TALLINN UNIVERSITY OF TECHNOLOGY

Faculty of Information Technology

Department of Software Science

ITC70LT

Safak Tarazan 156348 IVCM

DEAD RECKONING IMPLEMENTATION ON MINI UAV TO MITIGATE GNSS SIGNAL SPOOFING-JAMMING ATTACKS

Master thesis

Truls T. Ringkjob

Master's Degree in Cyber Security

Lecturer IT College, Visiting Lecturer TTÜ

Tallinn 2017

Author’s declaration of originality

Author’s declaration of originality is an essential and compulsory part of every thesis. It always follows the title page. The statement of author’s declaration of originality is presented as follows:

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Safak Tarazan

24.05.2017

Abstract

UAV (Unmanned Aviation Vehicle) relies on satellite navigation systems to determine its location. However, latest researches have shown that it is fairly easy to jam or spoof the satellite signal. There have been real cases where UAV was landed by enemy' forces by jamming and spoofing the satellite signal. Inevitability, this threatens the CIA (Confidentiality, Integrity and Availability) of the UAV as well as the people's security. Fortunately, there are various methods to detect satellite signal spoofing and jamming attacks. However, there is a gap between detection of these kinds of attacks and reacting to it. In this manner, most of the mini UAVs are configured to land over there, hang in the air or take no action when they detect satellite signal spoofing or jamming attack. Thus, lacking post incident activities in case of this kind of attacks still enables the success of the attack and this still threatens the CIA of the UAVs.

This paper proposes a way to send the UAV back to its home or its secure base without relying on a satellite signal, when UAV detects a satellite signal spoofing and jamming attack, by implementing dead reckoning system on the UAV. Additionally, proposed solution is converted into a Python code, code is tested against simulated UAV, and various tests are conducted.

This thesis is written in English and is 54 pages long, including 5 chapters, 14 figures and 17 tables.

Annotations

Pimesi rakendamine mini UAV leevendamise GNSS Signal tüssamine summutamise Rünnakute

UAV (Unmanned Aviation Vehicle) ehk mehitamata õhusõiduki navigatsiooni süsteemid baseeruvad satelliitsideühendusel asukoha määramisel. Hiljutised uurimused on aga näidanud, et sellist satelliidi signaali on võrdlemisi lihtne nii segada kui võltsida. Veelgi enam, praeguseks on olnud juba juhtumeid, kus vastaspool on maandanud mehitamata õhusõiduki kasutades selleks satelliidi signaali segamist ning oma signaaliga asendamist. Paratamatult ohustab see nii UAV-de CIA printsiipi (Confidentiality, Integrity and Availability ehk Konfidentsiaalsuse, Terviklikkuse ja Kättesaadavuse põhimõtet) kui ka inimeste julgeolekut. Õnneks on olemas mitmeid erinevaid meetodeid signaali segamisel ja asendamisel põhinevate rünnakute märkamiseks. Paraku valitseb selliste rünnakute märkamise ja neile reageerimise vahel ajavahemik. Sellest tulenevalt on enamik mini-UAV-sid seadistatud kas kohapeal maanduma, õhus püsima ühe koha peal või üldse mitte reageerima kui on märgatud satelliitside segamist ja võltsimist rakendav rünnak. Korralike reaktsioonide puudumise tõttu sellist tüüpi rünnakutele on seega enamike mehitamata õhusõidukite CIA printsiip haavatav.

Käesolev magistritöö pakub välja alternatiivse meetodi, kuidas saata mehitamata õhusõiduk ehk UAV tagasi koju ilma satelliitsidet kasutamata. Kui UAV tunnetab satelliidi signaali segamist rakendaks ta orienteerumiseks pimenavigatsiooni süsteemi, mis on mehitamata õhusõidukisse selleks puhuks sisse ehitatud. Välja pakutav pimenavigatsiooni süsteem on kirjutatud Python'is ning koodi on katsetatud simulatsioonis erinevate testidega.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 54 leheküljel, 5 peatükki, 15 joonist, 17 tabelit.

Table of abbreviations and terms

APL	Applied Physics Laboratory
CAA	Civil Aviation Authority
CCS	Command and Control System
CIA	Confidentiality, integrity and availability
CLI	Command Line
DIY	Do It Yourself
DOF	Degree of Freedom
DR	Dead Reckoning
DoD	Department of Defense
ESC	Electronic Spin Controller
FAA	Federal Aviation Administration
FPV	First Person View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
HW	Hardware
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
ISR	Intelligence, Surveillance and Reconnaissance
MCS	Master Control Station
NN	Neural Networks
OS	Operating System
PDR	Personal Dead Reckoning
PNR	Pseudo Random Code
QZSS	Quasi-Zenith Satellite System

RC	Radio Controller
SDK	Software Development Kit
SDR	Software Defined Radio
STIL	Software In the Loop
TCP	Transmission Control Protocol
UAS	Unmanned Aviation Systems
UAV	Unmanned Aviation Vehicle
UAVS	Unmanned Aircraft Vehicle System
UDP	User Datagram Protocol
USA	United States of America

Table of contents

List of figures

List of tables

Acknowledgements

I appreciate every single creature who tries to make this world a better place. I also would like to express my respect to myself for the things that I have accomplished in the past and current times. Additionally, I wish happiness, accuracy, precession and adaptability to all creatures.

1. Introduction

Recently, unmanned aviation vehicle, also known by UAV, usage has increased dramatically and they are being used in many sectors such as ISR as well as in commercial applications. The researchers forecast that UAS (unmanned aviation systems) - acronym defined by FAA - market will continue to grow in various sectors such as evaluating/managing, delivering and transportation as given in the Figure 1. When we look at the numbers, Teal Group expects that, in the next decade, UAS production will be tripled from $4 billion annually to $14 billion in global level [1].

This rapid movement and development in UAS market got cyber security researchers' attention. Thus, various studies were conducted which focuses on UAS. Since UAS relays on GNSS (global navigation satellite system) to determine its location, and satellite signal known to be susceptible to interference, GNSS security became one of the hottest topic in UAS navigation security.

Studies have shown that, it is fairly easy to compromise satellite signal which leads to undesired behaviors on UAS. Even, this can be achieved with small budgets as around 1000$ using SDR (software defined radio) [2]. In this regard, there have been several researches which are given in the following sections. However, when there is enough budget and resource, GNSS signal jamming or spoofing attacks can be performed against even to tactical/strategical level army UAS. United States' strategical level RQ-170 UAV, which was landed by Iranian arm by using GNSS signal jamming and spoofing techniques, is one of the recent real case of this [3].

Figure 1. Predicted UAV market grow timeline[1]

In this context, various studies were conducted and several GNSS signal spoofing detection methods were implemented. Table 1 briefs some of these detection mechanisms for GPS. Even some (i.e. U-blox, one of the commonly used GPS/GNSS receiver brand in micro UAS market [4], [5]) of the GPS receiver modules come with an embedded GPS/GNSS spoofing or jamming detection mechanisms [6],[7]. However, although in some cases GNSS signal spoofing or jamming detection mechanism comes out of the box, the post-incident activities are either very weak or not even exists in micro UAV. For instance, while UAVs by Parrot or 3DR companies have no countermeasures to react GNSS signal spoofing or jamming attacks, some of the DJI UAVs lands or hovers in the air in case of these kinds of attacks. [8]. In this manner, lacking post-incident activities during these attacks can be compromised and can be used to capture or to land the UAV. DroneDefender, which is portable long range GPS jammer, would be the best example of capturing the certain model of UAV by jamming the GPS signal [9].

Mini UAV are the devices used commercial applications and public area. Besides capturing these UAVs by GNSS spoofing or jamming, an attacker may intentionally or unintentionally harm the civilians. Moreover, even naturally disrupted GNSS signals may le[1]ad UAV to fly away and threaten the civilians and environments. To eliminate mini UAVs being captured by the attacker using the GNSS attacks mentioned in the

[1] Source: Timeline developed by CyPhy Works Inc. and reported in DroneLife.com, April 8, 2014, http://dronelife.com/2014/04/08/greiner-2015-the-year-of-the-protecting-and-inspecting-drone

previous sections, this paper proposes a solution where UAV can go back to home or a secure place under GNSS spoofing or jamming attack. Moreover, introduces a severity level assessment of GNSS attacks on UAV and actions to be taken according to these levels. Proposed solution also applies to circumstances when GNSS signal not available.

Table 1. GPS Spoofing detection methods summary[1]

Name	**Description**	**Encrypted**	**Networked or stand alone**	**Single or multi antenna**	**Layer**	**Receiver required capability**
Vestigial Signal Defense	Monitor the presence of legit vestige signals to distinguished more than one correlator peaks	No	Stand alone	Single	Signal proces sing	Filter out vestige legit GPS signals
Antenna-based joint attitude estimate n	Using the estimated DOA for discriminating between authentic GNSS and the spoofing signals.	No	Stand alone	Multi	Data bit	Miniaturized antenna array
Monitor the relative GPS signal strength	Monitor the abrupt changes in signal power that happen during spoofing attacks	No	Stand alone	Single	Signal proces sing	Signal strength monitoring
Check the time intervals	Check the time intervals between the acquisition of each satellite signal	No	Stand alone	Single	Signal proces sing	Monitor intervals between SV fix time

[1] Source: Santiago Andres, "Detection solution analysis for simplistic spoofing attacks in commercial mini and micro UAVs ", M.S thesis, Dept. Computer Science, Tallinn University of Technology, Tallinn, Estonia, 2016

Name	Description	Encrypted	Networked or stand alone	Single or multi antenna	Layer	Receiver required capability
Multi-antenna Spoofing Discrimination	GPS correlators are applied to numerous beam outputs to spot and locate spoofer	No	Stand alone	Multi	Signal processing	Multiple receiver antennas
Defence Based on Navigation Message Authentication on L1C, L2C, or L5 (NMA)	Insertion of a public-key digital signatures inside the GPS civil navigation message	Yes	Stand alone	Single	Signal processing	Modification in the GPS IS: 2 New CNAV Messages
Consistency Check with Other position technologies	Comparison of the position with the obtained by mobile or Wi-Fi networks	No	Networked	Single	Position solution and navigation level	Internet connection

1.1.1. Problem Statement

Unmanned aviation systems primarily use GNSS to determine their position. This can be either GPS, GLONASS, Beidou or similar satellite positioning system. Even UAS may use multiple of these positioning systems at once to provide redundancy. However, studies have shown that GNSS signal can be jammed or spoofed as given in the relevant sections. There have been real cases where army UAV was landed by enemy forces by using such attacks. Moreover, some companies have developed gun shaped GNSS jammers to take down UAVs that violates the law or threatens civilians. Other than this, it is known fact that GNSS signals is nothing but a modulated signal with a very low power. This makes GNSS signal vulnerable to other environmental blockers such as big clouds or buildings, which block GNSS signal indoor or certain areas so that GNSS

receiver cannot get the signal. When we look at these facts, unavailability GNSS signal or any jamming or spoofing attack to GNSS can be vital for UAS. Currently, most of the mini UAVs are configured to land over there, hang in the air or take no action in case of GPS attack detection. Inevitability, this threatens the CIA (Confidentiality, Integrity and Availability) of the UAV as well as the people's security.

