

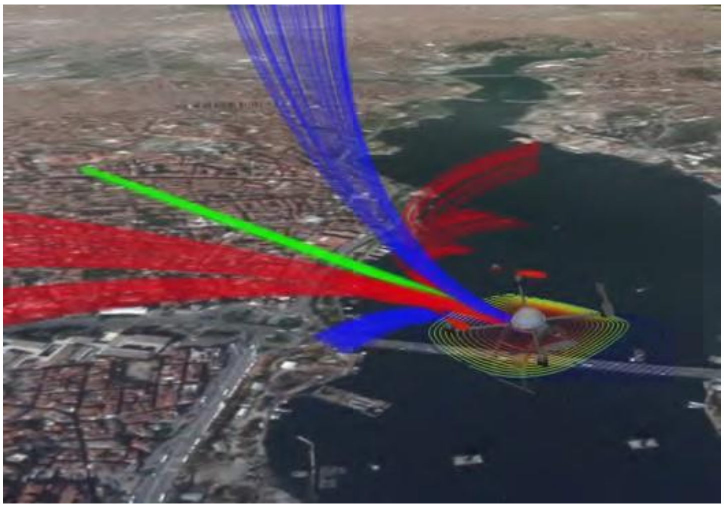

Figure 3.12: Running of sense-and-avoid algorithm with probabilistic modal maneuver search

3.5 Test Results

The system was demonstrated in three different scenarios. In the first, a single intruder was involved (seen in Figure 3.13). A potential collision was detected, the onboard FMS requested re-planning from the ground operator, but the operator did not respond immediately. The Collision Avoidance system generated its own 4D evasive solution right before the collision and executed it. Then the operation proceeds without a conflict.

In the meantime, intruder UAV changes its own heading and starts to go that direction (see in Figure 3.14). This situation can cause dificulties for avoidance algorithms. In this situation, collision detection algorithm is still controlling if there is any conflict or not. Therefore, it finds another conflict in some point and generate another non conflict optimal path.

Another picture is the demonstration of multiple UAV intruder case (see in Figure 3.15) . For these cases, algorithm’s computation time might be higher than one intruder case scenarios. Even for three intruder, collision detection algorithm and also collision avoidance algorithm finds optimal path for UAV.

