A Thesis

entitled

GNSS/GPS Navigation and Related Attacks Implementation in an Open Source UAV Simulation Testbed (UAVSim) using Open Source Satellite Simulator (OS3)

by

Farha Jahan

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Masters of Science Degree in Engineering

Dr. Weiqing Sun, Committee Chair

Dr. Mansoor Alam, Committee Member

Dr. Hwang Ho, Committee Member

Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo
May 2015

An Abstract of

GNSS/GPS Navigation and Related Attacks Implementation in an Open Source UAV Simulation Testbed (UAVSim) using Open Source Satellite Simulator (OS3)

by

Farha Jahan

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Masters of Science Degree in Engineering

The University of Toledo
May 2015

Unmanned systems or remotely piloted vehicles are of great importance in accomplishing task where human lives would be at risk. These systems are being deployed in areas which would otherwise be time-consuming, expensive and inconclusive if done on foot or by human intervention. The three major classes of these systems based on their operational environment are air, ground and underwater unmanned vehicles. Clearly, in terms of causing damage, unmanned aerial vehicles (UAVs) are the most efficient and have been known to change the course of several recent wars. If the security of these systems are compromised, it will pose a great threat to human lives as well as the nation. Therefore, it is important to analyze different types of possible attacks that can be attempted on these systems. Even today, Federal Aviation Administration (FAA) has limited the use of UAVs in the US National Airspace (NAS), primarily, due to the threat to general population. This makes real world testing difficult in an academic setup. The best solution to this problem is to have a simulation based environment where different operational scenarios, related cyber-attacks and their impacts on UAVs can be studied. The software based simulators are very economical to test different features of a UAV in terms of various defense mechanisms against cyber-attacks. In this thesis, we enhance UAVSim, a simulation test-bed for UAV Network cyber-security analysis developed by our team, with addition of

Global Navigation Satellite System (GNSS) and Global Positioning System (GPS). The test-bed allows users to easily experiment by adjusting different parameters for the satellite, UAV and attack host, and implement different kinds of attacks. In addition, each UAV host works on well-defined mobility framework, radio propagation models, etc., resembling real-world operational scenarios.

[To my Parents for their hard work and dedication to raise us to this level and higher in all odds.]

Acknowledgments

Few words of acknowledgment is not enough to express my gratitude towards every person who has directly or indirectly influenced and motivated me towards accomplishing my Master's degree. I would like to thank my adviser Dr. Weiqing Sun without whose leadership and guidance this accomplishment would have taken much longer. I would also like to thank my co-adviser and Department chair, Dr. Mansoor Alam and my thesis committee member Dr. Hwang Ho, for taking time out of their busy schedules to serve on my evaluation committee and give valuable comments and suggestions to improve my work.

My family has always been my pillar of support and encouragement. I am so grateful to them. I would also like to thank Ahmad Yazdan Javaid. He has been my friend, philosopher and guide. My journey would be 'all work and no play', without my beloved friends Sami, Niyaz, Salma, Rubia and others. They have spiced up my life with healthy discussions, study and fun nights, trips, celebrations and compliments.

I would like to thank COGS for providing me graduate assistantship. It was pleasure working with Dr. Patricia Komuniecki on occasions and all the COGS staff, especially my supervisor Teresa Hayez.

My gratitude would be incomplete without thanking Cheryl, Eric, Christy, John and Michelle who welcomed me as their family. Last but not the least, I would like to thank my aunt Nikhat and cousins Dr. Zafar Hassan, Nusrat Hassan and Ishrat Hassan for being there for me in a land away from home.

Contents

List of Tables

List of Figures

List of Abbreviations

ADS-B	Automatic Dependent Surveillance - Broadcast
C3UV	Center for Collaborative Control of Unmanned Vehicles
DDoS	Distributed Denial of Service
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
GCS	Ground Control Station
GSCS	Galileo Satellite Communication Simulator
GSSF	Galileo System Simulation Facility
LOS	Line of Sight
MSSE	Multi-Scale Satellite Simulation environment
NORAD	North American Aerospace Defense Command
Open-SESSAME	Open-Source, Extensible Spacecraft Simulation and Modelling Environment Framework
OS3	Open Source Satellite Simulator
PNT	Position Navigation and Time
SNACS	Satellite Navigation Radio Channel Signal Simulator
SPEEDES	Synchronous Parallel Environment for Emulation and Discrete Event Simulation
TLE	Two-Line Element
UAV	Unmanned Aerial Vehicle
RAIM	Receiver Autonomous Integrity Monitoring

Chapter 1

Introduction

1.1 Unmanned Aerial Vehicles

There has been a remarkable growth of UAV use in various military and civilian application domains, from war-front, surveillance, reconnaissance, etc., to areas like agricultural imaging, traffic monitoring, cartography, package delivery, etc. In areas where human reach is difficult or limited and/or lives are at risk, UAVs can play an important role. For example, these systems can be sent to distant or inaccessible planetary bodies for research (e.g., Philae aircraft landed on a comet after a 10 year journey [44]) or to detect and survey real-time catastrophes like earthquake [1] and forest fires [18]. UAVs have found their use in applications like proposed pizza delivery (Pizza Hut) [28], proposed local package delivery as well as hiring drone pilots for tests (Amazon) [16, 47], agricultural chemical deployment [6], ecological surveys [5], natural event monitoring (Hurricane Hunter [26]), disaster management [41] and humanitarian response (e.g., damage assessment, search and rescue operations, dropping relief supplies in case of emergency [19]), 3-D Mapping and photogrammetry [56], wildlife protection [24], etc.

With increased importance of unmanned systems in the military domain due to their positive impact on human effectiveness and safety, their use in other applications

have only been increased. Commercial UAVs are proposed to get permits from FAA by 2015 and around 7500 are expected to be seen in air by 2020 [26] compared to the number being negligible in 2014. These have also promoted research in industries and academia. Even today, FAA doesn't require people to obtain license for drones which are used for recreational purpose, but they do limit its usage up to a height of 400feet and away from airport and air traffic [17]. Availability of low-cost mini-UAVs and DIY drones have promoted individual use and research as well. Constructing a drone is nowadays possible in as less as $300 at home by purchasing separate parts from online stores [40]. Non-requirement of license for such cases and easy construction of a drone poses a huge threat to the general population by its possible malicious usage, especially, if their navigational equipment is not secure.

Although UAV development started in 1960s, primary objectives of research has been its mission-accomplishment capability, reliability, and efficiency in terms of time and power. The most important aspects of issues related to cyber-security are vulnerability, breach and threat identification; and corresponding attack prevention, mitigation and recovery. Very few works focus on the security of these unmanned systems, which primarily focus on the causes and methods of security breaches at the lower-level system components. The need of a simulation testbed which can simulate single or multi-UAV behavior and provide a realistic response in case of an attack, served as our primary motivation. It is clear that navigation is one of the vital aspects of unmanned systems, therefore, its availability is significant. The secondary goal of providing the academia with inexpensive (with respect to cost as well as time) mode of performing these simulations which are near-real world, was also accomplished simultaneously.

1.2 Role and Importance of Navigation

To understand the importance of navigation in any task, it is necessary to understand what role it plays. Knowing the task area in advance clearly gives an edge to personnel and ensures safety of all involved. Aircraft, gunship and UAV operators already suffer with stress and performance pressure due to the responsibility of human lives. Accurate and reliable navigation data reduces mission related anxiety, impact on their mental health and optimal operational performance of the 'human-machine team' [53].

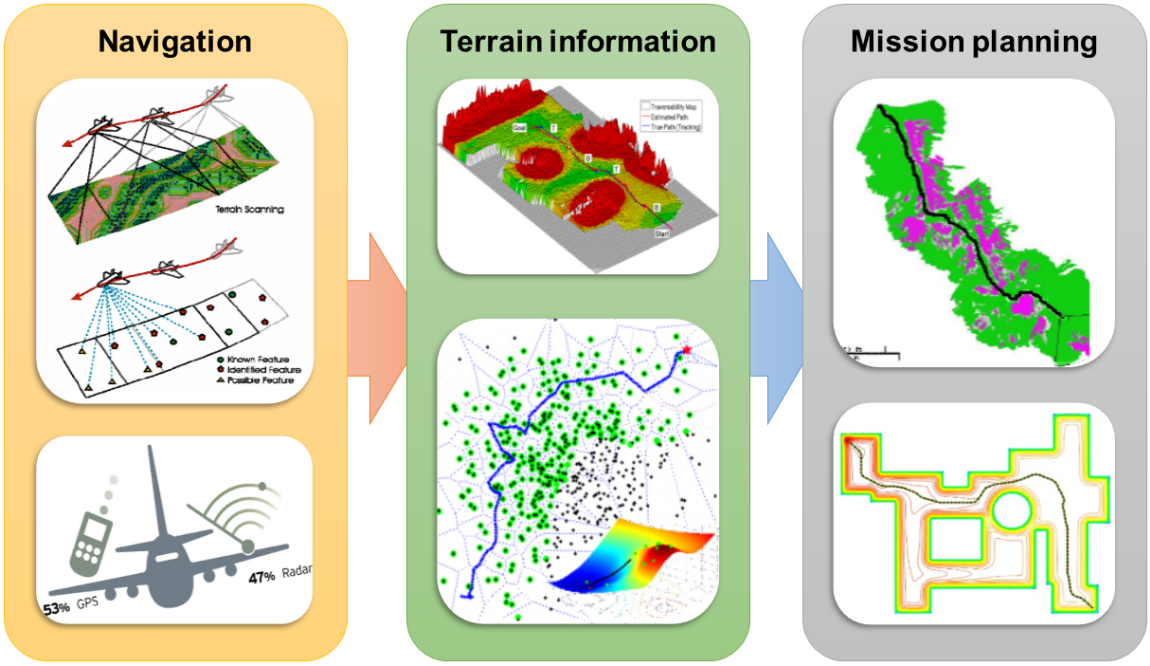

Figure 1-1: Role of navigation in UAV operations

Figure 1-1 depicts role of navigation in UAV operations. Other roles navigation plays in defense operations include but are note limited to:

- Prior terrain mapping and safety zone information
- Mission reconnaissance and surveillance

- Real-time navigation in unknown indoor/outdoor environment
- Enemy location awareness and safe path planning
- Reducing civilian casualties through prior planning

1.3 Recent Attacks and Failures

With emerging technology, drones are no more limited to practical or experimental purposes. They can be used by hobbyists, pranksters and troublemakers as well. This increases threat on public areas and an adverse use of cheap technology. After the Iran's claim on RQ-170 capture, in-depth study of vulnerabilities of UAVs where made which explains how easily it can be compromised and attacked. In 2012, North Korea launched a GPS Jamming attacks on the border of South Korea which disrupted navigation of aircrafts, ships and in-car navigation [20]. The paper [39] discusses about recent attacks on UAVs that where reported while [13] reports failures of 49 large drones since 2001. The figure 1-2 shows UAV crashes of severe category and Pentagon's plan to extend the operation base to 110 in 39 US states.

The paper [27] evaluates the risk and vulnerability of UAV for cyber-attacks based on the components that makeup the UAV architecture like the environment, communication links, data storage, fault handling mechanisms etc. As a report [9] published in 2001, list the causes of failure for UAVs where 'Insufficient testing before purchase' is common to all failed programs and a lot of money invested in these programs went in vain. If we have a proper testbed to test these systems before flight over civil lands, midair collision or ground casualty can be prevented and hence the loss of human lives and large investments in failed experiments.

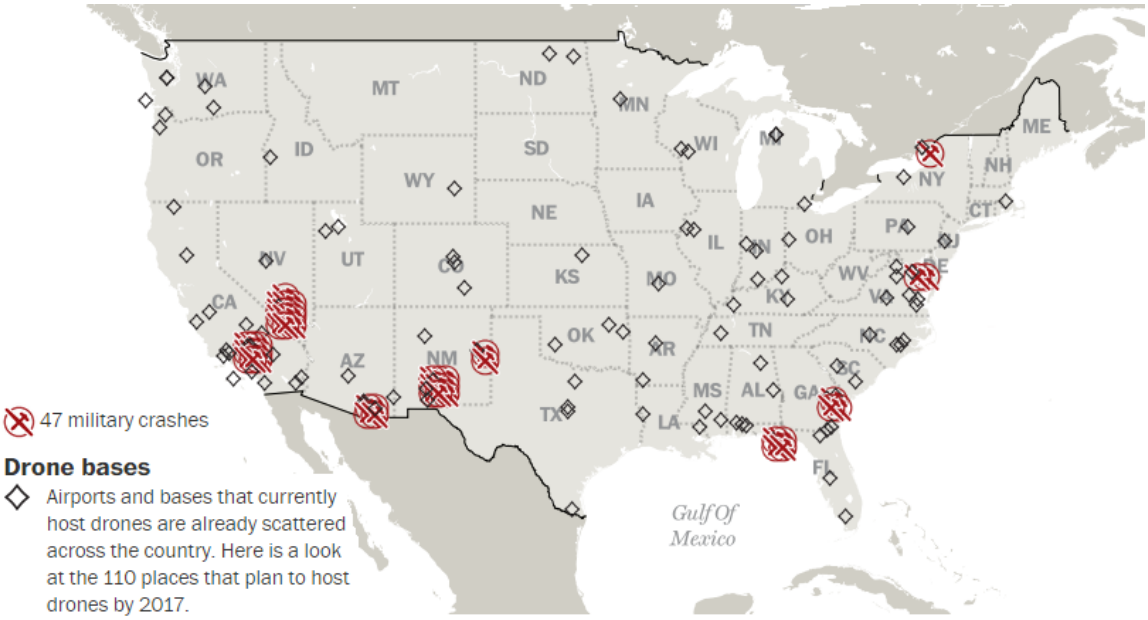

Figure 1-2: Figure demonstrates the 47 Class A UAV crashes between 2001 and 2013 and the drone operations to expand 110 bases [13].

1.4 Goal and Objectives

Although UAV development started in 1960s, primary objectives of research has been its mission-accomplishment capability, reliability, and efficiency in terms of time and power. The most important aspects of issues related to cyber-security are vulnerability, breach and threat identification; and corresponding attack prevention, mitigation and recovery. Very few works focus on the security of these unmanned systems, which primarily focus on the causes and methods of security breaches at the lower-level system components. The need of a simulation testbed which can simulate single or multi-UAV behavior and provide a realistic response in case of an attack, served as our primary motivation. It is clear that navigation is one of the vital aspects of unmanned systems, therefore, its availability is significant. The secondary goal of providing the academia with inexpensive (with respect to cost as well as time) mode of performing these simulations which are near-real world, was also accomplished

simultaneously.

This work focuses on enhancing our testbed to include the navigation module so that the next level of navigation related attacks can be simulated. The main component of a navigation module is its GPS unit which helps it to know its actual position. It also helps the UAV during different preset navigation modes, such as, position hold, return-to-home, autonomous flight and object avoidance [4]. Depending on the area in which the UAV is deployed, GPS can be used to link data to its spatial position. This method is termed as geo-referencing [7]. UAVs have found their application in various fields and operations, primarily, due to their navigation capability. The GPS signals are typically very weak, less than 100W and are transmitted over a range of $20,000 - 25,000$ kilometers. This makes them fall below the noise floor spectrum when they reach the earth's surface [63]. These signals are vulnerable to failure, disruption and interference from unintentional or deliberate sources. Clearly, navigation is one of the most important module of a remotely controlled UAV or when the UAV is not in the sight of an operator. The increased dependency of UAVs on GPS signals for localization, navigation and time-synchronization has made it a focus area for adversaries and thus led to discovery of its vulnerabilities to attacks like Spoofing and Jamming. Simulations related to UAV operations involving GPS and related navigational aspects is quite important for correct attack simulations as well as other general simulations.

1.5 Organization of Thesis

Several works in the past have focused on single UAV dynamics simulation for model development and testing various enhancements in these existing models to make new ones. We discuss more about related works in the area of UAV testbed development and studies related to GPS attacks on UAVs in Section 2.1 and previous

works done by our team in Section 2.6. Section 3 introduces GPS and presents our implementation of GNSS and GPS and the GPS related attacks. Section 4 presents the design, architecture, operation and attacks performed on the UAVs by our enhanced testbed, UAVSim. Section 5 presents the simulation analysis and results, and the paper is finally concluded in Section 6.

Chapter 2

Related Work

2.1 Literature Survey

This section discusses the works that have been done in the area of UAV and GPS simulation. Out of these, many focus on modeling of a single UAV in a closed lab or in an open but controlled environment to improve the system performance, flight range and usability, without necessarily implementing cyber-attacks, its impact and related risks. A lack of close to real simulation of a UAV Network (UAVNet) including UAVs, Satellites, Ground Control Stations (GCS) and adversaries, was noticed. An ideal testbed should allow inter and intra-component communication, and component level behavioral analysis. Although many of these simulations implement GPS device or a GPS software simulator to generate Position, Navigation and Time (PNT) data signals, none of them have designed a working close-to-real GPS or GNSS system. We present a classification of these existing simulators based on presence of GPS implementation. We also discuss some UAV-independent GPS/GNSS implementations available.

2.2 UAV Simulators without GPS Implementation

The paper [42] mentions GPS spoofing but the implementation was out of scope while [43] uses carrier phase of GPS signals through GPS receivers to obtain altitude and positional measurement. A visual training simulator based on mission equipment is discussed in [73] which trains the operator as well as tests the subsystems. GUI based simulation testbed used UAVs equipped with Piccolo II auto-pilots along with other hardware to demonstrate a wide range of information-oriented applications [54]. Another testbed was developed for wireless networks on small UAVs to analyze its monitoring architecture and parameters such as delay, throughput, range, etc [8]. All of these simulators and testbeds either works in a way in which GPS is not required or employs an external GPS device or software.