In today's world, various types of UAS used in different applications such as military missions, autonomous remote inspections or even just for fun. Lately, when we look at the numbers in reports, UAS market is expected to be tripled annually in global worlds [1]. When this is the case, naturally, we may expect to see more UAV out there in the market and even in our daily life.

When we consider these facts, there is a need to create a location estimation system which can act as a backup location estimator that estimates the location for the UAS when the GNSS signal is not available. This is possible by using various sensors on the UAS and this method is known to be Dead Reckoning.

This thesis aims to implement Dead Reckoning on a mini UAV in order to provide location estimation for the UAV to reach home or a secure place in case of unavailability of GNSS. By this way, this study expects to prevent UAV crashes or being hijacked when GNSS signal is not available by natural causes or disturbed by an attacker.

1.1.2. Research Questions

1. Is it possible to save UAS from being captured or malfunctioned due to spoofing/jamming attack to GNSS or unavailability of GNSS signal?
    1. Is it possible to detect GNSS spoofing or jamming and how? If it is possible, what are the approaches?
    2. What existing methods can potentially be used to save a UAS under GNSS spoofing or jamming attack?
2. How can IMU based DR can be implemented on mini UAV?
    1. What is the performance of IMU based DR on mini UAV?
3. How DR can be used in mini UAV when there is GNSS spoofing or jamming?
    1. How should UAV act in case of such attack?

1.2. Related Work

There have been various studies to estimate position data even though GPS signal is not available. These studies are known as DR (Dead Reckoning). DR method has been applied in many forms to provide indoor positioning where GPS signal is not available. In general, there are two type of DR implementations; vision based and IMU based. In Michigan University, scientists have developed PDR (personal dead reckoning) system where six DOF IMU attached to user's boot to estimate user's location in GPS-denied environments. In this study, they have concluded that PDR can accurately measure linear displacements with an error less than 2% per distance travelled [10]. Additionally, in Bremen University studies have shown that PDR measurement accuracy can be improved by enhancing it NN [11]. On top of this, several algorithms have been developed to improve IMU based DR accuracy and conclusion was IMU based PDR can produce more accurate location estimation when the appropriate algorithm is used [12].

Recently, vision based DR also got fair attention. In Zurich University, studies assured that it is also possible to estimate indoor positioning using panoramic vision systems. However, in this study, they have used a blimp-type of UAS and the images taken from UAV are processed on the ground station [13]. Okayama University engineers studied on the optical flow of ground image and came to the conclusion that using only vision based DR is not very reliable to estimate location data on omnidirectional ground vehicles [14]. Moreover, in Sidney University laboratories, vision based DR, enhanced with machine learning, successfully tested on a UAV. Additionally study showed that machine learning increase accuracy in vision based DR by increasing the machine rate between images [15].

In parallel to that, non-vision based DR implementation on UAV was proposed in 1972 and Lear Siegler Inc [16] takes relevant patent. In this regard, a cell phone aided DR approach implemented on a mini UAV at Minnesota University UAV laboratory. However, the designed system assumes to have cell phone tower coverage whereby the onboard cell phone receiver has line-of-sight communications with at least two cell phone towers for at least 50% of the time during the flight [17]. A Similar study has

been conducted in ETH Zurich Autonomous Systems Lab and researchers introduces a state estimation framework that allows estimating the attitude, full metric speed and the orthogonal metric distance of an IMU-camera system with respect to a plane [18].

To conclude, there are various applications of both IMU and vision based DR to various vehicles. However, there no IMU based DR implementation on mini UAS.

1.3. Contribution

Since there is no available IMU based DR reckoning implementation on mini UAS in the literature, this study provides a basis for the approach. Moreover, study provides a lab environment to test, verify and analyze the IMU based DR on mini UAS in order to observe the performance of the approach. The outcome of the study is python code that controls the behavior of a mini UAS in case of GNSS spoofing or jamming. The resulting code can be tested on both simulator environment and real physical mini UAV with latest firmware.

1.4. Methodology

In this study, firstly, a literature review is performed regarding the thesis topic. In the literature review part, the general structure of UAS, their current applications, market growth expectations, common problems and possible attacks to UAS' navigation system especially to GNSS are reviewed. Also, current DR implementation suggestions and implementations are reviewed. Later on, these findings are enriched by more detailed, technical and multidisciplinary information such as legal aspects of UAS usage. Meanwhile, problem statement and research questions establish more clearly, as presented in the previous sections. In order to fulfill the technical understanding and implementation, general mini UAS architecture, its components, GNSS and GNSS working principles are investigated. Later on, a solution proposal is presented. To realize this solution, incremental software development method is preferred and solution is implemented. A test environment is set up to be able to test the proposed solution and to perform various test cases. Lately, the test results are given and the conclusion is presented. Last but not least, future work of the study is given.

1.5. Thesis Organization

This thesis contains mainly four sections; introduction, technical background, solution development, test & results and conclusion & future work.

Introduction part contains problem statement, research questions. Additionally, it covers the related work done in the area. Moreover, study contribution is given and also the research methodology is explained.

Technical background section discusses navigation system overview specially GPS and also illustrates GNSS spoofing/jamming attacks. In addition to this, mini UAS architecture is explained and UAV applications are given by examples. Last but not least, usage of UAS and jammers are discussed by giving an example from USA applications.

In the solution development section, firstly a solution is proposed for the given problem in the problem statement. Secondly, a lab environment proposed and set up to be able to conduct experiments on proposed solution. Lastly, the solution is implemented by code in a simulation environment.

Test and results part contains various test cases and scenarios to provide a DR performance on mini UAS.

As for last section, conclusion and future work part summarizes the whole of this thesis and states the study outcomes. Moreover, it covers possible future work that can be built up on this study.

2. Technical Background

2.1. Navigation Systems Overview

The idea of GNSS goes back until the 1950s. Everything began with successfully launched satellite Sputnik (ball-shaped satellite with four antennas) by the Soviet Union in October 4, 1957. This exciting technology has played the most important role in GPS discovery besides demonstrating the Soviet Union's technology superiority over the USA. It also ushered in political and scientific reforms [19]. Although Sputnik's battery died after 3 weeks and it was burned in the atmosphere in three months, it was told that by this revolutionary achievement humanity has entered into space age [20]. Meanwhile, a group of scientists wanted to receive the signal broadcasted from Sputnik in an APL and they managed to capture and turn it into digital sound. Moreover, they recorded these signals with a time stamp. During these studies, they have discovered that, it was possible to predict the satellite's location from a known ground point from earth. Later on, they have realized that it was also possible to predict the receiver's location from a known satellite position and this was the born of GNSS idea in 1957 [21]. As it was a race in the space, later on, many satellites were launched on different dates by different nations as given in the Table 2 [22].

Table 2. Countries and GNSS systems

First Satellite Launch Year	**Country**	**Project**
1978	USA	GPS
2005	Europe	Galileo
2000	China	BeiDou
2010	Japan	QZSS

However, GNSS signal is known to be susceptible to interference and jamming. These vulnerabilities can be exploited, either intentionally or unintentionally, and cause GNSS signal to become unavailable in a given geographical area [23]. This may lead GNSS

receiving system to lose its position and may cause harm to civil infrastructure as stated in U.S. Transportation infrastructure's vulnerability to civil GPS disruption in 2001.

2.1.1. GPS

GPS, which is operated by the United States DoD (Department of Defense), was developed in the 1980s and became fully operational in 1995 [24]. There are three main elements which enable the GPS as shown in the Figure 2; space segment, control segment and user segment. The space segment consists of the satellites. Control segment is the unit which has bidirectional communication with space segment to observe satellites health and satellites' data accuracy. Control segment is responsible for correcting the broadcasted data from satellite by uploading correction data to the satellites using antennas located on the ground station. This upload step is performed by MCS in control segment. On the other hand, user segment is the end users where satellite data is processed. Note the communication between space segment and user segment is unidirectional.

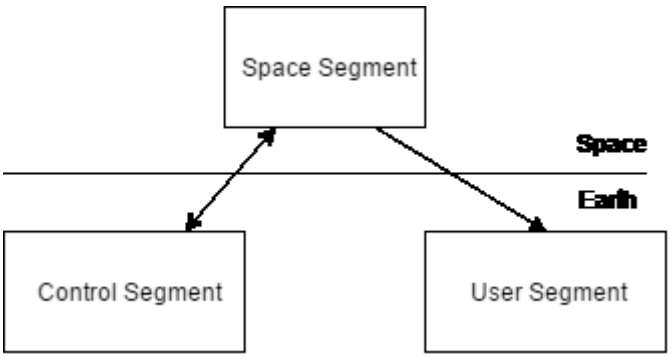

Figure 2. GPS segments

In the space segment, satellites send the signals on L band as a low powered radio signal. In this band, L1 has 1575.42 MHz carrier frequency for civilian usage and L2 has 1227.60 MHz carrier frequency for military usage. L1 and L2 carriers carry different kinds of codes and by this way, they provide different services to military and civilian usage. These carrier signals also carry a PNR (pseudo random code) code. PNR codes unique to each satellite and known by the GPS receivers. Once receiver captures the PNR from a satellite, it calculates the phase shift of the PNR signal by comparing the locally generated PNR signal and this gives the time passed during travel of signal from a satellite to the receiver. Since the speed of light is known, multiplication of this time and speed of the light gives the distance between the receiver and the satellite.

However, this is not enough to determine the location of the receiver. By a single satellite data, it can be only assured that the receiver is somewhere on a circle with a known radius from the satellite. To extract the exact location of the receiver, at least four satellites signal is required. In this way, the intersection of four circles with known radius from four different satellites is the location of the receiver [25]. Currently, although all GPS satellites provide PNR codes, there are additional services are available by modernized new generation satellites known as GPS Block IIIA or GPS [26]. However, this project is still ongoing and not expected to be launched until 2017 [27].

2.1.2. GNSS Spoofing/Jamming Thread

As described in the previous sections, GNSS signal is nothing but a piece of modulated signal which has very low minimum received power around -161.4 dBW. Hence, this signal can be easily interfered with the surrounding obstacles or other signals present in between space segment and user segment. Moreover, it can be jammed or spoofed as illustrated in Table 3 GNSS vulnerability report. In this manner relevant academic studies and real cases of attacks given in the previous sections. A very simplistic example of the given scenario in the blow can be accomplished by SDR. SDR is a hardware which can generate signals at various frequencies. Although this frequency variation is limited by the hardware limitations, the user can adjust the desired frequency level within hardware lower and upper frequency boundaries. These studies, real cases and the vulnerability report given below show that GNSS signal is open to certain cyber threats.