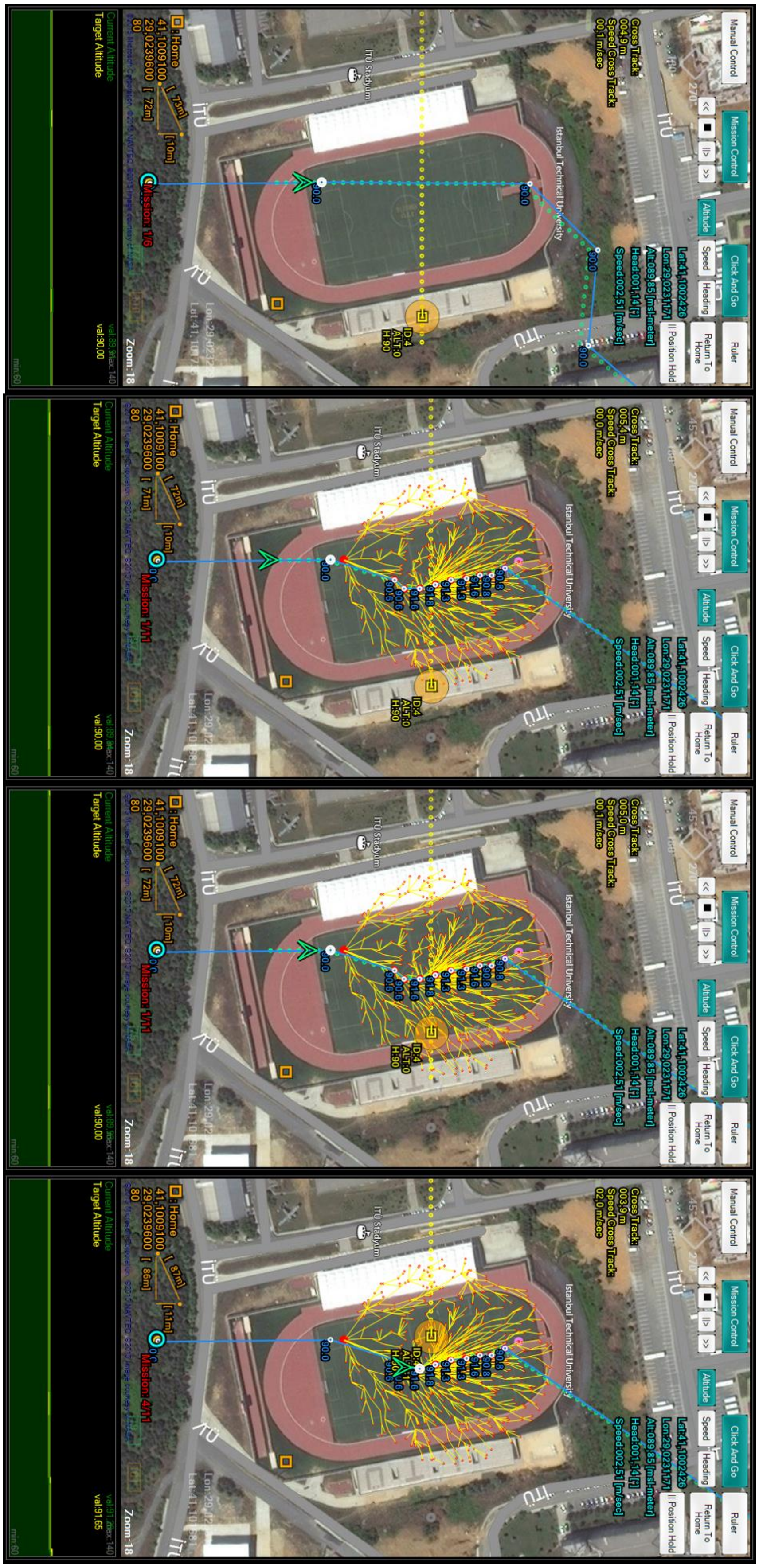

Figure 3.13: One intruder case

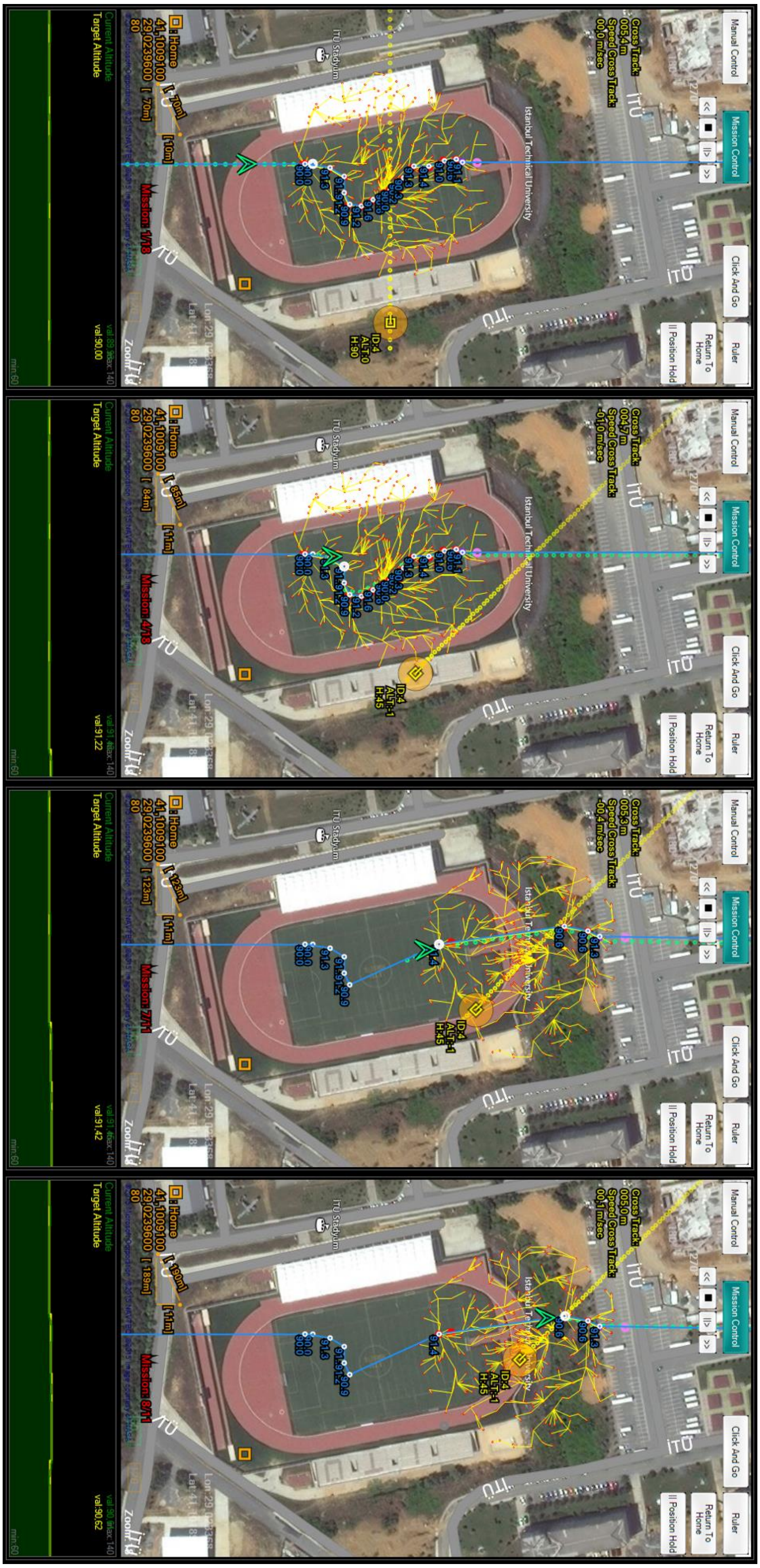

Figure 3.14: Intruder heading change case

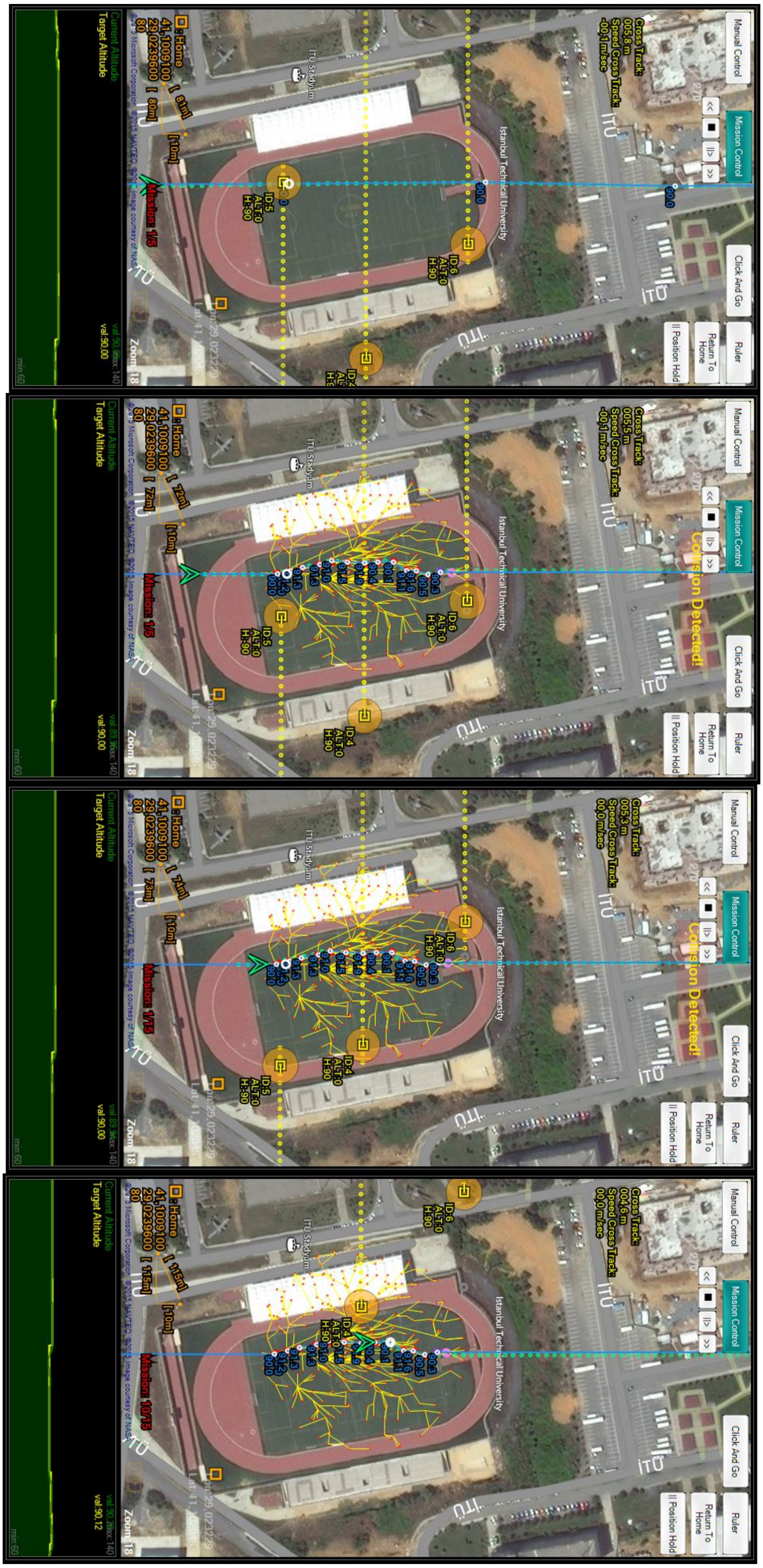

Figure 3.15: Three intruder case

For the results of RRT-Star algorithm, it is very significant that sample number is direcly effect the path's optimality. On the other hand, higher sample number means more computation time for solve the conflict. To demonstrate this, one intruder and three intruder case is demonstrated 30 times for each different sample number. And the result is shown in Figure 3.16 and Figure 3.17.