Another work, focused on simulation of a swarm of UAVs, LaBRI involves deployment of actual UAVs on a field for specific applications and check their survivability [8]. Software based simulators are quite economical in testing different features of a UAV in terms of various defense mechanisms against cyber-attacks. More software based network simulation systems were developed for a swarm of UAVs [12], [25], but these involved use of laptops or other hardware as UAVs. Two more important simulation testbeds for such swarms of UAVs were also developed, SPEEDES (Synchronous Parallel Environment for Emulation and Discrete Event Simulation) [10, 11] and *C3UV* (Center for Collaborative Control of Unmanned Vehicles) [54]. SPEEDES simulates a swarm of UAVs on a high performance parallel computer in order to match the actual speed and communication rate of the UAV network and *C3UV* testbed focuses on the fact that information acquisition through collaborative sensing and control are highly coupled. Two related works in the area of network security simulation are also quite different due to the unavailability of wireless security analysis through simulation incorporating node mobility. One of them, ARENA, was pro-

posed in 2007 and includes multi-level attack simulation in the network but does not focus on all layers as well as individual modules of vital network components [45], such as UAV in our case. The other one, Ordered Scenario based Network Security Simulator, was proposed in 2005 and has the same limitation of simulating only wired components [74].

2.3 UAV Simulators with GPS Implementation

A visual $3D$ flight simulation software based on Matlab and Simulink was an early attempt towards simulating UAV which used navigation module of FlightGear [55]. FlightGear provides a generic GPS support with GPS receivers yet to be implemented [71]. Another project called UAV Playground was developed in Java used FlightGear to receive GPS data and implemented GPS tracking in Google Earth [35]. In industry, aeronautical division of IDS Corporation has implemented an unmanned aerial vehicle simulator Hero UAVSim [29] composed of ground control stations (GCS) [30], UAV Simulator and a sensor payload simulator [31]. The UAV Simulator has GPS based auto tracking capability and GPS outage mitigation measures, while GCS has an integrated GPS receiver to determine its actual position.

2.4 Independent GPS Simulation

Several works have been done in the industry to simulate GPS and GNSS systems. LabSat simulator [46] is a low cost simulator which provides option of selecting from different GNSS such as GPS, GLONASS, Beidou, and Galileo. It generates genuine navigation signal that can be stored, replayed and used in different applications. Spirent implements GPS and GNSS through hardware (e.g. GSS9000 multi-frequency, multi-GNSS RF constellation simulator) as well as software (e.g. SimGEN,

SimAUTO, SimINERTIAL, etc.) to simulate navigation signals for professional, controllable and repeatable testing in the lab [62]. National Instruments (NI) has a GPS Simulator which can produce GPS signals of up to 12 satellites C/A codes for 24 hours for testing GPS receivers [32]. Many other simulators such as IFEN Inc. NavX-NCS Professional/Essential, CAST Navigation SGX GPS Satellite Simulator, AeroFlex Portable GPS/Galileo/SBAS Positional Simulator GPSG-1000, etc., also simulate GPS and GNSS but differ with respect to the range of signals produced and constellation implemented. All these devices are quite expensive and could not be considered as low-cost for a UAV simulation testbed developed in an academic setup.

2.5 GPS Security Related Works

In 2012, Todd Humphreys' and his research team successfully demonstrated that UAVs are vulnerable to GPS Spoofing and led it believe that it was rising up while deceiving it to fly downwards [15]. They have presented their work in [60] explaining how a civil GPS receiver can be easily spoofed. Several works has been published by the author in context of GPS Spoofing in gps receivers such as [65], [52]. In [3], effect of time accuracy has been assessed and time-based spoofing has been studied to demonstrate the vulnerability of time synchronization protocols to spoofing attacks while [64] discusses the requirements of successful GPS Spoofing Attacks.

2.6 Our Previous Work

Several works have been done during the study, design and development phases of our software based simulation environment UAVSim. As the beginning study phase, single UAV and overall UAVNet models were defined in order to establish a close to real representation of the system and its components. This representation would in

turn facilitate creation of a software model. Other contribution of this work included: analytical threat analysis, risk analysis, and attack impact evaluation using FlightGear simulation software [39]. During the second phase of development, UAVSim was developed using OMNeT++ and INET, and few cyber-attacks (Jamming and Distributed Denial of Service(DDoS)) were implemented. One of the major contribution of this phase was an interactive GUI for users [36]. Continuing the work to the next phase of enhancement and further validation, advanced features like multi-user support, server based centralized simulation, etc. were implemented and the testbed was shown to be performing reasonably with limited resource in a generic computing infrastructure for DDoS attack [38]. An extended performance analysis was then carried out using two resource intensive attacks, DDoS and Jamming, as well as UAV swarm simulations, to showcase the testbed's capability to simulate all other attacks which will consume less resources on the underlying computing infrastructure [2].

The work presented in this thesis is an extension of the development phase of the simulation testbed. The previous work [36] describes UAVSim in detail. The third phase proved capability of UAVSim to be used for swarm simulation along with its primary purpose of security simulations of UAVNet.

Chapter 3

GPS/SATNAV Overview

3.1 GPS Basics

A satellite navigation or SATNAV is a system of satellites that provides autonomous geo-spatial positioning with global coverage and is termed as Global Navigation Satellite System (GNSS). Satellite sends radio signal along a line of sight (LOS) to electronic receiver, which calculates its position. These receivers are called Global Positioning System or GPS receivers [58]. Currently, the US GPS and the Russian GLONASS are the only GNSS supporting global coverage. Other prospective GNSSs are Compass of China and Galileo of Europe. Any GNSS has three operational segments - the space segment, the ground/control segment and the user segment.

The GPS space segment consists of 32 satellites as of 7 October 2014, in six orbital planes [23]. The satellites operate at an inclination of 55 degrees to the equator and complete their revolution around the earth in 12 hours, i.e., two revolutions around the earth in one day. A GPS satellite sends two ranging codes: C/A, used for public usage, and Precision code, used by the military. It has five frequency bands $L1 - L5$ where $L1$ and $L2$ are used and $L5$ is reserved for Safety-of-Life. The signal consists of GPS data and time, ephemeris data and the almanac. $L3$ band is used for nuclear

Figure 3-1: Global Positioning System requires minimum of 24 satellites

weapon detonation, detection and treaty or ban enforcement. $L4$ is not used yet.

The ground/control segment consists of ground stations that track the GPS satellites, monitor their transmissions, perform analysis, and send commands and data to the constellation [23]. The ground network consists of one master control station, one alternate master station, 12 command and control antennas, and 16 monitoring sites.

The end-user antenna and GPS receiver makes the user segment which receives and calculates the position, velocity and time based on the data from the GPS Satellite. The receiver has to acquire and maintain lock on four satellites to get its accurate position in the $3D$ space.

3.2 Trilateration

Trilateration is the process of determining location or position of an object using the geometrical concepts of circles, spheres and triangles [72]. It is used in navigation, surveying and in GPS. The sensor nodes in mobile sensor networks are usually used in various applications to collect sensor data and location information for location

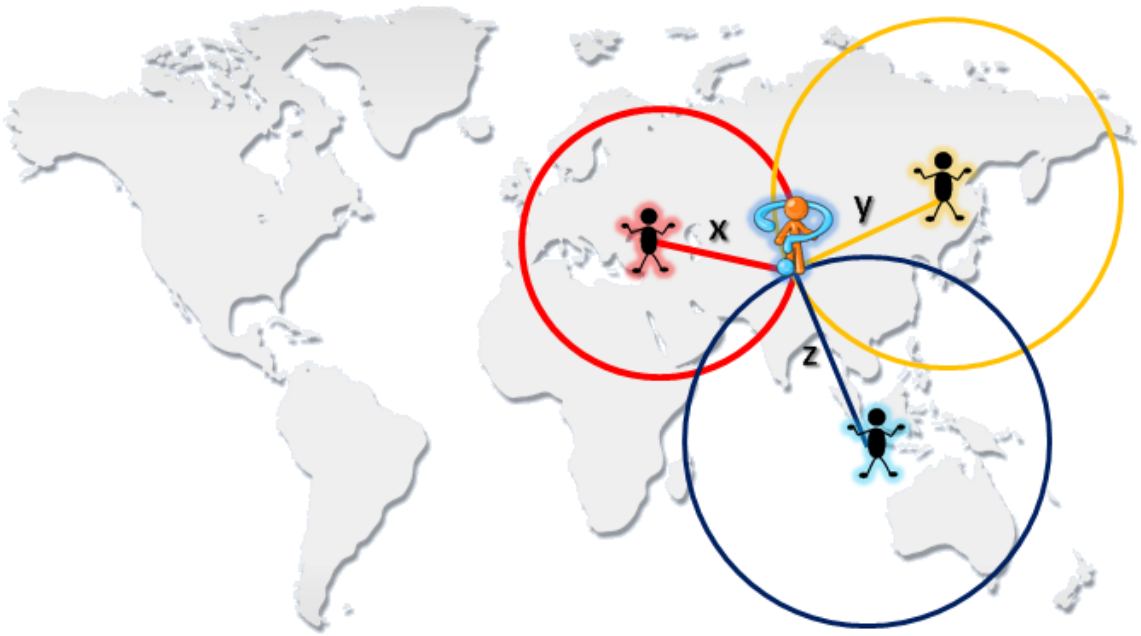

Figure 3-2: Trilateration

based services. To be more clear, suppose a man M standing on the world map has lost his location information. If a friend calls him and say that he is x distance away from place A and another says that he is y distance from place B, M can make two circles with A and B as centers and x and y as radii of the circles respectively. He can be at either of the two intersection points of the circles. If a third friend says that he is at a distance z from location C, then the intersection of the three circles gives his approximate location. This concept is called $2D$ Trilateration and is shown in Figure 3-2. Substituting friends with satellites and considering circles to be spheres, we can calculate our location in $3D$ space using what is called a $3D$ Trilateration. Many $2D$ and $3D$ trilateration algorithms have been proposed to determine location accurately. The mathematical expressions to calculate a position in $2D$ and $3D$ are discussed below as proposed in [59].