Table 3. GNSS vulnerability report

	GPS Jamming Attack	**GPS Spoofing Attack**
Business Asset	UAV mission data, UAV home location, captured images/videos	UAV mission data, UAV home location, captured images/videos
Information System Asset	UAV memory, UAV controller	UAV memory, UAV controller
Security criterion	CIA triad(Confidentiality, integrity and availability) of UAV mission data, UAV home location, captured images/videos	CIA triad of UAV mission data, UAV home location, captured images/videos

	GPS Jamming Attack	GPS Spoofing Attack
Risk	An attacker may spoof GNSS signal to make UAV land and by exploiting the lacking the secondary navigation system on UAV and gather sensitive info in the UAV. This kills CIA triad of UAV mission data, UAV home location, captured images/videos	An attacker or unavailability of GNSS signal may cause UAV to land which leads loss of the UAV and UAV data due to lacking the secondary navigation system on UAV. This kills CIA triad of UAV mission data, UAV home location, captured images/videos
Impact	UAV and UAV data are not available	UAV and UAV data are not available
Event	An attacker may spoof GNSS signal to make UAV land and by exploiting the lacking of the secondary navigation system on UAV and gather sensitive info in the UAV	An attacker or unavailability of GNSS signal may cause UAV to land which leads loss of the UAV and UAV data due to lacking the secondary navigation system on UAV
Vulnerability	Lacking of secondary navigation system and post incident activity	Lacking of secondary navigation system and post incident activity
Thread	An attacker may spoof GNSS signal to make UAV land and gather the info in the UAV.	An attacker or unavailability of GNSS signal may cause UAV to land which leads loss of the data in UAV
Thread Agent	Attacker	Attacker, or GNSS signal unavailability
Attack Method	1. Spoof GNSS signal 2. Make UAV land 3. Analyse the data in UAV	1. Jam GNSS signal or GNSS signal is not available 2. Make UAV land 3. Analyse the data in UAV
Security Requirements	Secure GNSS communication or secondary navigation system and post incident activity plan	100% available GNSS system or secondary navigation system and post incident activity plan

2.2. UAS

2.2.1. UAS Architecture

There are various types of UAS in the market and UAS components may vary depending on its category. In general, UAS categorization can be done in two way depending on their physical structure and functional properties. When the physical structure is considered, there are four categories as given below [28]:

Fixed-wing UAV: UAVs that requires runway
Rotary-wing UAV: UAVs that can land and takeoff vertically
Blimps: balloon shaped UAVs with low speed
Flapping-wing UAV: UAVs with flexible wings. The combination of fixed-wing and rotary-wing UAV.

When we consider the functional properties of the UAS, they are grouped by four main criteria as given in Figure 3; maximum takeoff weight, maximum flight altitude, endurance and data link range. Depending on these properties, they can service for different kind of missions such as surveillance, environmental measurements, parcel delivery and much more. Depending on the UAS category, UAS may contain the following elements [29]:

- Multiple UAV
- Ground control shelters
- A mission planning shelter
- A launch and recovery shelter
- Ground data terminals
- Remote video terminals
- Modular mission payload modules
- Air data relays
- Miscellaneous launch, recovery, and ground support equipment

	Category (acronym)	Maximum Take Off Weight (kg)	Maximum Flight Altitude (m)	Endurance (hours)	Data Link Range (Km)	Example	
						Missions	Systems
Micro/Mini UAVs	Micro (MAV)	0.10	250	1	< 10	Scouting, NBC sampling, surveillance inside buildings	Black Widow, MicroStar, Microbat, FanCopter, QuattroCopter, Mosquito, Hornet, Mite
	Mini	< 30	150-300	< 2	< 10	Film and broadcast industries, agriculture, pollution measurements, surveillance inside buildings, communications relay and EW	Mikado, Aladin, Tracker, DragonEye, Raven, Pointer II, Carolo C40/P50, Skorpio, R-Max and R-50, RoboCopter, YH-300SL
Tactical UAVs	Close Range (CR)	150	3.000	2-4	10-30	RSTA, mine detection, search & rescue, EW	Observer I, Phantom, Copter 4, Mikado, RoboCopter 300, Pointer, Camcopter, Aerial and Agricultural RMax
	Short Range (SR)	200	3.000	3-6	30-70	BDA, RSTA, EW, mine detection	Scorpi 6/30, Luna, SilverFox, EyeView, Firebird, R-Max Agri/ Photo, Hornet, Raven, phantom, GoldenEye 100, Flyrt, Neptune
	Medium Range (MR)	150-500	3.000-5.000	6-10	70-200	BDA, RSTA, EW, mine detection, NBC sampling	Hunter B, Mücke, Aerostar, Sniper, Falco, Armor X7, Smart UAV, UCAR, Eagle Eye+, Alice, Extender, Shadow 200/400
	Long Range (LR)	-	5.000	6-13	200-500	RSTA, BDA, communications relay	Hunter, Vigilante 502
	Endurance (EN)	500-1.500	5.000-8.000	12-24	> 500	BDA, RSTA, EW, communications relay, NBC sampling	Aerosonde, Vulture II Exp, Shadow 600, Searcher II, Hermes 450S/450T/700
	Medium Altitude, Long Endurance (MALE)	1.000-1.500	5.000-8.000	24-48	> 500	BDA, RSTA, EW weapons delivery, communications relay, NBC sampling	Skyforce, Hermes 1500, Heron TP, MQ-1 Predator, Predator-IT, Eagle-1/2, Darkstar, E-Hunter, Dominator
Strategic UAVs	High Altitude, Long Endurance (HALE)	2.500-12.500	15.000-20.000	24-48	> 2.000	BDA, RSTA, EW, communications relay, boost phase intercept launch vehicle, airport security	Global Hawk, Raptor, Condor, Theseus, Helios, Predator B/C, Libellule, EuroHawk, Mercator, SensorCraft, Global Observer, Pathfinder Plus,
Special Task UAVs	Lethal (LET)	250	3.000-4.000	3-4	300	Anti-radar, anti-ship, anti-aircraft, anti-infrastructure	MALI, Harpy, Lark, Marula
	Decoys (DEC)	250	50-5.000	< 4	0-500	Aerial and naval deception	Flyrt, MALD, Nulka, ITALD, Chukar
	Stratospheric (Strato)	TBD	20.000-30.000	> 48	> 2.000	-	Pegasus
	Exo-stratospheric (EXO)	TBD	> 30.000	TBD	TBD	-	MarsFlyer, MAC-1

Figure 3. UAV categories[1]

However, when we look at mini UAS, in general, components can be grouped in two; UAV and RC controller as given in Figure 4. On the RC controller side, there is an embedded operating system, RC transceiver and Wi-Fi access point. RC controller acts as a gate in between user and UAV. Optionally, a PC/Mobile device can be connected to RC controller to get/set telemetry data on the UAV. On the UAV side, there is a transceiver, autopilot, sensors, battery, ESC (electronic spin controller) and motors. Autopilot is mainly responsible for vehicle's trajectory control. In general, the autopilot

[1] Source: taken from Bento, M., "Unmanned Aerial Vehicles an Overview", 2008, p. 2, at http://www.insidegnss.com/auto/janfeb08-wp.pdf

is also connected to UAV sensors to processes sensor data. In some cases, sensor data can be processed in a separated layer from the autopilot. Additionally, IMU, accelerometer, gyroscope, magnetometer and barometers are commonly used sensors on mini UAV.

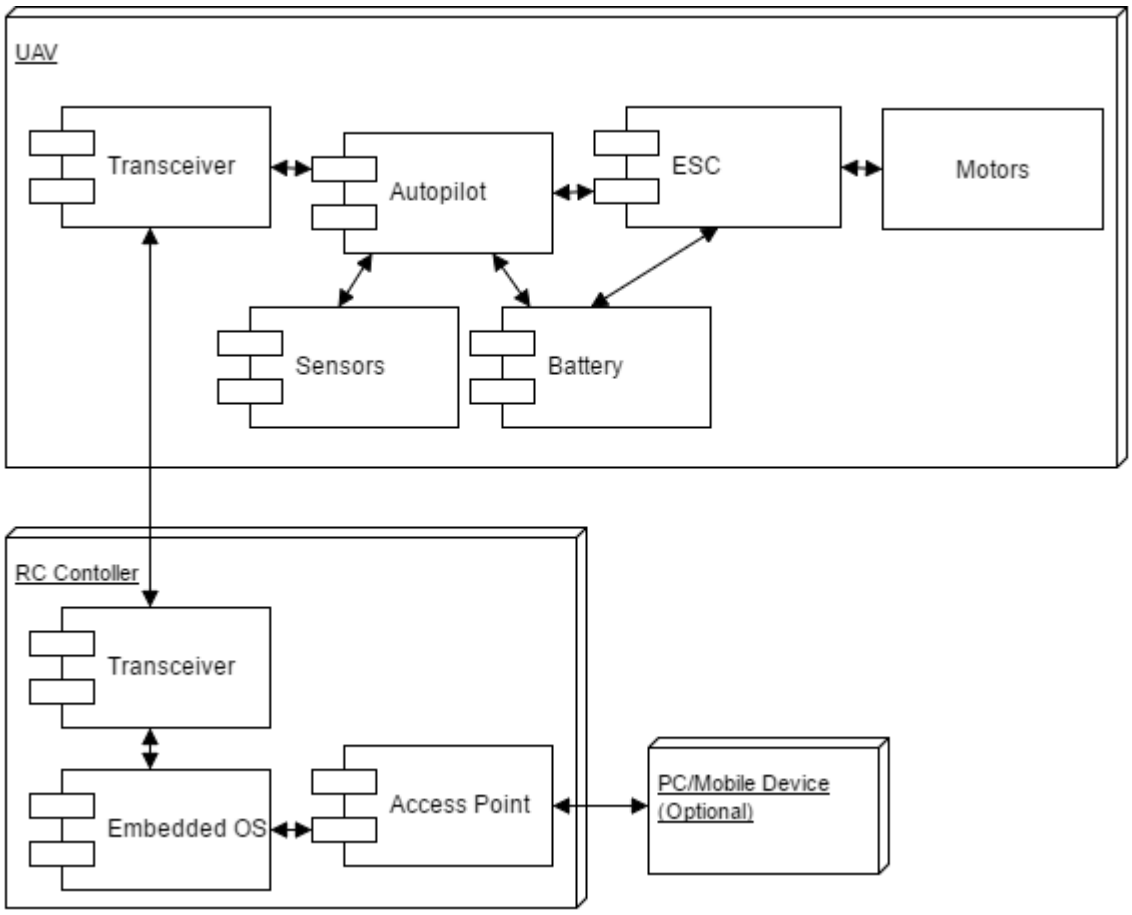

Figure 4. Mini UAS system architecture

2.2.2. UAS Applications

As mentioned in the previous sections, there are varieties of UAS which enables different kind of applications of unmanned aviation vehicles to various areas. This can be grouped as two; military and civil applications. Additionally, application of UAS depending on each of these areas can be broken down into three more sub-groups in regarding UAS tasks as; dull, dirty and dangerous [29]. Some of the missions and their categorization are summarized in the Table 4.

Dull: Repeating missions that don't require active user input such as autonomous railway inspection

Dirty: The missions that may be harmful to manned aircraft such as nuclear waste area observation

Dangerous: The missions that may put pilot's life in danger such as ISR missions

Table 4. UAS application categories

Military			**Civil**		
Dull	**Dirty**	**Dangerous**	**Dull**	**Dirty**	**Dangerous**
Electronic intelligence	Elimination of unexploded bombs	Decoying missiles by the emission of artificial signatures	Traffic spotting	Environmental monitoring	Power line survey
Relaying radio signals	Monitoring of nuclear, biological or chemical (NBC) contamination	Carrying bomb to the target	Pipeline survey	Disaster and crisis management search and rescue	Fire fighting
Airfield base security	Monitoring of unknown objects		Aerial photography		Disaster and crisis management search and rescue
			Communications relay		

2.3. Legal Aspects of Jammers and UAS

2.3.1. Jammers

Jammer is an electronic component which can disable a communication between targeted devices by filing the used spectrum with a random signal. In other words, jammers can be configured to generate and emit random signal in desired frequencies and this results unsuccessful connection on targeted devices due to enormous noise in

the communication channel. In general, when there are enough sources, any signal can be jammed without regard to encryption usage on targeted signal. Additionally, jammers can prevent various communications such as; RC vehicle, cell phone, GNSS, Wi-Fi, TV signal, radio, UAV CCS and list expends anything until anything that uses a radio signal.