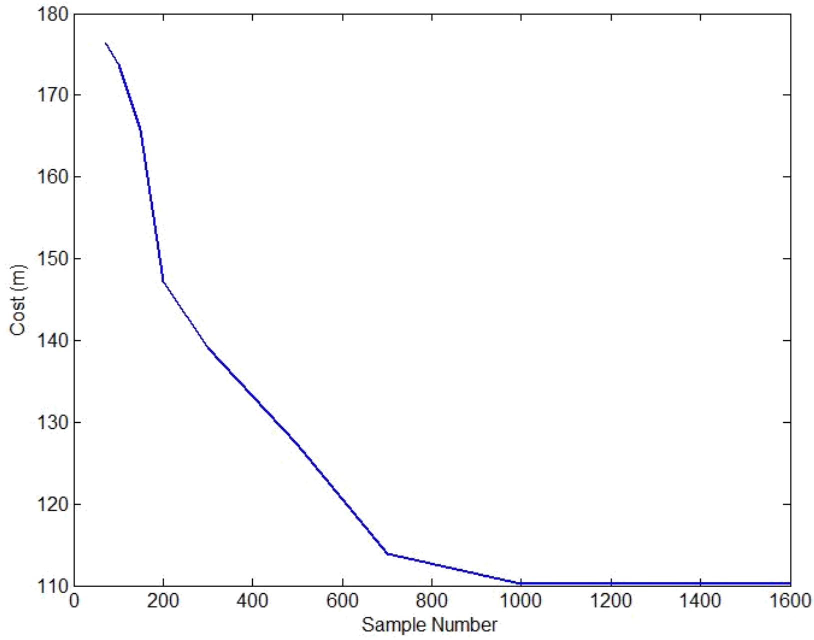

Figure 3.16: One intruder case

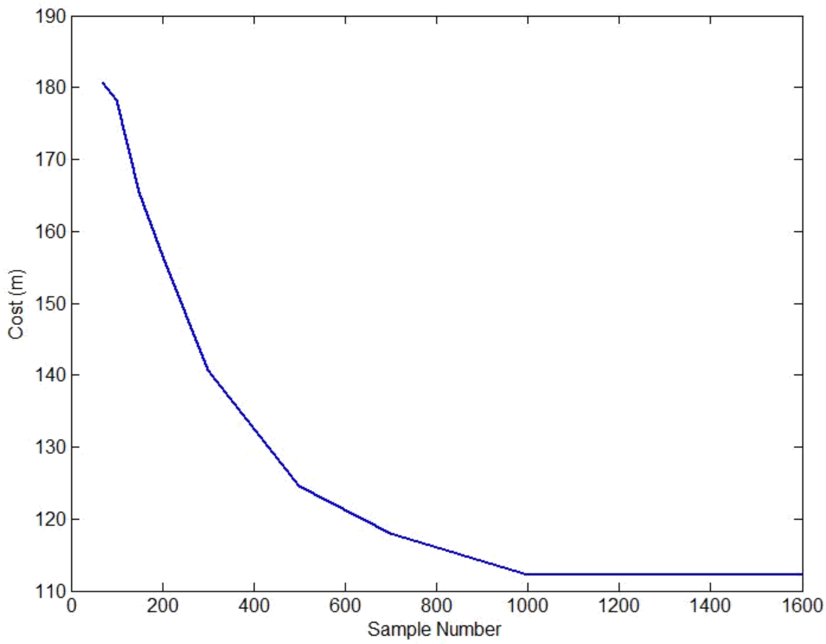

Figure 3.17: Three intruder case

For both cases, after a specific sample number, found path's cost is become the same. This shows that the algorithm can find optimal path for certain sample number. This situation is also the same for three intruder case. However, it is expected that for three intruder case optimal path cost is bigger than one intruder case. Therefore, we can say that cost and intruder number are proportional.

When sample number is big enough, found path's cost is become minimum it can be. But for that big sample number have disadvantage. Also computation time and sample number are proportional. Because that much sample time means that much loop and control for non conflict paths generation. So, it is crucial to decide sample number. If system have enough computation power, then bigger sample numbers can be choosen. But if computation power not enough, this means that maybe solution take too long to avoid collision. So, test of this situation is made and results is shown in Figure 3.18 and Figure 3.19.