3.2.1 2D Trilateration

An existing localization algorithm has been described here, which uses the fundamental concept of trilateration technique in $2D$. The position of an object can be determined by solving the following equations:

$$(x - x_1)^2 + (y - y_1)^2 = (d_1)^2$$

$$(x - x_2)^2 + (y - y_2)^2 = (d_2)^2$$

$$(x - x_3)^2 + (y - y_3)^2 = (d_3)^2$$

The equations are converted to linear equations by subtraction and substitution.

$$2.(x_2 - x_1).x + 2.(y_2 - y_1).y = \alpha$$

$$2.(x_3 - x_1).x + 2.(y_3 - y_1).y = \beta$$

where

$$\alpha = (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2)$$

$$\beta = (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)$$

In $2D$ space, the position (x, y) is obtained by solving the following matrix operations:

$$x = f(d_1, d_2, d_3) = \frac{\begin{vmatrix} \alpha & 2Y_1^2 \\ \beta & 2Y_1^3 \end{vmatrix}}{\begin{vmatrix} 2X_1^2 & 2Y_1^2 \\ 2X_1^3 & 2Y_1^3 \end{vmatrix}}$$

$$y = g(d_1, d_2, d_3) = \frac{\begin{vmatrix} 2X_1^2 & \alpha \\ 2X_1^3 & \beta \end{vmatrix}}{\begin{vmatrix} 2X_1^2 & 2Y_1^2 \\ 2X_1^3 & 2Y_1^3 \end{vmatrix}}$$

where X_i[j] and Y_i[j] are referred to $(x_i - x_j)$ and $(y_i - y_j)$ respectively.

3.2.2 3D Trilateration

Similar to $2D$ trilateration, $3D$ trilateration uses the following equations.

$$2.(x_2 - x_1).x + 2.(y_2 - y_1).y + 2.(z_2 - z_1).z = \alpha$$

$$2.(x_3 - x_1).x + 2.(y_3 - y).y + 2.(z_3 - z).z = \beta$$

$$2.(x_4 - x_1).x + 2.(y_4 - y).y + 2.(z_3 - z).z = \gamma$$

where

$$\alpha = (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) - (z_1^2 - z_2^2)$$

$$\beta = (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) - (z_1^2 - z3^2)$$

$$\gamma = (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) - (z_1^2 - z3^2)$$

By solving the following matrix based equations, we get the value of (x, y, z) in $3D$ space.

$$x = f(d_1, d_2, d_3, d_4) = \frac{\begin{vmatrix} \alpha & 2Y_1^2 & 2Z_1^2 \\ \beta & 2Y_1^3 & 2Z_1^3 \\ \gamma & 2Y_1^4 & 2Z_1^4 \end{vmatrix}}{\begin{vmatrix} 2X_1^2 & 2Y_1^2 & 2Z_1^2 \\ 2X_1^3 & 2Y_1^3 & 2Z_1^3 \\ 2X_1^4 & 2Y_1^4 & 2Z_1^4 \end{vmatrix}}$$

$$y = g(d_1, d_2, d_3, d_4) = \frac{\begin{vmatrix} 2X_1^2 & \alpha & 2Z_1^2 \\ 2X_1^3 & \beta & 2Z_1^3 \\ 2X_1^4 & \gamma & 2Z_1^4 \end{vmatrix}}{\begin{vmatrix} 2X_1^2 & 2Y_1^2 & 2Z_1^2 \\ 2X_1^3 & 2Y_1^3 & 2Z_1^3 \\ 2X_1^4 & 2Y_1^4 & 2Z_1^4 \end{vmatrix}}$$

$$y = g(d_1, d_2, d_3, d_4) = \frac{\begin{vmatrix} 2X_1^2 & 2Y_1^2 & \alpha \\ 2X_1^3 & 2Y_1^3 & \beta \\ 2X_1^4 & 2Y_1^4 & \gamma \end{vmatrix}}{\begin{vmatrix} 2X_1^2 & 2Y_1^2 & 2Z_1^2 \\ 2X_1^3 & 2Y_1^3 & 2Z_1^3 \\ 2X_1^4 & 2Y_1^4 & 2Z_1^4 \end{vmatrix}}$$

where X_i^{j}, Y_i^{j} and Z_i^{j} are referred to $(x_i - x_j)$, $(y_i - y_j)$ and $(z_i - z_j)$ respectively.

3.3 GPS Receiver

As discussed above, GPS receiver is the user segment of the GPS system. The signal sent by satellite contains the ephimeris and almanac data along with time and the satellite code.

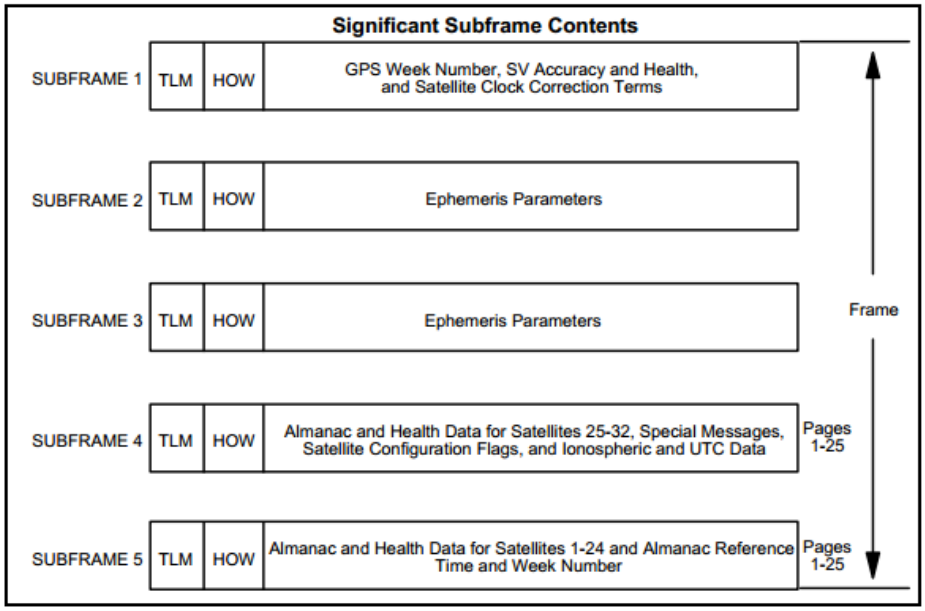

Figure 3-3: Navigation Message Content and Format Overview [21]

The figure 3-3 shows the sub-frames of the message. Ephemeris contain information about health of all GPS satellites and their exact position in the orbit. Almanac contain the coarse orbit and status information of each satellite, an ionosphere model and information to relate GPS derived time to Coordinated Universal Time (UTC). It takes around 25 frames and 12.5 minutes to transmit the complete almanac. Ephemeris data takes 18s to download with a speed of $50bps$. Therefore, when a GPS receiver is powered on for the first time, it downloads this data and obtains a lock on the $3(2D)$ or $4(3D)$ satellites available in its LOS. It takes around 45s to start. This is called the Cold Start. Once the lock is obtained, this data is updated every second.

The distance of satellite from receiver is calculated using the time taken by the satellite signal to reach the receiver and speed of light. Finally, it calculates its position using ephemeris and almanac data, and applying them to the aforementioned trilateration equations. It also takes into consideration the modeling errors, troposphere and ionosphere errors, and clock errors from the satellite.

3.4 Applications of GNSS/GPS

The GPS was originally invented for military usage and later was made available to the public. With the advancement in technology and devices that incorporates GPS, civilian usage has also increased nowadays. GPS device manufacturers like Garmin, and cellphone devices give very accurate location and time information with their 12 parallel channel receivers. Its application is not limited to transport services or location information. Listed below are some of the areas where the GNSS/GPS signals are used:

- Aviation industry uses GNSS signals for en-route navigation, automatic dependent surveillance - broadcast (ADS-B) in case of absence of radar signals, mid-air refueling, photogrammetry, etc.

- The road transport applications include in-car navigation, autonomous car driving, route guidance, traffic and lane control, trip travel information such as construction and traffic details, alternate routes, speed limits, etc.

- GPS ambulance and other vehicles can be used for emergency relief to gauge the level of disaster and find stranded motorists and victims. It can be used in emergency calls, rescue operations, crime prevention, tracking and stolen vehicle recovery.

- Marine transport services find GPS signals useful during ocean, coastal and in-land waterway navigation, automatic docking, cargo handling, dredging, automatic collision avoidance, vessel traffic services, etc.

- The rail transport industry uses GPS for signaling and train control, high speed warning, power supply control, door control supervision, level-crossing protection, etc. It also uses GPS for management services like fleet management,

cargo monitoring, etc. and as well as for passenger information system such as pre-trip and on-trip information.

- Outdoor sports like geo-caching, cycling and running employs GPS for measurement of distance, tracks and direction info.
- GPS finds its usage in scientific applications like surveying, meteorology and climate research, environmental and construction monitoring, global reference systems and geo-dynamics, etc.
- It can be used as a personnel protection and tracking aid for blind and handicapped civilians or people suffering from Alzheimer's disease, etc.
- Agricultural farms and fisheries use GPS for land area mapping, yield monitoring, planting, spraying, etc.
- Time sensitive applications like digital broadcasting, power generation and distribution, frequency/time calibration services and maintenance of time standards, etc., uses time data of the GPS signals.