There are plenty of jammers available on the market. Depending on jammers' frequency and coverage range, they can be very complex and expensive. The higher the frequency and ranges go, the higher the complexity and cost rise. However, the vice versa is also true. For instance, Figure 5 illustrates a simple mobile phone jammer which costs a couple of Euros.

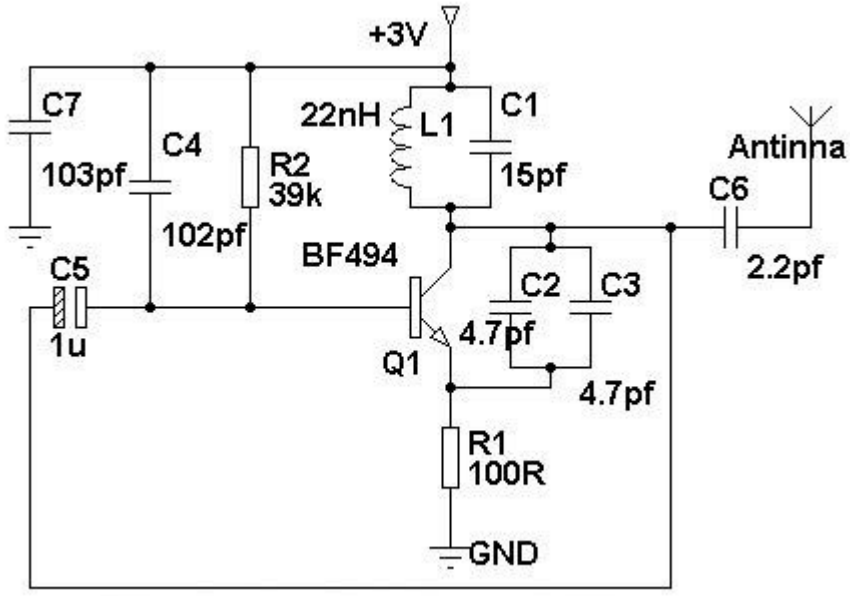

Figure 5. Mobile phone jammer[1]

As mentioned previously jammers can disable various of radio signal based services and this may cause various harm for service users and civilians. When this is the case, there are several of rules and regulations in countries regarding jammer usage. In USA, these regulations can be summarized as follows [30]:

[1]Source: Simple Mobile Phone Jammer Circuit, January 12, 2012, https://circuitsstream.blogspot.com/2012/09/simple-mobile-phone-jammer-circuit.html

- It is illegal to interference to satellite communication, US government communication and radio communication of any station licensed or authorized by US government
- It is illegal to manufacture, import and market jammer
- Radio transmitters, jammer is also a radio transmitter, and operators should be licensed or authorized under US government
- Any violation of these rules cost $122,500 per any single act

Although these are general regulations, there are many exemptions also exists. These exemptions and regulation details can be found on FCC's website. However, some jammer applications' legality is still difficult to answer and in this cases FCC advice civilians to contact FCC for more clear answers [31]. Some of the jammer application which is open to comments is given as follows:

- Is it legal to use jammer inside a car, bus or a plane to block mobile phone communication in the vehicle?
- Can jammer be used during exams in education centers to prevent cheating by blocking mobile phone communication?
- Is it legal to apply various frequency of radio signal to a chemical element to conduct some experiments?

Additionally, studies have shown that it is possible to locate RF jammers in certain conditions [32, 33, 34]. It is not very likely to detect a jammer which affects a very low range and produce very low powered radio signal. This fact also leaves an open question: Is it possible to detect all kind of jammers and prevent their usage legally?

2.3.2. UAS

It is a known fact that UAS usage is increasing day by day. Even we can see some UAS applications in our daily life in the form of air delivery systems, live broadcasting tools or even just as a toy to have fun. However, UAS may cause serious issues to civilians, environment and even to animals. When we consider civil micro or mini UAS, basically, they may cause the followings situations;

- UAV's propeller may damage or kill animals

- UAV's propeller may damage or kill humans
- UAS can be used to deliver illegal good over borders
- People may invade other people's privacy by using the camera on the UAS
- UAV could be weaponized to destroy target
- Technically insufficient drone operator may unintentionally crash UAS into critical infrastructure or onto people
- Some cyber-attacks can be carried out using UAS to increase complexity for traceability.
- Poisonous chemicals can be delivered to target using UAS
- Can be used for ISR purposes by terrorists
- UAS may trespass over people's private property

When we look at the given scenarios above and consider the fact that studies expect that UAS production will be tripled in next decade, as mentioned in previous sections, UAS rules and regulations is mandatory and these circumstances push lawmakers to bring some set of rules and regulations to UAS area. However, there are some key questions to enable these regulations such as:

1. What to regulate, what is the scope?
2. How to regulate existing UAS?
3. Should UAS manufacturing also standardized and regulated?
4. How to prove circumstances that a UAS violating the law?
5. How to capture or land the UAVs that is violating the law?
6. Should lawmaker also regulate the things that UAV can carry?
7. How to enforce these laws to DIY UAVs?
8. How should legal entities use UAS?
9. In case of violation of law should UAS data be shared between different countries?

These are somehow difficult questions to answer, so that different countries are developing a different set of rules and regulations. When we look at the regulation in fourth amendment aspect we can see that different states have different regulations. For instance, regarding trespassing, Wells C. Bennett says that "Not all states define trespassing or drone surveillance in the same way, or apply identical privacy

protections." Additionally, it is also said that judgment is generally done by covering how majorities see of personal privacy and social privacy norm in [35]. Some countries, such as USA, regulate the UAS usage in a way that even legal entities need a warrant to use UAS against high-risk targets such as terrorists. Moreover, some USA states also regulate sharing of the data gathered by UAS among different counties. While some states allow sharing this data, some of them prohibit such as Tennessee. Moreover, certain states such as Oregon requires deleting gathered UAS data after 2-3 days and some states requires public entities to provide an annual report of their UAS usage to USA government [36].

In general, when we look at FAA regulations in USA, FAA applies several restrictions and check lists depending on UAS category and UAS flight purpose as shown in Table 5. According to these regulations, if someone wants to fly a UAS over than 0.55 lbs in USA, UAS must be registered online using FAA website which is https://registermyuas.faa.gov. During this registering process, FAA's system displays an alphanumeric registration number and this number should be marked on the UAS in a readable way. Additionally, this registration is valid for only three years and registration is possible only if the UAS operator is older than thirteen years old [37].

Table 5. FAA UAV flight regulations

	Fly for Fun	**Fly for Work**
Pilot Requirements	No pilot requirements	Must have Remote Pilot Airman Certificate Must be 16 years old Must pass TSA vetting
Aircraft Requirements	Must be registered if over 0.55 lbs.	Must be less than 55 lbs. Must be registered if over 0.55 lbs. (online) Must undergo pre-flight check to ensure UAS is in condition for safe operation
Location Requirements	5 miles from airports without prior notification to airport and air traffic control	Class G airspace*

	Fly for Fun	Fly for Work
Operating Rules	Must ALWAYS yield right of way to manned aircraft Must keep the aircraft in sight (visual line-of-sight) UAS must be under 55 lbs. Must follow community-based safety guidelines Must notify airport and air traffic control tower before flying within 5 miles of an airport	Must keep the aircraft in sight (visual line-of-sight)* Must fly under 400 feet* Must fly during the day* Must fly at or below 100 mph* Must yield right of way to manned aircraft* Must NOT fly over people* Must NOT fly from a moving vehicle*
Example Applications	Educational or recreational flying only	Flying for commercial use (e.g. providing aerial surveying or photography services) Flying incidental to a business (e.g. doing roof inspections or real estate photography)
Legal or Regulatory Basis	Public Law 112-95, Section 336 – Special Rule for Model Aircraft FAA Interpretation of the Special Rule for Model Aircraft	Title 14 of the Code of Federal Regulation (14 CFR) Part 107

Similarly, when we look at Estonia, UAS usage is also regulated with certain set of laws as given below[1];

- Aviation Act
- Regulation of the Government of the Republic No 240
- Regulation of the Minister of Economic Affairs and Infrastructure No 24
- Regulation of the Government of the Republic No 189
- General Precept of Director General of the Estonian Civil Aviation Administration No 4.1-7/15/33 of 9 June 2015

[1]Source: Civil aviation administration of Estonia "How to operate UA, Internet: https://www.ecaa.ee/en/how-operate-ua, [May. 24, 2017]

In Estonia, special UAV flight permit is required if the UAV operator plans to operate in certain controlled/restricted areas. Estonian CAA provides map of these areas. In order to get flight permit, operator have to make an application to https://www.ecaa.ee and provide basic information such as name, address, phone number and email address. Later on, this information is registered in CAA systems and also made publicly available on the internet [38, 39].

Once there are certain set of rules and regulations, it is also important a must to have to detect and counteract to the UAS that violating the law. However, it is not always very straightforward operation to detect and capture an unwanted UAS. In this manner, different countries are using various techniques to prevent this occasion. While countries are deploying jammer based drone defend systems such Turkey, USA and Australia some counties trains eagles to catch UAS such as Netherlands. On the other hand, some countries also have the power to take down a drone using laser based drone defend systems like China [40, 41, 42].

2.4. 3DR Solo Mini UAS

In this thesis, 3DR Solo UAS has been chosen since it is based on open source and the HW is customizable when needed. 3DR Solo has Linux based OS on both UAV and on the remote controller. Additionally, it uses Pixhawk 2 which very commonly open source autopilot. Although 3DR Solo UAV doesn't come with embedded camera, it can be installed if needed. For this thesis, the embedded IMU, gyroscope, magnetometer and barometer sensor is sufficient. The other technical specifications are summarized in the Table 6 [43, 44, 45]. All these sensors, autopilot and other components are deployed as shown in Figure 6 DR system architecture.