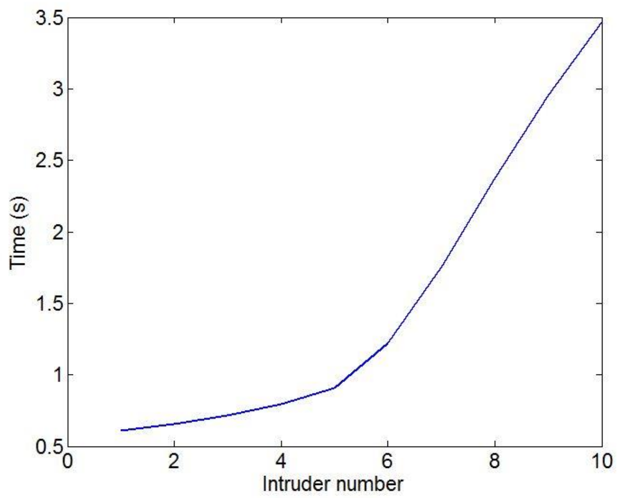

Figure 3.18: Computation time - Intruder number

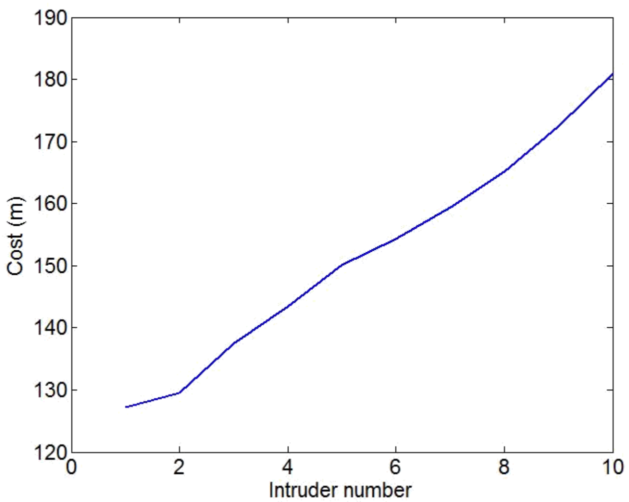

Figure 3.19: Cost - Intruder number

4. CONCLUSİON AND FUTURE WORK

In this thesis, there is two major design is introduced for UAV. First one is INS and AHRS design. This design is very significant for autonomous flight. Because of lack of INS and AHRS products for multi-copter dynamics is the main issue for this design. Second one is flight management system with collision avoidance for UAV. In next 20 years, civil aviation and also multi-copter type UAV's will grow too much that in time, these systems become uncontrollable. This situation may cause some major difficulties and also might become a reason of a lot of damage through people, environment, buildings or will be endanger to each other. Therefore, common communication technology is proposed for both aerial vehicles and ground stations and ATCs. With this change, authorities can surveillance all vehicles from one place. For hardware change from all vehicles and stations are difficult but it is necessary for solve the main problem. But it is also crucial that communication frames should be detailed, because of bandwidths and also need for vehicles and stations. So, there is aircraft intent based frames and fligt intent based frames are recommended. For collision avoidance algorithm, RRT-Star algorithm is chosen. The main reason of choosing this algorithm is being probabilistic. In 3D, deterministic algorithms take too long, and also do not give optimal path solutions. But in RRT-Star, solution converges to optimal path for infinite sample number. So, integration of RRT-Star algorithm is made and for test, hardware in the loop simulation is done. In simulation, testbed, avionics and simulation programs is introduced. For testing communication, ADS-B emulator is designed. This emulator broadcastly transmit specific information which is given. As a result of this, hardware in the loop tests results and also some major situations are given.

For INS and AHRS design, in future, some other filters which is particle filter and extended kalman filter can be tried to improve accuracy of INS and AHRS outputs. Also, not with these hardwares but also vision based navigation or lidar can be integrated to this system. In some non-gps situations, these hardwares can help to continue the navigation task. Moreover, with more accurate INS and AHRS design, multi-uav operations and path planning algorithms can be studied in outdoor. These systems will be used in Campus UAV project and also in IHATAR projects which is mentioned in introduction.

The represented system design and also demonstration is published at International Conference of Unmanned Aerial Systems (ICUAS) in 2014 [31]. The system presented here is based on the projection that many more and increasingly non-standardized types of aerial systems will be integrated into the national airspace. Only ground systems will be able to store large parametric databases of all aircraft models. In this case, only ground systems will be able to predict future trajectories, while airborne systems must frequently share information with accuracy metrics. For real-flight demonstrations, an experimental FMS with autonomous collision avoidance algorithms and low-level control system was deployed on a complete aerial testbed. A Command and Control (C2) station was built. This paper presented the system architecture and design for modeling, control, and the ground station. Performance and demonstration tests were described, which focused on the hardware, so are clearly not adequate for comprehensive real-world validation. For the sake of simplicity, we kept complexity low and used fixed-speed intruders whose actual flight trajectories almost perfectly aligned with the predicted trajectories. As for future development, we are planning to add high-level parametric (e.g., positional inaccuracies, data-link losses) and behavioral uncertainties (e.g., operator errors, faulty commands, low-visibility, emergencies) into experiments. This will make it possible to properly validate for the complete system involving both humans and machines.