3.5 GPS Vulnerabilities

Since GPS is low-cost and easily available worldwide, its application has found many uses as discussed in the above section. But this doesn't not make it invulnerable to attacks or failures. Rather it can more easily be subjected to intentional and unintentional threats. GNSS itself is vulnerable to natural causes or technical limitations. These vulnerabilities can generate incorrect signals and misleading PNT information which could be hazardous. In this section we discuss the vulnerabilities of GNSS/GPS, factors that impact GPS vulnerabilities and consequences.

3.5.1 Vulnerabilities of GNSS/GPS

Vulnerabilities of GNSS/GPS can be classified as: unintentional interference, intentional interference and human factors.

3.5.1.1 Unintentional Interference

Radio Frequency Interference- Various radio frequency signals from undesired sources are considered as interference. These RF waves are emitted by high power transmissions, television, mobile communication, mobile satellite services, and electronic devices. Other navigational systems or sensors such as RADAR, Tactical Air Navigation (TACAN), Pseudolites can cause unintentional interference which can be much stronger in strength. Distance Measuring Equipment (DME) and Automatic Dependent Surveillance (ADS) are other such systems that sends navigational information and can interrupt the GPS signals.

Intense Solar Radiations- As sun approaches the maximum part of its eleven year cycle, it emits high intensity solar flares and solar storms known as coronal mass ejections (CMEs). These flares and storms are burst of magnetic energy equivalent to millions of 100-megaton hydrogen bombs exploding at the same time. The radio waves produced across the entire electromagnetic spectrum reaches the earth and causes disruptions in the various signals including the GPS and other navigational systems. These events can also cause disruptions in the satellites which can temporarily shut down leading to widespread disruptions on the ground. One such event occurred in December 6, 2006 which affected the GPS system causing large number of receivers to stop tracking the GPS signals [51]. Similar events occurred in 2012 [22] and 2014 [33] but not much damage to the GPS system was reported.

Space Weather- It can be defined as conditions on the sun and in the solar wind, magnetosphere, ionosphere and thermosphere that can influence the performance and

reliability of space-borne and ground-based technological systems and can endanger human life or health [61]. These activities causes ionic disturbances. During geomagnetic storms total electron count (TEC) in the ionosphere can increase more than 100%. This changes the refraction indices for ionosphere and troposphere causing signal information delay and hence result in position error. Ionospehric scintillation in GPS signals arises from rapid spatial and temporal variations (less than about 15 seconds) in the ionosphere. This causes loss of lock on the satellites or 'cycle slippage' and can even affect dual-frequency gps receivers. Solar energetic particles (SEP)-a type of cosmic rays, X-rays, radio bursts, extreme ultraviolet (EUV) radiations can cause spacecraft damage, satellite orbit decay and disorientation, geolocation errors, space track and launch trajectory errors, etc.; affecting all the segments of a GNSS. Nuclear explosions in the upper atmosphere can cause similar effects to solar storms, potentially affecting the operation of GNSS for weeks due to propagation anomalies [63].

Multipath Error- Multipath interference occurs when a GPS receiver picks up reflected GPS signals from buildings or high towers along with normal signals from satellites to calculate its position. This can give grossly erroneous results. With advanced antenna-filtering techniques, receiver filtering and processing techniques, most new GPS receivers are very effective at mitigating multipath errors. If unchecked, multipath can still cause an error of ten to hundred meters.

Atomic Clock Drift Accurate timing has become a crucial parameter for companies which uses GPS to time-stamp their business activities. Since business transactions or power utilities requires precise timing, GPS clocks need to be highly synchronized and traceable to national and international standards.On occasions these clocks behave unpredictably which can produce high discrepancy in time and position data way before it is detected and corrected. Clock anomalies in GPS satellite atomic clock can be caused due to natural aging or bad navigation data upload. Various studies

has been done to detect such anomalies and techniques to mitigate the error.

Bad Signal- Unusual signal envelope transmission due to fault in signal modulation or generation causes unpredictable behavior in receivers. Most recent incident was in March 2009, when SVN-49 had multipath effect on GPS signals due to an on-board experimental L5 signal generator [37].

3.5.2 Deliberate Interference

Jamming- Jamming devices basically transmits radio frequency as noise which interfere with lawful communications as cell phone calls, messaging, Wi-fi and GPS system and hence leads to denial of service.GPS Jamming is the act of interfering with the ability of receivers to lock onto the GPS signal, eliminating the ability of the user to determine 3D position or calculate other information such as time, speed, bearing, track, trip distance, and distance to destination [34]. These frequencies can be filtered to some degree by adaptive antennas and noise-filtering in well-designed receivers. The higher the power of jamming signal, the more wider the circle of damage it will reach. These jammers can have continuous wave form signal or pulsed signal. Several anti-jamming techniques have been proposed to prevent narrow band interference, wide band interference, etc, but there is still no absolute solution to this problem. Jammers are illegal to market, sell or use in US soil.

Spoofing- Seemingly, the most easy method to mislead a GPS receiver would be to transmit false measurements leading to a wrong position, velocity, and time calculations. Spoofing involves fabricating and transmitting seemingly genuine GNSS signal. It aims at spoofing GPS signals to give a false sense of accurate physical location and results in mission path diversion. It is nowadays comparatively easier to launch such an attack due to the availability of off-the-shelf GPS signal generators. Satellite constellation preservation and signal transmission precision are of utmost importance in such an attack so that spoofing is not detected.A mechanism

called Receiver Autonomous Integrity Monitoring (RAIM) have been used in the design and development of anti-spoofing methods through fault detection and exclusion implementation.

Meaconing- Meaconing involves rebroadcasting the received GNSS signal with some delay in order to confuse enemy navigation. Meaconing has been noticed to happen unintentionally as well due to close by low impedance GNSS antennas. These attacks can be detected by analyzing the clock bias of the GPS receiver over time [49].

3.5.3 Human Factors

Receiver Bugs- Design requirements for a GNSS receiver does not require certification testing in all sectors. Certain specific circumstances can cause firmware bugs to be unearthed, such as handling of unhealthy satellites or tracking nonstandard codes. One such experiment was done to exploit sotware bugs in underlying receivers using a hardware that cost only $\$2,500$ and can cause a wide variety of GPS devices to malfunction within a 30 mile radius. Middle-of-the-earth attack was successfully conducted on Trimble NetRS which costs $\$19,000$ which goes into an endless reboot loop that persists even after the incorrect data is no longer supplied [14].

System Upgrade Bad Navigation Data- Navigation data for next 24 hours is uploaded to GPS satellites every 24 hours by a Master Control Station in advance. Even a chunk of bad data can cause catastrophes. Three reported incidents without much harm happened in GLONASS in June 2002, March 2000 and March 1993 [57].

Leap seconds and roll-overs- Leap seconds are not handles correctly in all GPS receivers and such a roll-over in August 1999 resulted in malfunction of few receivers.

Chapter 4

Enhanced UAVSim Design

4.1 Introduction

As discussed in our previous work, OMNET++ is the base simulator for our testbed. INET2.2 has been used for various mobility and radio propagation models, for wired and wireless communication. Primary aim of UAVSim design and implementation was to have a simulator that is more accurate and efficient in modeling real-world scenarios for UAVs and ultimately should be able to simulate the GPS related attacks such as GPS Jamming and GPS Spoofing. These attacks would not be possible without a GPS signal receiver module. To implement the navigation module as an enhancement to the existing UAVSim, the first requirement was to have a stable implementation of GNSS which would provide GPS signals for position determination. Using any of the GPS Simulators, discussed in Section 2.4, was not feasible due to cost and scale of implementation. The other requirement was to find a software simulator that would integrate easily with OMNET++. A study of different navigation simulators was done and compared to find the best solution for our research.

Galileo System Simulation Facility (GSSF) [75] is mainly to reproduce and analyze the functionality and performance of Galileo navigation system. Implemented

in C++, it provides simulation for longer time periods and large geographical area coverage. Available as a free licensed software from its website, [www.gssf.info], it also provides raw Galileo and GPS signal generation, express mode simulation and good functionality to analyze and visualize data.

The Satellite Navigation Radio Channel Signal Simulator (SNACS) [58] is a single satellite source GNSS signal generator. It is open-sourced and implemented in C++ with parallel processing. Its radio channel input and simulation results can be analyzed in MATLAB.

Open-SESSAME (Open-SOURCE, EXTENSIBLE SPACECRAFT SIMULATION AND MODELING ENVIRONMENT FRAMEWORK) [66] is another simulator that provides dynamics simulation for spacecrafts for developing hardware as well as testing flight algorithms. Based on C++, it not only provides attitude and orbit modeling, but can also be applied for orbit simulation, space environment assessment or control algorithm validation.

OS3 (Open Source Satellite Simulator) [69, 50] is an OMNET++ based simulation platform for evaluating satellite communication protocol. It was developed as a framework for simulating various kinds of satellite-based communication. OS3 provides a generic satellite constellation that seamlessly integrates real satellite tracks and weather data to simulate different conditions along with good visualization. OS3 was released under public license and is now part of the INET framework. Its code is available for modification and enhancement. Implementation of a highly accurate and stable satellite movement and modeling in OS3 provided a good base for development of our GPS system. Being platform independent, OS3 can be employed easily on any system. We discuss more about why this particular simulation platform was chosen.