Table 6. 3DR Solo UAV technical specification

	SPECIFICATIONS	**DETAILS**
GENERAL	**Brand**	3DR Robotics
	Model	Solo
	UAS Type	Multi-rotor Mini UAS

	SPECIFICATIONS	DETAILS
SOFTWARE	**OS**	3DR Poky based on Yocto Linux 1.5.1
	Follow me	-
	SDK	Available
HW & PHYSICAL	**Wight**	1500gr
	Processor	ARM Cortex A9 (i.MX6 Solo by Freescale), 1Ghz, 1 CPU core with VPU and GPU
	Data Link Frequency	2.4 GHz
	GPS Receiver	U-Blox NEO-7N
	Secondary Navigation System	-
	Camera	By default none but can be installed
	Sensors	3 Axis accelerometers IMU 3 Axis gyroscope 3 Axis magnetometer Barometer
	Autopilot	Pixhawk 2
	HW Extension Port	Solo Accessory bay
	Battery	Lithium polymer 5200 mAh 14.8 Vdc
	Motors	880kV
PERFORMANCE	**Endurance**	25m
	Maximum Altitude	~350m however suggested: 122m
	Maximum Payload Weight	420gr

	SPECIFICATIONS	DETAILS
	Maximum Speed	89km/h
CONTROLLER	**Processor**	ARM Cortex A9 (i.MX6 Solo by Freescale), 1Ghz, 1 CPU core with VPU and GPU
	TX Power Output	17dBm

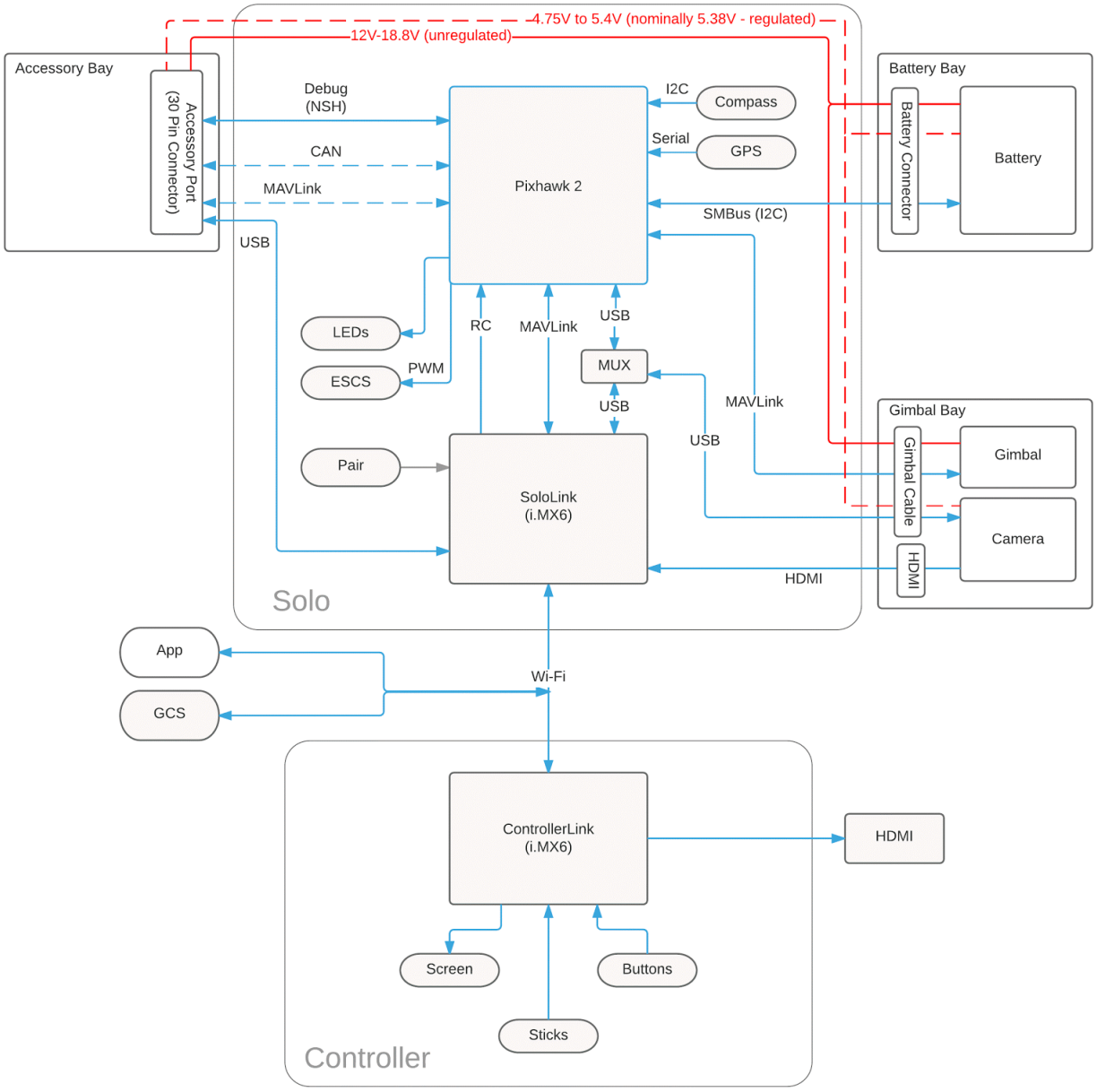

Figure 6. 3DR system architecture[1]

[1] Source: 3DR, "Architectural Overview", 2015, at https://dev.3dr.com/concept-architecture.html

3. Solution Development

3.1. Proposed Solution

On Ardupilot autopilot system in normal scenario, Kalman Filter gets data from GNSS, IMU, accelerometer, barometer and gyroscope as shown in Figure 7. According to these values, filter estimates the UAV's location on the 3D plane. Additionally, when there is no data available from any of these sensors, implemented Kalman Filter estimates the position using only available sensors' data on the UAV.

As we know from previous sections, GPS data is not always available and in this study, it is assumed that GPS signal is not available for a certain period of time due to a cyber-attack.

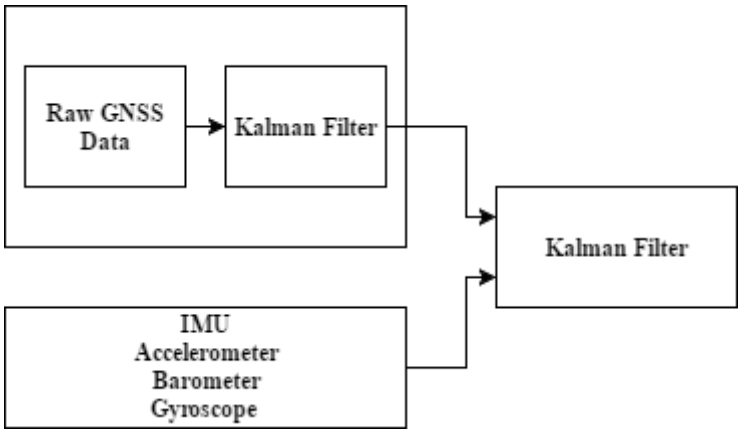

Figure 7. Autopilot location estimation

This thesis proposes to disable GPS feed to Kalman Filter when GNSS spoofing or jamming attack detected and send back UAV to launch point. In order to accomplish this, as shown in Figure 8 activity diagram, spoofing or jamming signal should be gathered from GNSS. Later on, if UAV is autonomous flight mode, it should disable GNSS feed to Kalman Filter, warn user and start heading to launch point until spoofing or jamming is no longer present. If there is no more thread anymore, GNSS feed can be enabled and the operator can be informed if s/he wants to update mission or not. Depending on the decision mission should continue and all this cycle repeats until completing the flight. On the other hand, if UAV is in manual flight mode and GNSS attack is detected, again GNSS feed should be disabled and the warning message should

be displayed to the operator until the attack is no longer exists. By this way, this study foresight that UAV can go back to launch point or a point that close to launch point by taking advantage of DR when there is GNSS spoofing or jamming.

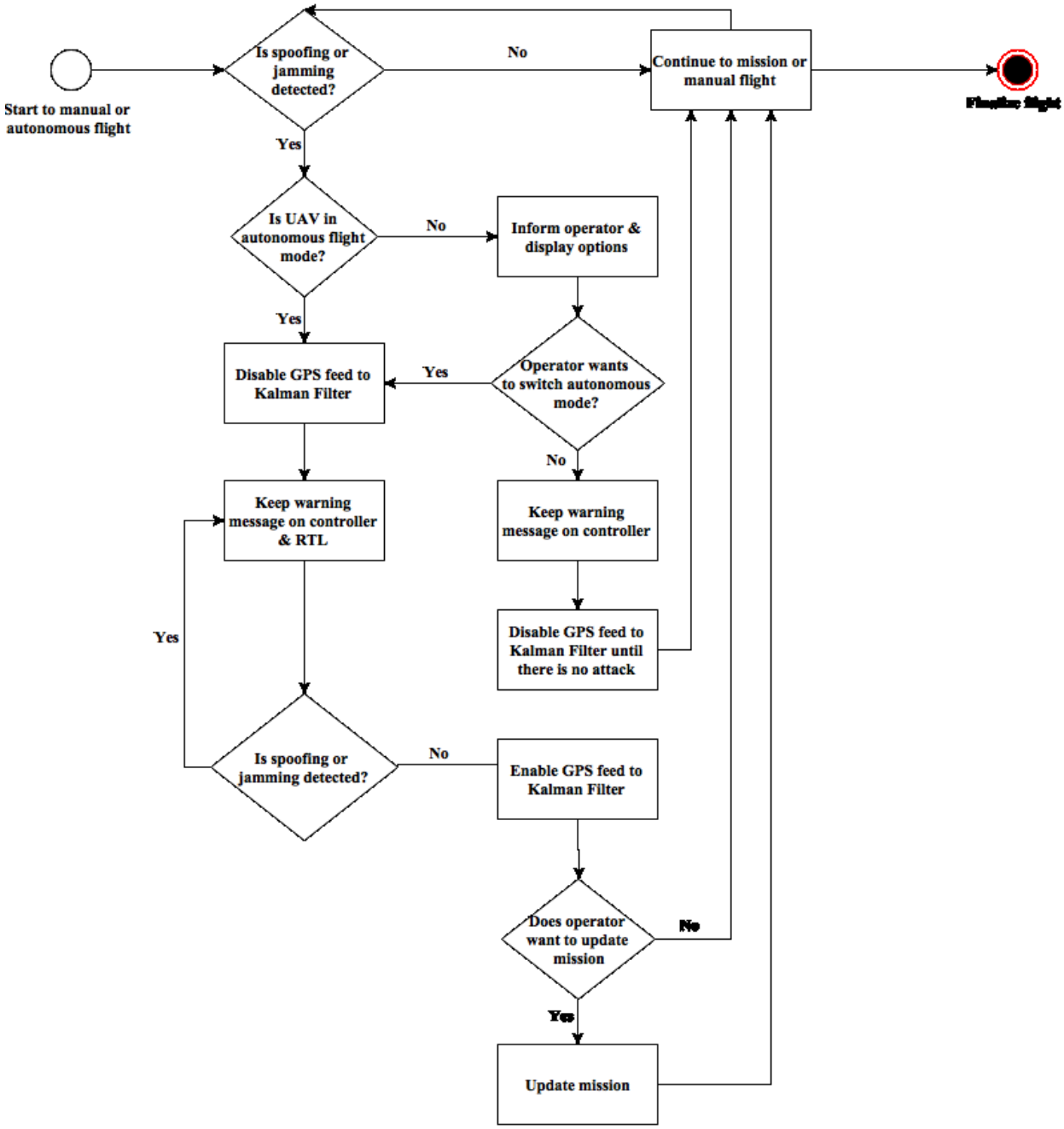

Figure 8. Solution activity diagram

3.2. Test Environment Setup

The following test setup has been set on Lenovo y5070 idea pad.

In order to be able to use 3DR Drone-Kit Python SDK, the following software setup is needed in the following order;

1. **Python**

Python is high level programming language [46] that is needed to develop applications using 3DR Drone-Kit Python SDK. Since Python is interpreted language, this allows us to develop an application in both Unix and Windows based environments.