REFERENCES

[1] Dalamagkidis, K., Valavanis, K.P. and Piegl, L.A. (2008) '*On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the national Airspace system*', *Progress in Aerospace Sciences*, 44(7-8), pp. 503–519. doi: 10.1016/j.paerosci.2008.08.001.

[2] Zaloga, S.J., Rockwell, D., Finnegan, P.: *World Unmanned Aerial Vehicle Systems, Market Profile and Forecast* (2011 edn). Tech. rep., Teal Group Co. Fairfax (2011)

[3] *UAV Drones Market by Type (Fixed Wing, Rotary Blade, Nano, Hybrid), Application (Law Enforcement, Precision Agriculture, Media and Entertainment, Retail), & Geography (Americas, Europe, APAC, RoW)* - Analysis & Forecast to 2020

[4] Novatel, *APN-064 Rev A IMU Errors and Their Effects*, Feb 2014, White Paper

[5] Michael Barrett, *Analyzing And Modeling Low-cost Mems Imus For Use In An Inertial Navigation System*, MsC Thesis, 2014

[6] National Instruments, *Sensor Terminology*, 2013, White paper

[7] Beard, R.W., McLain, T.W. and Beard, al W. (2012) *Small unmanned aircraft: Theory and practice*. United States: Princeton University Press.

[8] J.-M. Zogg, GPS: Essentials of Satellite Navigation. http://zogg-jm.ch/Dateien/GPS_Compendium(GPS-x-02007).pdf, u-blox AG, 2009.

[9] Grewal, Mohinder S.; Weill, Lawrence Randolph; Andrews, Angus P. (2001). *Global positioning systems, inertial navigation, and integration*. John Wiley and Sons.ISBN 978-0-47135-032-3.

[10] Parkinson; Spilker (1996). *The global positioning system. American Institute of Aeronautics & Astronomy*. ISBN 978-1-56347-106-3.

[11] Mechtly, E. A., 1973: The *International System of Units, Physical Constants and Conversion Factors. NASA SP-7012*, Second Revision, National Aeronautics and Space Administration, Washington, D.C.

[12] I. Lugo-Cardenas, G. Flores, S. Salazar, and R. Lozano, "*Dubins path generation for a fixed wing UAV*," 2014 International Conference on Unmanned Aircraft Systems (ICUAS), May 2014.

[13] Honeywell, Inc. Applications of Magnetoresistive Sensors in Navigation Systems http://www.ssec.honeywell.com/position-sensors/datasheets/sae.pdf

[14] Honeywell, Inc. Applications of Magnetic Sensors for Low Cost Compass Systems http://www.ssec.honeywell.com/magnetic/datasheets/lowcost.pdf

[15] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global positioning systems, inertial navigation, and integration*. New York, NY, United States: Wiley-Interscience, 2002.

[16] K. P. Valavanis, Ed., *Advances in unmanned aerial vehicles: State of the art and the road to autonomy (intelligent systems, control and automation: Science and engineering) ... And automation: Science and engineering*). New York, NY, United States: Springer-Verlag New York, 2007.

[17] 112th Congress, "*Faa modernisation and reform act of 2012*," Conference Report H.R.658, February 1, 2012.

[18] F. A. Administration, "*Integration of civil unmanned aircraft systems (uas) in the national airspace system (nas) roadmap*," No. 2012-AJG502, 2013.

[19] T. Koo, B. Sinopoli, A. Sangiovanni-Vincentelli, and S. Sastry, "*A formal approach to reactive system design: unmanned aerial vehicle flight management system design example*," in Computer Aided Control System Design, 1999. Proceedings of the 1999 IEEE International Symposium on, 1999, pp. 522–527.

[20] J. Krozel and D. Andrisani, "*Intent Inference and Strategic Path Prediction*," in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Jan. 2005.