4.2 CNI_OS3

The common and major limitation of the above mentioned simulation software is difficulty in integration with OMNET++. CNI_OS3 implemented simple satellite mobility (such as Global Positioning System) without any satellite communication implementation and satisfied all our other requirements. The foundation of this work was Galileo Satellite Communication Simulator (GSCS) [48] (also known as Multi-scale Satellite Simulation Environment (MSSE)). Although GSCS was based on INET framework of OMNET++ simulation engine, it was Galileo satellite navigation system specific. CNI_OS3 implements a generic satellite constellation. CNI_OS3 uses a TLE (Two-Line Element) format file for fetching initial positions of various satellites being used for simulation. Depending upon which Navigation System TLE file we use, that specific navigation system can be simulated. For example, we can use TLE file of 31 GNSS satellites to simulate GNSS while we can use a TLE file of 30 Galileo satellites to simulate Galileo Navigation System. It provides an accurate satellite movement simulation with live weather data, high resolution altitude data, different visualization options, etc. The comparison of these existing satellite frameworks have been done in [69]. Therefore, it can be concluded that other than communication, all other features were available in CNI_OS3..

4.2.1 Features

Below are some of the additional features of CNI_OS3 that supported its selection for use in UAVSim [70].

- Detailed modeling of communication aspects
- Dynamic integration of already implemented protocol stacks
- Modular architecture and easy extensibility

- Usable for any arbitrary constellation
- GUI for easy and comfortable handling
- Configuration of scenario-specific parameters
- Two different visualization methods
  - Worldview
  - Local Evaluation/Azimuth representation

4.2.2 Limitations

Below are the limitations of CNI_OS3, some of which require to be addressed before moving on to the implementation:

- Satellites were not capable to send signals and establish a communication link.
- GPS or any other navigation technique was not implemented. A crude localization was available which used object oriented programming where an observer could directly access the satellite coordinates using its object and then save them without calculating its own position using those coordinates.
- Only two types of satellite mobility models were implemented.

4.3 UAVSim Modules

To overcome the limitations of CNI_OS3, we added new modules to our existing work UAVSim, in addition to the six core modules to the UAVSim as depicted in figure 4-1. The new modules are shown in pink in the figure and defined in brief below. More about the core modules can be found in [37].

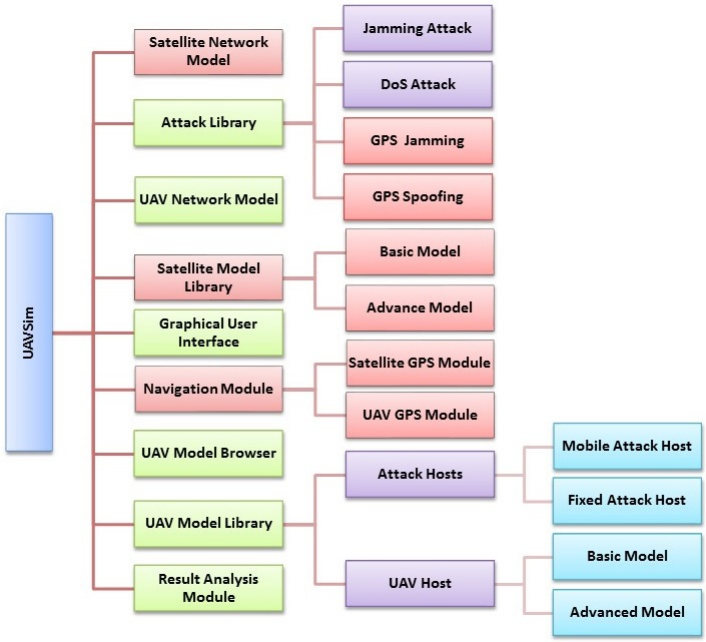

Figure 4-1: Architectural Design of UAVSim

4.3.1 Satellite Model Library

Satellite model library has the standard satellite model which inherits its basic features from the satellite model defined in CNI_OS3. The GPS functionality has been added to the satellite model through development of a broadcasting based application which sends positions information to the receivers through radio signals of $L1$ frequency range as per the standard GPS implementation. The GPS/SATNAV implementation has been described in detail in section 3.

4.3.2 Satellite Network Module

Similar to UAV network module, this module defines the network stack of satellites. It defines the communication protocol, transmission power, access points, etc. This modules uses several basic packages from INET for satellite communication while the satellite mobility packages are used from CNI_OS3.

4.3.3 Navigation Module

The navigation module has the receiver end GPS application which enables the hosts to receive satellite navigation signals carrying its position information. The information from four or more satellites are required to calculate the position of the UAV using Multi-lateration in the $3D$ space. In our implementation, we have approximated the implementation to a $2D$ localization, therefore, a minimum of three satellites are required.

4.3.4 Attack Library

The attack library contains various attacks that were implemented in the previous work. These include single target DDoS and multiple target Jamming attacks. With the implementation of GNSS and navigation module, GPS Spoofing and GPS Jamming attacks where included in this library.

4.4 Implementation

This section elaborates the implementation details of our enhanced UAVSim. The communication between UAVs and satellites or any GPS receiver is unidirectional. Satellites broadcast signals without waiting for any acknowledgment. That is why a connectionless broadcasting protocol was used to implement GPS. The packets contain satellite index, X coordinates and Y coordinates of the satellite sending the packet, in map-pixels, and distance of the satellite from each host. For the first time when a UAV receives the packet, it locks on that particular satellite. After obtaining lock on three different satellites (in 2D implementation), UAV calculates its position. This is called 'Cold Start'. It start accepting packets from these three locked satellites and calculate its position using the $2D$ trilateration equations as defined in section 3.2.1. Point to note here is that the packets should be from each

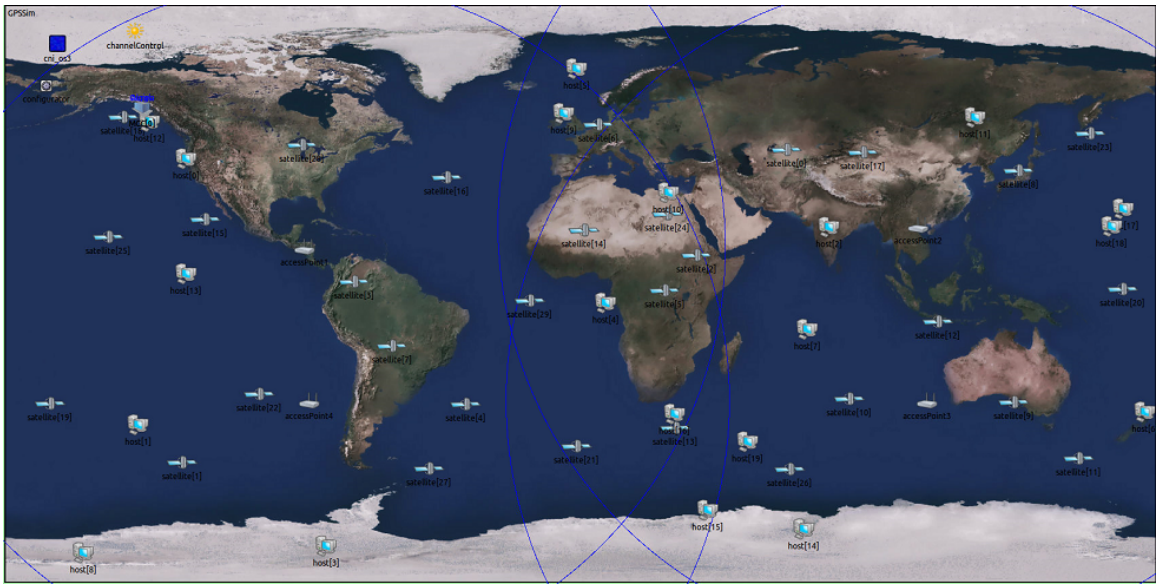

Figure 4-2: Simulation world Map

locked satellite. If UAV doesn't receive packet from any of the locked satellites for $10s$, the lock on that satellite is released and a new satellite is searched for, to lock on to. Such a situation is termed as Locate-the-sky.

The implementation of GPS is very basic and an approximation of the real GPS. The authors have implemented the localization in 2D, considering the map pixels as the position of the satellites and hosts. The scaling of the earth to the map has been done as follows. Considering, the radius of the earth is 6371 km, the circumference of the earth will be

$$2\pi r = 2 * 3.14 * 6371$$

If we imagine to open the globe vertically, the circumference will give the length of the map while the width of the map will be half the circumference. The map dimensions are defined as

$$1080 * 2160$$

The figure 4-2 shows the world map in the simulation environment. So horizontally,

each pixel on the map corresponds to

$$\frac{2 * 3.14 * 6371 * 10^3}{2160} = 18.5km$$

or vertically,

$$\frac{3.14 * 6371 * 10^3}{1080} = 18.5km$$

Therefore,

$$1m = \frac{1}{(18.5 * 10^3)} pixel$$

Different UAV models available today have different speeds depending upon its design and usage. The Arcangel-1 has a speed of 150 km/h while airforce mission UAVs like Patroller range of UAVs can fly at a speed of 241 km/h. Taking an approximate speed of 250 km/h of a UAV or 69.5 m/s, the speed of the satellite on the simulation map will be

$$69.5/(18.5 * 10^3) = 0.0037 pixelspersecond.$$

The user defined parameters can be defined in INI file such as the number of satellites and UAV hosts. The simulation world map shown in figure 4-2 shows 30 satellites in orbit. The satellites revolves around the earth in an orbit which is inclined at an angle of 55 degrees. The motion of the satellites in their orbit around the planet appears as a sine wave when tracked from ground as shown in figure 4-3.