2. **Pip**

Pip is a package manager installs python packages. This recommended tool for python allows us to install packages using a couple of lines.

3. **Drone Kit package:**

Drone kit is a project that developed by 3DR Robotics to provide SDK for their UAS. Basically, an application that is written using Drone-Kit SDK runs on 3DR Solo's embedded system. Additionally, this SDK also provides API to get or set Ardupilot parameters that enable great control over the UAS. It also allows performing complex algorithms on the companion computer on the UAS. Moreover, Drone-Kit is [47]:

- Open source
- Supports UAVs that use (Micro Aviation Vehicle) MAVLink protocol [48]
- Runs on both Linux, Mac, Mac OS X and Windows

4. Drone Kit-stil package: 3DR

Drone-Kit STIL is open source UAV simulator. STIL runs on both Windows, Mac OS X and Linux (x68 architecture only). It is also possible to run STIL in a virtual machine. Most importantly Drone-Kit STIL can connect to multiple GCS or RC [49]. This simulator also supports the following UAVs:

- solo-1.2.0
- solo-2.0.18
- solo-2.0.20
- plane-3.3.0
- copter-3.3
- rover-2.50

5. Virtualenv

Virtualenv tool is used to isolate, create and manage different virtual python environments. It also keeps tracks the dependencies and keeps the projects isolated. Also removes version mismatch issues during development [50].

6. MAVproxy

MAVproxy is minimal UAV GCS station to control MAVLink based UAVs. It has powerful features such as [51]:

- Portable
- Extendable
- CLI based
- Lightweight
- Contains modules to support external components such as joysticks
- It can be used as proxy for MAVLink

7. Ardupilot Mission Planner

Mission Planner is full-featured GCS for ArduPilot project. Mission Planner allows users to:

- Update software on UAV's autopilot
- Setup and tune UAV
- Set up autonomous fight by marking on map
- Analyze logs on the UAV
- Supports STIL
- Operate UAV in FPV

8. Python IDE (Optional)

In this thesis, PyCharm Python IDE is used. PyCharm is Python development environment to support developer [52].

Once all the packages and applications are installed, Drone-Kit-STIL started. STIL listens to any MAVLink message on TCP port 5760 on the local PC. In order to be able to connect STIL to Mission Planner, MAVProxy has been set to proxy MAVLink messages from port 14550, 14551 to 5760. This enables to develop code on the IDE using Drone-Kit and to see immediate results in a GUI on the Mission Planner as shown in the Figure 9 and 10.

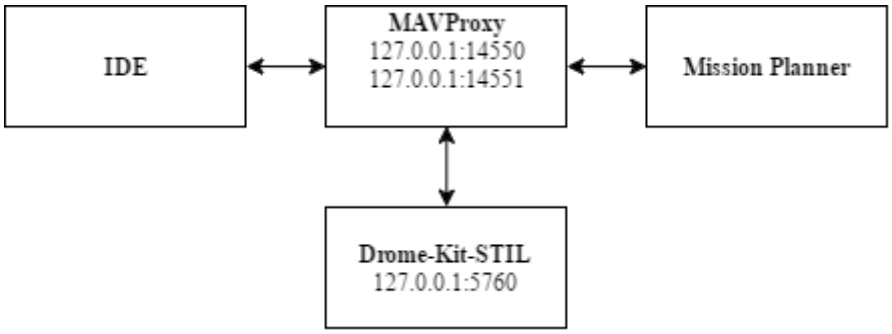

Figure 9. Lab setup

Figure 10. lab setup mission planner view

3.3. Solution Implementation

Firstly, UAV simulation is started using "*dronekit-sitl rover*" command on windows command line. As shown in the Figure 11, this piece of code, downloads UAV simulator, sets the initial home position and binds to TCP port 5760. From this point, UAV simulator is ready to receive MAVLink commands to perform the operation.

Figure 11. Start simulator

Once UAV simulator is up and running, MAVLink tool is started with three parameters:

--master: this parameter is to set protocol, IP and port of the running simulator
--out: This is the end which bridged with master
--out: This is the end which bridged with master

When the protocol is not specified, the default protocol is UDP. By this setup, whenever local PC receives a MAVLink message to port 14550 via UDP protocol, it will be forwarded to local PC port 5760 via TCP protocol. Vice versa is also valid. Similarly, this enabled bidirectional MAVLink messaging between local pc port 14551 and master node as presented in Figure 12.

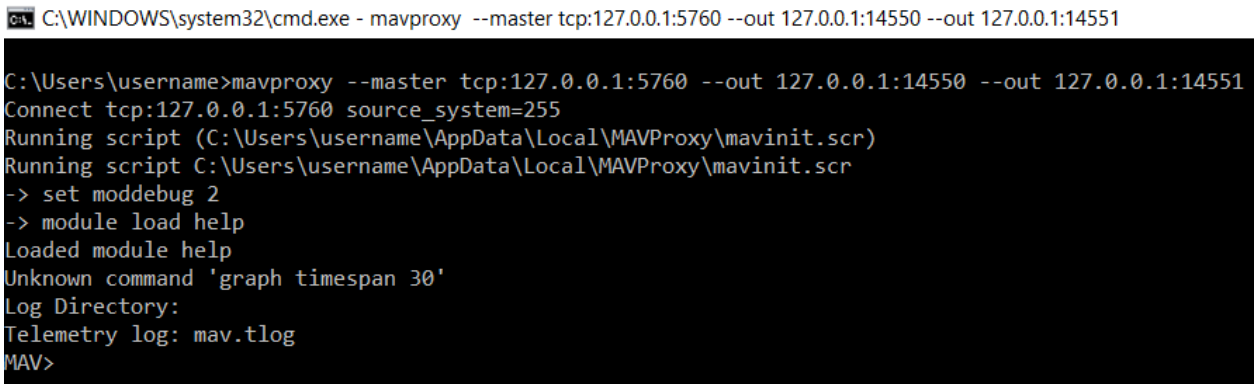

Figure 12. start MAVLink proxy

Later on, Mission Planner has been started and local port is set to be 14551 as shown in the Figure 13 and 14. By this way, all connections are provided to run code that is developed.

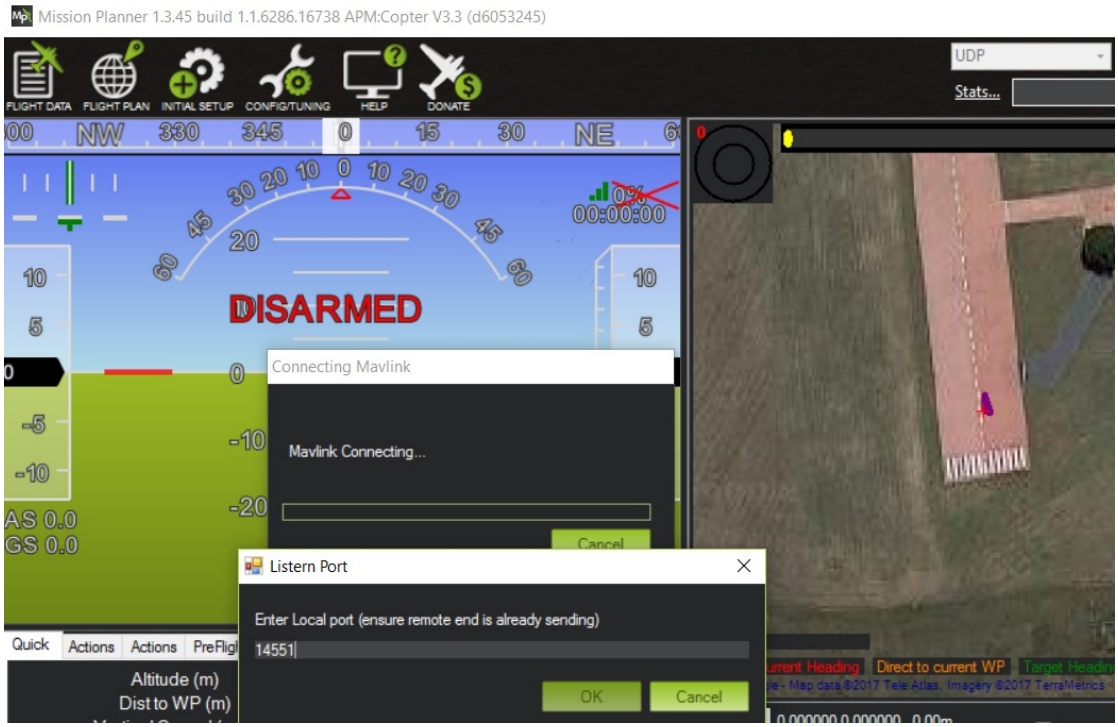

Figure 13. Mission planner, SITL connection

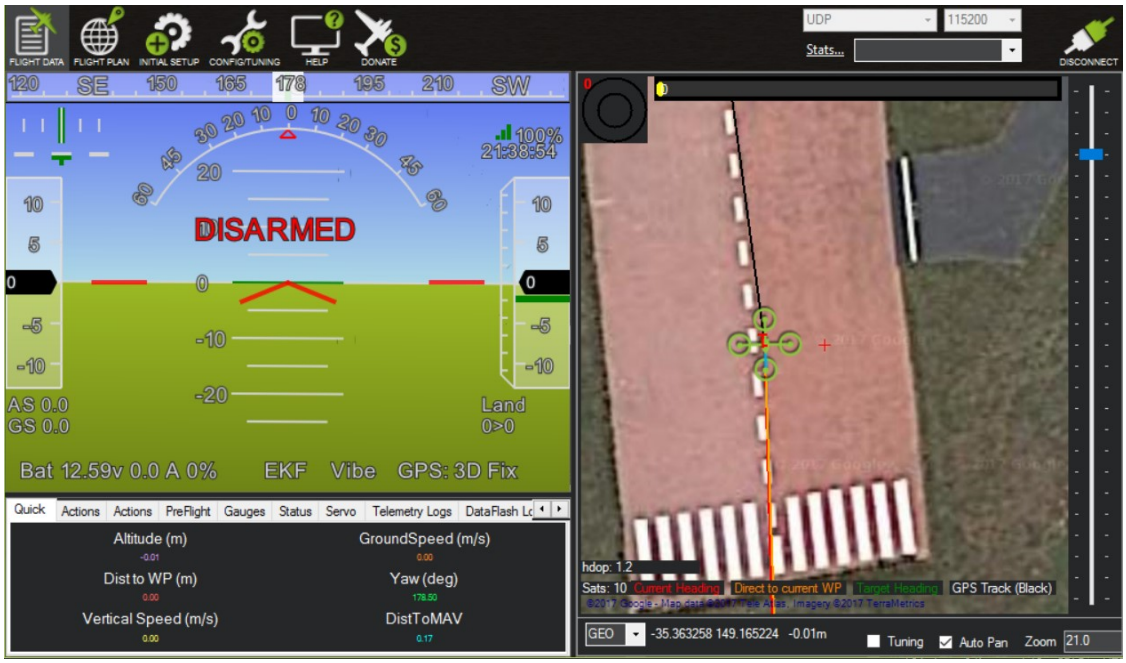

Figure 14. Mission planner SITL ready to flight

In normal cases, UAVs that is controlled by Ardu autopilot system, has a failsafe mechanism. This failsafe mechanism enables UAV to land to its current location in case of fail or emergency. Various events can trigger this such as; losing a rotor, battery limit and processing board failures. Similarly, GPS related events can also trigger this mechanism. For instance, when the GPS signal is lost due to weather conditions, failsafe is triggered.