[21] J. L. Yepes, I. Hwang, and M. Rotea, "*An Intent-Based Trajectory Prediction Algorithm for Air Traffic Control*," in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Jan. 2005.

[22] E. Koyuncu and G. Inalhan, "*Exploiting delayed and imperfect information for generating approximate uav target interception strategy*," Journal of Intelligent and Robotic Systems, vol. 69, no. 1-4, pp. 313–329, 2013. [Online]. Available: http://dx.doi.org/10. 1007/s10846-012-9693-6

[23] I. Hwang and C. E. Seah, "*Intent-based probabilistic conflict detection for the next generation air transportation system*," in Proceedings of the IEEE, 2008, pp. 2040–2059.

[24] J. Planning and D. Office, "*Concept of operations for the next generation air transportation system ver. 3.2*," JPDO Report, 2007.

[25] EUROCONTROL, "*The concept of operations at a glance*," SESAR Definition Phase Report, 2008.

[26] J. Besada, G. Frontera, J. Crespo, E. Casado, and J. Lopez-Leones, "*Automated aircraft trajectory prediction based on formal intentrelated language processing*," Intelligent Transportation Systems, IEEE Transactions on, vol. 14, no. 3, pp. 1067–1082, 2013.

[27] E. Koyuncu, N. K. Ure, and G. Inalhan, "*Integration of path/maneuver planning in complex environments for agile maneuvering ucavs*," Journal of Intelligent and Robotics Systems, 2003, doi: 10.1007/s10846009-9367-1.

[28] L. L. Javier and V. R. Miguel, "*Providing data for predicting aircraft trajectory*," Patent EP20110382020 20110128, 08 01, 2012.

[29] J. Lopez, M. Vilaplana, I. Bayraktutar, J. Klooster, J. Asensio, G. McDonald, and P. Kappertz, "*Towards an open test bed for the study of trajectory synchronization in the future atm system: The asis initiative*," in Integrated Communications, Navigation and Surveillance Conference, 2009. ICNS '09., 2009, pp. 1–14.

[30] S. Karaman and E. Frazzoli, "*Incremental Sampling-based Algorithms for Optimal Motion Planning*," Robotics Science and Systems, 2010.

[31] Tarhan, A.F., Koyuncu, E., Hasanzade, M., Ozdemir, U. and Inalhan, G. (2014) '*Formal intent based flight management system design for unmanned aerial vehicles'*, 2014 International Conference on Unmanned Aircraft Systems (ICUAS). doi: 10.1109/icuas.2014.6842349.

[32] Tarhan, A.F. (2014). *Modelling, Control and Implementation of an Unmanned Vertical Teke-off and landing Aircraft* (MsC Thesis). Istanbul Technical University, Graduate Shool of Sience Engineering and Technology, Istanbul

CURRICULUM VITAE

Name Surname **:** Mehmet Hasanzade

Place and Date of Birth **:** Altındağ – 22.02.1992

E-Mail **:** hasanzade.mehmet@gmail.com

EDUCATION **:**

- **B.Sc.** **:** 2014, Istanbul Technical University, Fakulty of Electrical and Electronic Engineering, Telecommunication Engineering
- **M.Sc.** **:** 2016, Istanbul Technical University, Fakulty of Electrical and Electronic Engineering, Control and Automation Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- PAVO Elektronik San.Tic.A.Ş., intern (2010)
- ITU Solar Car Team – Embedded System Hardware Designer (2011 – 2013)
- TubitakG Solar Car Race – Third Place (2012)
- TubirakG Solar Car Race – Best Design (2012)
- ITU Control and Avionics Laboratory, intern (2012)
- ITU Control and Avionics Laboratory, Project Assistant (2012 - …)

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Tarhan, A.F., Koyuncu, E., Hasanzade, M., Ozdemir, U. and Inalhan, G. (2014) 'Formal intent based flight management system design for unmanned aerial vehicles', 2014 International Conference on Unmanned Aircraft Systems (ICUAS). doi: 10.1109/icuas.2014.6842349.