For illustration, 20 host are shown on the map. The number of hosts can be varied for each simulation. Along with satellites and hosts, there is a CNI_OS3, channel control and mission control centers depicted on the map. The CNI_OS3 module defines the webservice related data, weather control data and calculation module which calculates the distance, attenuation and other channel characteristics. There are access points which act as a central transmitter and receiver of wireless

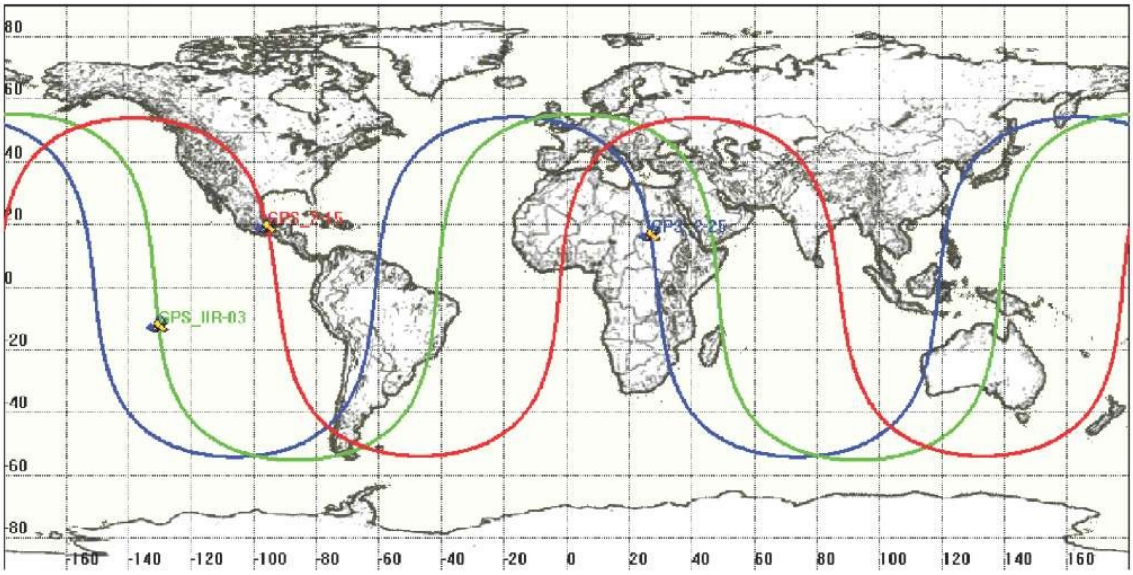

Figure 4-3: Satellite Ground Track [68]

radio signals. The blue circles shows the range of each access point. If a host is outside the range it won't receive any packets. Such that the host is always in the line-of-sight of one or more satellites, range of access points needs to cover the whole area of the map. A minimum of four access points is sufficient to cover the whole area with the given range.

Figure 4-4 explains the implementation of the GPS system. The space segment constitute GPS satellites while UAV hosts is the the user segment as it receives GPS signal. The base satellite model with NORAD (North American Aerospace Defense Command) module, mobility model, and notification board has been defined in CNI_OS3. Standard satellite model extends base satellite while adding communication protocol stack from INET module. This protocol allows communication between network layers of the satellite so that the packets can be broadcast.

The GPS Application has been designed over a connectionless broadcasting protocol provided by INET. This application is required in the standard satellite model to create and broadcast packets. It can be enabled/disabled in .INI file. On UAV host,

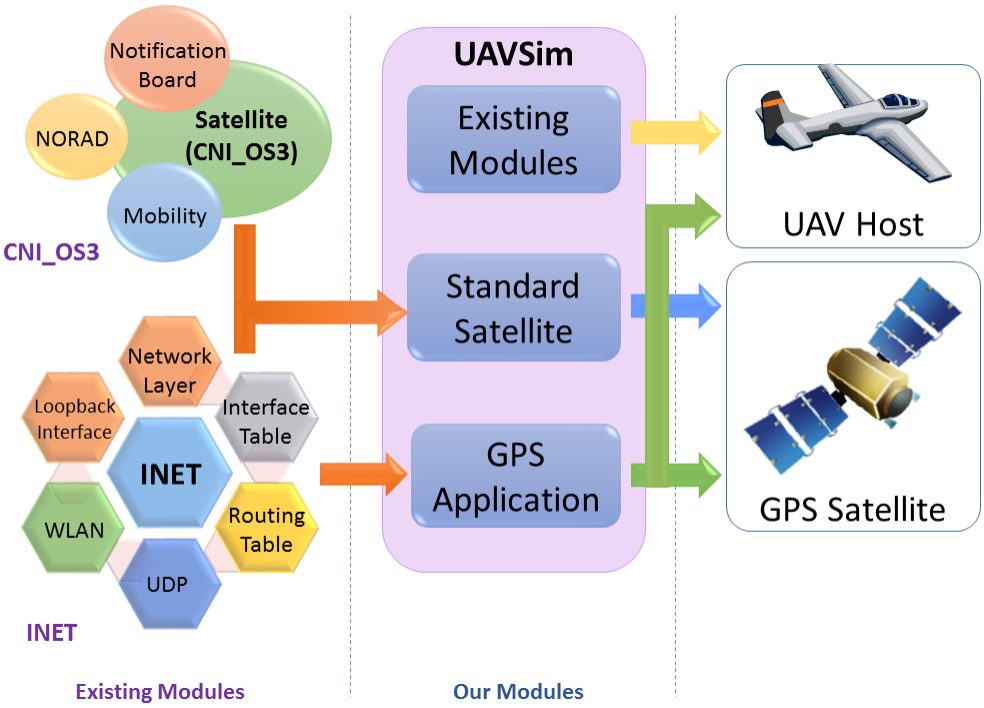

Figure 4-4: GNSS/GPS Implemetation

this application acts as a GPS receiver in the navigation module. It receives packets from satellites and process received data for localization. Other than enable/disable parameter, .INI file also defines local port, destination port, message length, packet send interval, mode of traffic to be sent, and the destination addresses. Mode of traffic to be sent can be set to 'ONCE', 'PER_BURST' or 'PER_SEND'. If there are more than one destination address and mode of traffic parameter-'chooseDestAddrMode' is set to 'ONCE', then one of the address is randomly selected for the whole simulation run. Similarly, destination address is chosen for each burst or for each packet depending on the value of chooseDestAddrMode parameter as 'PER_BURST' or PER_SEND respectively.

Algorithm 1 defines processes of the GPS Application. After initializing global variables, it fetches destination addresses. The function 'processSend()' calls generateBurst() function which create packets and send them to destination hosts as

defined by the mode of traffic parameter 'chooseDestAddrMode'. For our simulation, chooseDestAddrMode is set to PER_BURST. processPacket() is defined for GPS receivers i.e. UAVHosts. It is non-functional for satellites and attackHost. When GPS packets are received by UAVs, they are processed and data sent by the satellites are extracted to evaluate its position. Finally, statistics are collected on outgoing and incoming packets.

Algorithm 1: GPS App

```
1: procedure GPSApp
2:     initialize()
3:     processStart() ← destination addresses
4:     processSend() ← generateBurst of packets
5:     generateBurst() ← createPacket()
6:     processPacket()
7:     finish()
```

Algorithm 2 shows the steps and parameters required to create a packet for GPS signal. The packet has sourceId and msgId as two parameters. msgId gets incremented by 1 for each packet send. Each packet has satellite data, its location and distance information.

Algorithm 2: Create Packet

```
   procedure createPacket
2:     sourceId ← getId
       msgId ← numSent
4:     satId ← getIndex
       Xcoord ← PositionX
6:     Ycoord ← PositionY
       setDistance ← distance of host from satellite
```

Algorithm 3 shows how the packet is processed. The packet is checked for sourceId amd msgId. When simulation starts, UAV would have no lock on the satellites. It will get the position data from the packets received from three unique satellites.

Satellite index helps it to identify the satellites. A counter variable 'countPacket' is incremented to one after packet is received from distinct satellite index otherwise it is discarded. Once count of packets becomes equal to three, position of the UAV is calculated using trilateration and lock is set on those satellites while counter is set to zero. Next time when the packets are received, algorithm checks for those satellite indexes on which lock has been provided. It keeps on discarding the packets from other satellites and wait until it receives packets from the locked satellites. In real world, such a scenario happens when there is no satellite signal and GPS device wait to receive signals.