As this study proposes to disable GPS source to rely on only Dead Reckoning, the failsafe mechanism should be disabled. This can be achieved by setting the 'GPS_FAILSAFE'parameter to '0' on the UAV. Drone-kit allows users to set 'GPS_FAILSAFE' even during a mission. Likely, the user can set 'AHRS_GPS_USE' parameter. This parameter enables or disables the GPS input to EKF. Additionally, this parameter will be used to simulate unavailability of GPS signal during the test.
In following code, firstly global parameters, UAV simulator connection string are defined. Later on, the code connects to UAV and waits for UAV to reach a certain level of altitude to go for a mission. Additionally, there is 'any_parameter_callback' method which is a callback function that mimics constant checking of GPS spoof or jam is presents or not. If such attack exists, code disables GPS failsafe and GPS source to EKF. Later on, if UAV is in autoflight mode, code directs UAV to the home location by

informing the user otherwise, user can still control the UAV manually but s/he will be presented warning message. This process repeats until the attack is no longer presents. When the UAV is safe from the attack, the code changes 'AHRS_GPS_USE' and 'GPS_FAILSAFE' parameters to initial state and retires to complete the mission. Once the mission is completed, UAV goes back to launch point, in other words, to the home. Since callback function constantly checks if attack is present again or not, this process repeats through the mission.

```
from dronekit import connect, VehicleMode, LocationGlobalRelative
import time

connection_string = 'udpin:0.0.0.0:14550'

#global parameters
global_takeoff_altitude = 1
global_airspeed = 10
global_groundspeed = 10

# Connect to the Vehicle
print 'Connecting to uav on: %s' % connection_string
uav = connect(connection_string, wait_ready=True)

def arm_and_takeoff(aTargetAltitude):

  print "Arming motors"
  # Copter should arm in GUIDED mode
  uav.mode = VehicleMode("GUIDED")
  uav.armed = True

  # Confirm uav armed before attempting to take off
  while not uav.armed:
     print " Waiting for arming..."
     time.sleep(1)

  print "Taking off!"
  uav.simple_takeoff(aTargetAltitude)  # Take off to target altitude

  # Wait until the uav reaches a safe height before processing the goto (otherwise the command
  #  after Vehicle.simple_takeoff will execute immediately).
  while True:
     print " Altitude: ", uav.location.global_relative_frame.alt
     # Break and return from function just below target altitude.
     if uav.location.global_relative_frame.alt >= aTargetAltitude * 0.95:
        print "Reached target altitude"
        break
     time.sleep(1)

arm_and_takeoff(global_takeoff_altitude)

print "Initial 'AHRS_GPS_USE' parameter is: %s" % uav.parameters['AHRS_GPS_USE']
uav.parameters['AHRS_GPS_USE'] = 1
uav.parameters['GPS_FAILSAFE'] = 1
```

```
def go_to_mission(message, lat, lon, alt, sec):
  print "%s" % message
  mission_point = LocationGlobalRelative(lat, lon, alt)
  uav.simple_goto(mission_point)
  print "***MISSION COMPLETED***"
  time.sleep(sec)

def go_home_autonomously():
  print "UAV is in autonomous flight mode"
  print "Disabling EKF GPS source"
  uav.parameters['AHRS_GPS_USE'] = 0
  print "AHRS_GPS_USE is set to 0 and EKF GPS source is disabled"
  print "***ALERT ON RC CONTROLLER***"
  go_to_mission('***Going to home***', -35.36326, 149.16522, 1, 5)

def any_parameter_callback(self, attr_name, value):

  if attr_name =='AHRS_GPS_USE':
     print "%s changed to: %s. UAV is being jammed." % (attr_name, value)
     print "***Disable GPS_FAILSAFE***"
     uav.parameters['GPS_FAILSAFE'] = 0
     print "Checking if UAV is in autonomous flight mode"
     if uav.parameters['AHRS_GPS_USE'] == 0.0:
        if uav.mode == 'GUIDED':
           go_home_autonomously()
        else:
           print "UAV is in manual flight mode"
           print "***ALERT ON RC CONTROLLER***"
           user_auto_mode = int(raw_input('Do you want to fly home autonomously(y-n)?'))
           if user_auto_mode== 'y':
              go_home_autonomously()
           else:
              print "***ALERT ON RC CONTROLLER***"
     else:
        print "***Spoofing is no longer present***"
        print "***Enable GPS_FAILSAFE***"
        uav.parameters['GPS_FAILSAFE'] = 1
        go_to_mission('***CONTINUING TO THE MISSION***', -35.26326, 49.16522, 1, 5)

uav.parameters.add_attribute_listener('*', any_parameter_callback)

print "Set airspeed to %s" % global_airspeed
uav.airspeed = global_airspeed

print "Set groundspeed to %s" % global_groundspeed
```

```
uav.groundspeed = global_groundspeed

go_to_mission('***START TO A MISSION***', -35.26326, 149.16522, 1, 10)

uav.parameters['AHRS_GPS_USE'] = 0
uav.parameters['AHRS_GPS_USE'] = 1

uav.mode = VehicleMode("RTL")

print "Close uav object"
uav.close()
```

4. Tests and Results

4.1. Test Conditions

In this section, various tests cases are given and tests are conducted accordingly to the test cases. All tests are performed against Ardupilot Sitl simulator. Firstly, test case flights are performed without an attack to GNSS signal and later on it is assumed that attack to GNSS signal is detected in the target point. Additionally, all tests are repeated ten times with attack and without attack and mean values are given in the results. The test conditions and general simulator settings are given in Table 7.

Table 7. General test conditions

Weather	Open air, optimum conditions that UAV can operate
Turbulence	Is added in certain test cases
Obstacles	No obstacle present in flight area
Radio link latency	No latency
Pressure due to height	Ignored
UAV head position	Always placed facing against to target point
Target point	Target is assumed to be a location where GPS jammer/spoofer present
Number GPS satellites	It is simulated that there are six satellite locks

Additionally, each test details, configurations and results are given in the relevant test sections.

4.2. Test and Results

Test 1

In this test case, simulation environment and UAV is configured with the following parameters as given in the Table X. Basically, a flight mission set using only one axis; X and UAV is expected to fly 100 seconds to target point on a straight line. On the target point, it is simulated by code that GPS spoofing/jamming is detected, relevant functions triggered, and UAV is expected to fly initial point.

Table 8. Test 1

Turbulence	None
Flight Axis	X
Flight Duration	100 seconds
Number of yaws	0
Ground Speed	10 m/sn
Altitude	10 m

Table 9. Test 1 results

	Landed distance relative to home X axis (m)	**Landed distance relative to home Y axis (m)**
DR disabled	2.3	2.0
DR disabled + random turbulence	2.4	2.3
DR enabled	2.9	2.8
DR enabled + random turbulence	3.1	3.0

Test 2

In this test case, simulation environment and UAV is configured with the following parameters as given in the Table X. Basically, a flight mission set using only two axis; X and Y. UAV is expected to fly 100 second to target point on a line. On the target point, it is simulated by code that GPS spoofing/jamming is detected and relevant functions triggered and UAV is expected to fly initial point.

Table 10. Test 2

Turbulence	None
Flight Axis	X, Y
Flight Duration	100 seconds
Number of one whole yaw turn	In random times and random order 5 to right 5 to left full yaw
Ground Speed	10 m/sn
Altitude	10m

Table 11. Test 2 results

	Landed distance relative to X axis (m)	**Landed distance relative to Y axis (m)**
DR disabled	2.3	2.5
DR disabled + random turbulence	2.2	2.1
DR enabled	4.8	5.1
DR enabled + random turbulence	5.3	5.5

Test 3

In this test case, simulation environment and UAV is configured with the following parameters as given in the Table X. Basically, a flight mission set using X,Y and Z axis. UAV is expected to fly 100 seconds to target point. On the target point, it is simulated by code that GPS spoofing/jamming is detected and relevant functions triggered and UAV is expected to fly initial point.

Table 12. Test 3

Turbulence	None
Flight Axis	X, Y, Z
Flight Duration	100 seconds
Number of yaws	In random times and random order 5 to right 5 to left full yaw
Ground Speed	10 m/s
Altitude	10m

Table 13. Test 3 results

	Landed distance relative to X axis (m)	**Landed distance relative to Y axis (m)**
DR disabled	2.3	2.1
DR disabled + random turbulence	2.2	2.5
DR enabled	6.4	6.8
DR enabled + random turbulence	8.3	8.7

Test 4

In this test case, simulation environment and UAV is configured with the following parameters as given in the Table X. Basically, a flight mission set using X,Y and Z axis. UAV is expected to fly 100 seconds to target point. On the target point, it is simulated by code that GPS spoofing/jamming is detected and relevant functions triggered and UAV is expected to fly initial point.

Table 14. Test 4

Turbulence	None
Flight Axis	X, Y, Z
Flight Duration	100 seconds
Number of yaws	0
Ground Speed	Randomly changed during the flight between 1-10 m/s
Altitude	10m

Table 15. Test 4 results

	Landed distance relative to X axis (m)	**Landed distance relative to Y axis (m)**
DR disabled	2.1	2.4
DR disabled + random turbulence	2.1	2.2
DR enabled	7.6	8.2
DR enabled + random turbulence	10.2	12

Test 5

In this test case, simulation environment and UAV is configured with the following parameters as given in the Table X. Basically, a flight mission set using X,Y and Z axis. UAV is expected to fly 100 seconds to target point. On the target point, it is simulated by code that GPS spoofing/jamming is detected and relevant functions triggered and UAV is expected to fly initial point.

Table 16. Test 5

Turbulence	None
Flight Axis	X, Y, Z
Flight Duration	100 seconds
Number of yaws	In random times and random order 5 to right 5 to left full yaw
Ground Speed	Randomly changed during the flight between 1-10 m/s
Altitude	10m

Table 17. Test 5 results

	Landed distance relative to X axis (m)	**Landed distance relative to Y axis (m)**
DR disabled	2.0	2.2
DR disabled + random turbulence	2.1	2.3
DR enabled	16.2	14.5
DR enabled + random turbulence	19.2	21.4

5. Conclusion

This study focused on dead reckoning implementation on a mini UAV to mitigate GNSS signal spoofing-jamming attacks. Addition to this, the thesis tried to provide answers the questions as: Is it possible to save UAV and send it to its home location when there is such attack to UAV's navigation system. Moreover, it was researched if IMU based DR can be implemented on a mini UAV and how this kind of DR can be used on UAV in case of a jamming or spoofing attack to GNSS signal.

To be able to answer these questions, related works were reviewed and it was identified that current related works were lacking post incident activities when GNSS signal is not available for any reason. Later on, it was proposed that this issue can be resolved using IMU based DR implementation on a mini UAV. In order to realize this proposal, GNSS overview, GNSS signal level attacks, UAS systems architectures and applications were studied. Additionally, legal aspects of jammers and UAS usages were overviewed according to USA's regulations.

With this kind of a technical background study provided a basis to implement proposed solution. Since this was going to be an empirical study, a test environment is setup. Solution is implemented using Drone-Kit SDK and resulting code is executed against the simulator in the test environment. Additionally, sever test cases were prepared to observe IMU based DR's performance and results on the mini UAV.

The test results, comments and possible future work is giving in the following relevant sections.

5.1. Validation and Test Results

Validation was done against test environment as stated in the previous sections. However, this simulation setup is the exact same mimics of the real latest version of 3DR Solo UAV. As, I don't have flight certificate to fly the UAV and as it is not containing the latest firmware, the tests were not conducted on the real UAV.

Nevertheless, resulting code can be tested against on a real 3DR Solo UAS once I have the UAV with latest firmware setup. Moreover, same proposed logic can be used any kind of UAS which has Ardupilot autopilot system.

In order to test and validate DR implementation, five test cases were prepared. Each tests repeated with random turbulence in order to see its effect on DR based navigation. Tests aims are giving as follows:

5.1.1. Test Case One

In this case, it was investigated that what would be the displacement error according to home location when GNSS signal is not available and DR is enabled. It was found that, when UAV is flying with a constant speed on a one axis either X or Y with our any yaw, displacement error is not very far from GPS. It is just ~0.7m (relative to landing point) more than the GPS error. When there is random turbulence added this error goes around ~1.5m (relative to landing point) meaning that turbulence almost doubles the error in these conditions.

5.1.2. Test Case Two

This case has same setup with test case one but with two main differences; flight axis and yaws. It was observed that yaws and multiple axises also contributes to error and the extracted error was ~2.75m (relative to landing point). Additionally turbulence also affected the error as given in the relevant test result.

5.1.3. Test Case Three

This case repeats the case number three with additional Z axis. It was observed that this didn't affect the error dramatically. Error increased to ~4m (relative to landing point). However, turbulence affected a bit more compared to the other cases and error went up to ~6m (relative to landing point).

5.1.4. Test Case Four

In this case, yaw effect was removed and ground speed effect on error was tested. It was found that randomly speed changes on the UAV greatly affect the error and it was around ~6m (relative to landing point).

5.1.5. Test Case Five

This case simulated a random flight with various yaws and random ground speed. It was observed that, all these variations increased the error dramatically. The error went up to 18-20m (relative to landing point).

In all these test cases, although UAV was not successfully landed on exact same home location, it landed relatively close to home location with certain errors. It can be concluded that, IMU based DR implementation is effective on simple flight patterns. However, when the flight pattern gets more complicated, the error increased dramatically. Nevertheless, this solution can be used on UAV. By this way, at least UAV may possibly go out of jamming/spoofing area and get more close to home location.

5.2. Future Work

As future work, firstly, I predict that UAV can be send to initial flight position by using RC channel overrides which means RC controller channel input can be put into stack and later on can be popped and executed again to reverse the flight.

Secondly, since this study contains a functional flight simulator, test cases can be increased and more control conditions can be added to observe the effect on the displacement error.

Moreover, in some applications, UAV's home location can be a moving platform such as a ship. It can be also studied how to send UAV back home on a moving platform in case of GNSS signal loss.

References

[1] Canis, Bill. "Unmanned Aircraft Systems (UAS): Commercial Outlook for a New Industry." Congressional Research Service (2015): 7-5700.

[2] Santiago Andres, “Detection solution analysis for simplistic spoofing attacks in commercial mini and micro UAVs ”, M.S thesis, Dept. Computer Science, Tallinn University of Technology, Tallinn, Estonia, 2016

[3] Kerns, Andrew J., et al. "Unmanned aircraft capture and control via GPS spoofing." Journal of Field Robotics 31.4 (2014): 617-636.

[4] 3DR, "3DR uBlox GPS with onboard compass”, Internet: https://3dr.com/wp-content/uploads/2013/08/3DR-uBlox-GPS-web-version.pdf, [November 1, 2016]

[5] Parrot, "AR Drone 2/GPS”, Internet: http://wiki.paparazziuav.org/wiki/AR_Drone_2/GPS, [November 17, 2016]

[6] Thiel Andreas and Michael Ammann, “Anti-Jamming techniques in u-blox GPS receivers”, 2009

[7] Ublox, "GNSS Firmware 3.01 for u-blox 8/M8 Standard,”, Internet: https://www.u-blox.com/sites/default/files/GNSS-FW3.01_ReleaseNotes_(UBX-16000319)_Public.pdf, [Oct 17, 2016]

[8] Kim Hartman and Keir Gilies, “UAV exploitation: A new domain for cyber power”, 2016

[9] Battelle, “Battelle DroneDefender”, Internet: http://www.battelle.org/our-work/national-security/tactical-systems/battelle-dronedefender, [November 17, 2016]

[10] Ojeda, Lauro, and Johann Borenstein. "Non-GPS navigation with the personal dead-reckoning system." Defense and Security Symposium. International Society for Optics and Photonics, 2007.

[11] Beauregard, Stéphane. "A helmet-mounted pedestrian dead reckoning system." Applied Wearable Computing (IFAWC), 2006 3rd International Forum on. VDE, 2006.

[12] Jimenez, Antonio R., et al. "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU." Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on. IEEE, 2009.

[13] Srinivasan, Mandyan V., Javaan S. Chahl, and Shao-Wu Zhang. "Robot navigation by visual dead-reckoning: inspiration from insects." International Journal of Pattern Recognition and Artificial Intelligence 11.01 (1997): 35-47.

[14] Nagatani, Keiji, et al. "Improvement of odometry for omnidirectional vehicle using optical flow information." Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on. Vol. 1. IEEE, 2000.

[15] Guizilini, Vitor, and Fabio Ramos. "Visual odometry learning for unmanned aerial vehicles." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.

[16] Lykken, L., and E. Schulze. "Dead reckoning back-up navigational system for a drone." U.S. Patent No. 3,749,333. 31 Jul. 1973.

[17] Layh, Trevor, et al. "GPS-Denied Navigator for Small UAVs." (2014).

[18] Omari, Sammy, and Guillaume Ducard. "Metric visual-inertial navigation system using single optical flow feature." Control conference (ECC). 2013.

[19] NASA, “Sputnik and The Dawn of the Space Age”, Internet: https://history.nasa.gov/sputnik/, [March 6, 2017]

[20] David Hoffman, “Sputnik Mania”, Internet: http://www.ted.com/talks/david_hoffman_shares_his_sputnik_mania, [March 6, 2017]

[21] Guier, William H., and George C. Weiffenbach. "Genesis of satellite navigation." Johns Hopkins APL technical digest 19.1 (1998): 15.

[22] Trimble, “The First Global Navigation Satellite System”, 2007

[23] Layh, Trevor, et al. "GPS-Denied Navigator for Small UAVs.", 2014

[24] Thompson, Richard B. "Global positioning system: the mathematics of GPS receivers." Mathematics magazine 71.4 (1998): 260-269.

[25] Trimble, “The First Global Navigation Satellite System”, 2007

[26] Col Steve Whitney, "Update on GPS Modernization Efforts”, Internet:http://www.gps.gov/governance/advisory/meetings/2015-06/whitney.pdf, [November 22, 2016]

[27] Mike Gruss, “Launch of First GPS 3 Satellite Now Not Expected Until 2017”, Internet: www.iceengg.edu/staff.html, [Nov. 24, 2016]

[28] “Unmanned Aerial Vehicles an Overview”, 2008, p. 2, at http://www.insidegnss.com/auto/janfeb08-wp.pdf

[29] Gupta, Suraj G., Mangesh M. Ghonge, and P. Jawandhiya. "Review of unmanned aircraft system (UAS)." *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 2.4 (2013).

[30] FCC, "Jammer Enforcement”, Internet: https://www.fcc.gov/general/jammer-enforcement [March 11, 2017]

[31] FCC, "GPS, Wi-Fi, and Cell Phone Jammers Frequently Asked Questions (FAQs)”, Internet: https://transition.fcc.gov/eb/jammerenforcement/jamfaq.pdf [March 11, 2017]

[32] Pelechrinis, Konstantinos, et al. "Lightweight jammer localization in wireless networks: System design and implementation." *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009.

[33] Liu, Zhenhua, et al. "Exploiting jamming-caused neighbor changes for jammer localization." *IEEE Transactions on Parallel and Distributed Systems* 23.3 (2012): 547-555.

[34] Cheng, Tianzhen, Ping Li, and Sencun Zhu. "An algorithm for jammer localization in wireless sensor networks." *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*. IEEE, 2012.

[35] Bennett, Wells C. "Civilian drones, privacy, and the federal-state balance." *The Brookings Institution. September* (2014).

[36] Matiteyahu, Taly. "Drone regulations and Fourth Amendment rights: the interaction of state drone statutes and the reasonable expectation of privacy." *Colum. JL & Soc. Probs.* 48 (2014): 265.

[37] FAA, "New Requirements for Registering and Marking Small Unmanned Aircraft", Internet: https://www.faa.gov/documentLibrary/media/Notice/N_8900.338.pdf, [March 13, 2017]

[38] Estonian CAA, "How to operate UA", Internet: https://www.ecaa.ee/en/how-operate-ua, [May 26, 2017]

[39] Estonian CAA, "Registri andmed Internet: https://www.ecaa.ee/et/lennundustehnika-ja-lennutegevus/ohusoidukite-register/registri-andmed, [May 26, 2017]

[40] Eric Hal Schwartz, "Virginia Startup Sells Drone Security to Turkey's Prime Minister", Internet: http://dcinno.streetwise.co/2016/12/21/dc-tech-drone-startup-droneshield-saving-turkey-prime-minister/, [March 13, 2017]

[41] Theguardian, "Eagles v drones: Dutch police to take on rogue aircraft with flying squad" Internet: https://www.theguardian.com/world/2016/sep/12/eagles-v-drones-dutch-police-take-on-rogue-aircraft-flying-squad, [March 13, 2017]

[42] Theguardian, "China unveils laser drone defence system ", Internet: https://www.theguardian.com/world/2014/nov/03/china-unveils-laser-drone-defence-system, [March 13, 2017]

[43] 3DR Robotics, "Solo Specs: Just the facts", Internet: https://news.3dr.com/solo-specs-just-the-facts-14480cb55722#.fzv9i5u29, [March 14, 2017]

[44] 3DR Robotics, "Architectural Overview", Internet: https://dev.3dr.com/concept-architecture.html, [March 14, 2017]

[45] 3DR Solo Pixhawk 2 Overview ", Internet: https://www.youtube.com/watch?v=LY8FpvaccN0 [March 15, 2017]

[46] "Pyhton", Internet: https://www.python.org/, [March 17, 2017]

[47] "About Dronekit", Internet: http://python.dronekit.io/about/overview.html, [March 17, 2017]

[48] "Mavlink", Internet: http://qgroundcontrol.org/mavlink/start, [March 18, 2017]

[49] "Setting up a simulated vehicle (STIL)", Internet: http://python.dronekit.io/develop/sitl_setup.html, [March 18, 2017]

[50] “Virtual environments”, Internet: http://docs.python-guide.org/en/latest/dev/virtualenvs/, [March 21, 2017]

[51] “MAVProxy”, Internet: http://ardupilot.github.io/MAVProxy/html/index.html, [March 23, 2017]

[52] “PyCharm”, Internet: https://www.jetbrains.com/pycharm/, [March 25, 2017]