Algorithm 3: Process Packet

```
1: procedure processPacket
2:     if packet has sourceId and msgId then
3:         satelliteId ← satId
4:         if (countPacket ≤ 3) then
5:             if (lock == 0) then
6:                 countSat[countPacket] ← satelliteId
7:                 positionData[countPacket] ← position details of three unique satellites
8:                 countPacket ← countPacket + 1
9:         else
10:            positionData[countPacket] ← position details of the three locked satellites
11:            countPacket ← countPacket + 1
12:        if countPacket==3 then
13:            calculatePosition()
14:            countPacket ← 0
15:            lock ← 1
16:    close;
```

The figure 4-5 describes the classes defined in the UAVSim. INET and CNI_OS3 are predefined packages. UAVHost has been defined as previous work. GPSApp inherits Linear/Circle mobility from INET and SATSGP4Mobility from CNI_OS3 modules. Satellite Model inherits basic satellite feature from CNI_OS3 and communication protocols from INET through which it can communicate with other satellites. It inherits GPS Application to act as a GPS Satellites. GPSSimulation module de-

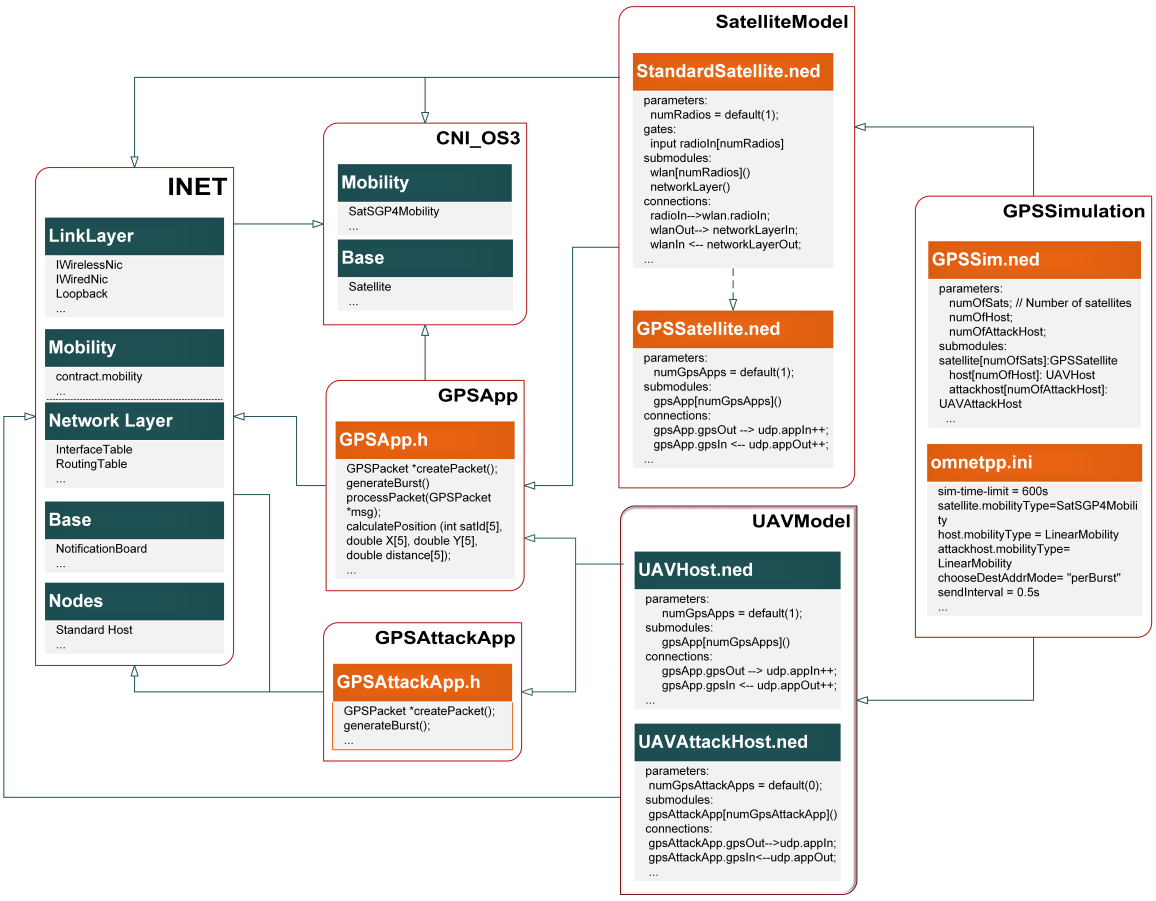

Figure 4-5: Class Diagram

fines the GNSS/GPS in the NED file with UAVs, attack hosts and mission control centers. The new modules are depicted in orange. UAV model has UAVHost and UAV attack host. UAV attack host has features equal to the UAV host except that it sends counterfeit packets similar to the packets sent by GPS Satellites. Also, packet processing feature of the attack host has been disabled as they won't be processing any packets.

4.4.1 GPS Attacks Implementation

GPS attacks where implemented by attack host which has equivalent or more features as the UAV host except that it can send counterfeit GPS signals. The

GPSAttack App shown in figure 4-5 is enabled for attack hosts to send the spoofed signals. These UAVs doesn't receive any packets broadcast by the satellites.

4.4.1.1 Jamming

We implemented this attack using a large number of attack hosts which roam around in the simulation area and keep transmitting noise signal with high data rate and power in order to jam all frequencies. Jammer implementation in our testbed was abstracted through use of several hosts to block each frequency intended to jam. This abstraction is being used due to the limitation of underlying simulation engine. Clearly, in order to jam all communication frequencies, the number of attack hosts required will be quite large. Therefore, we jammed only a few GHz of bandwidth based on the detected frequencies in that area.

4.4.1.2 GPS Spoofing

We implemented it using a spoofed GPS Signal Generator, which is in fact another UAV at almost double the altitude of the UAV being attacked. Due to the public nature of GPS implementation details, building such a generator would be quite easy. We assume higher height in order to mislead directional antennas installed on UAV for GPS signal reception. Also, the attack host maintains same angle and distance with the host at all times so that the host does not detect any suspicious activity. The spoofed signal can contain discrepancy in x co-ordinates, y co-ordinates or distance as represented in the algorithm below:

4.5 Constraints and Assumptions

Various constraints and assumptions made during the development of GNSS/GPS in UAVSim include the following:

Algorithm 4: Create Attack Packet

```
   procedure createPacket
2:     sourceId ← getId
       msgId ← numSent
4:     satId ← getIndex
       Xcoord ← PositionX+discrepancy
6:     Ycoord ← PositionY
       setDistance ← distance of host from satellite
```

- Although UAVSim implements both the space segment and user segment, the control segment is not implemented.

- Only the primary functionality of GPS, position calculation of the UAV is implemented at the receiver end.

- Use of ephemeris and almanac data for calculation of the position is implemented approximately. The satellites sends its position information relative to the UAV in the packet as a parameter rather than calculating the distance using speed of light and time of transmission.

- The project being nonfunded, available medium-end systems were used rather than high performance parallel computing system.

- Most of the system information is not available in the public domain and thus gathering various network and communication related information posed a major challenge.

- Instead of calculating the distance between satellite and the host through speed of light and time difference in transmission and reception, this distance is being sent in the packet itself. Reason being the limitation of OMNeT++ of Tx/Rx event timing being exactly the same (accurate up to a nanosecond) to make it appear real-time.

- We have approximated the implementation to a 2-D localization (trilateration) instead of 3-D (multilateration). This is due to the 2-D nature of OMNeT++ simulations.

- Capability of attackers have been assumed to be equal or more than the UAVs in simulations. This enables impact evaluation of more powerful adversaries as well as when our own systems are compromised.

- The communication environment of the simulation testbed takes into consideration disturbances caused by random noise, upper layers of atmosphere and communication signals present in the lower layers.

Chapter 5

Simulation Analysis and Results

5.1 Introduction

In this section we demonstrate the simulation results based on the implementation of navigation module in UAV and efficiency of an attack host to successfully launch a GPS Spoofing attack on UAV. The first part of the results are analyzed to demonstrate the correctness of the GPS implementation while the later part discusses how changes in the parameters of attack host signals can alter the path of UAV. The various scenarios and variation in parameters have been analyzed with respect to the localization and mobility type. The below tables shows the default values of the satellites, host and attack host during normal simulation. The default Simulation Time is taken as 600s unless stated otherwise. Attack simulations have been considered with one UAV and one attack host.

Table 5.1: Default Satellite Parameters

Parameter	Value
Mobility Type	SatSGP4Mobility
Transmitter Power	500W
Packet Interval	0.5s
Burst Duration	10s
Sleep Duration	0s
Position Update Interval	1s

Table 5.2: Default Host Parameters

Parameter	Value
Mobility Type	LinearMobility
Transmitter Power	10W
Speed	0.0037s
Burst Duration	10s
Sleep Duration	0s
Position Update Interval	1s

Table 5.3: Default Attack Host Parameters

Parameter	Value
Mobility Type	LinearMobility
Transmitter Power	10W
Speed	0.0037s
Burst Duration	10s
Sleep Duration	0s
Position Update Interval	1s

The background image of the map of earth used is the default one used in CNI_OS3. Other background images can easily be used by changing it in the configuration file. Using the testbed, we analyzed how the UAV performed with the navigation module implemented. The below analysis shows the accuracy of UAV localization implementation with different parameters and the average error. The average error is calculated by calculating the distance between actual UAV position and the calculated position and then taking out averaging over whole simulation result.

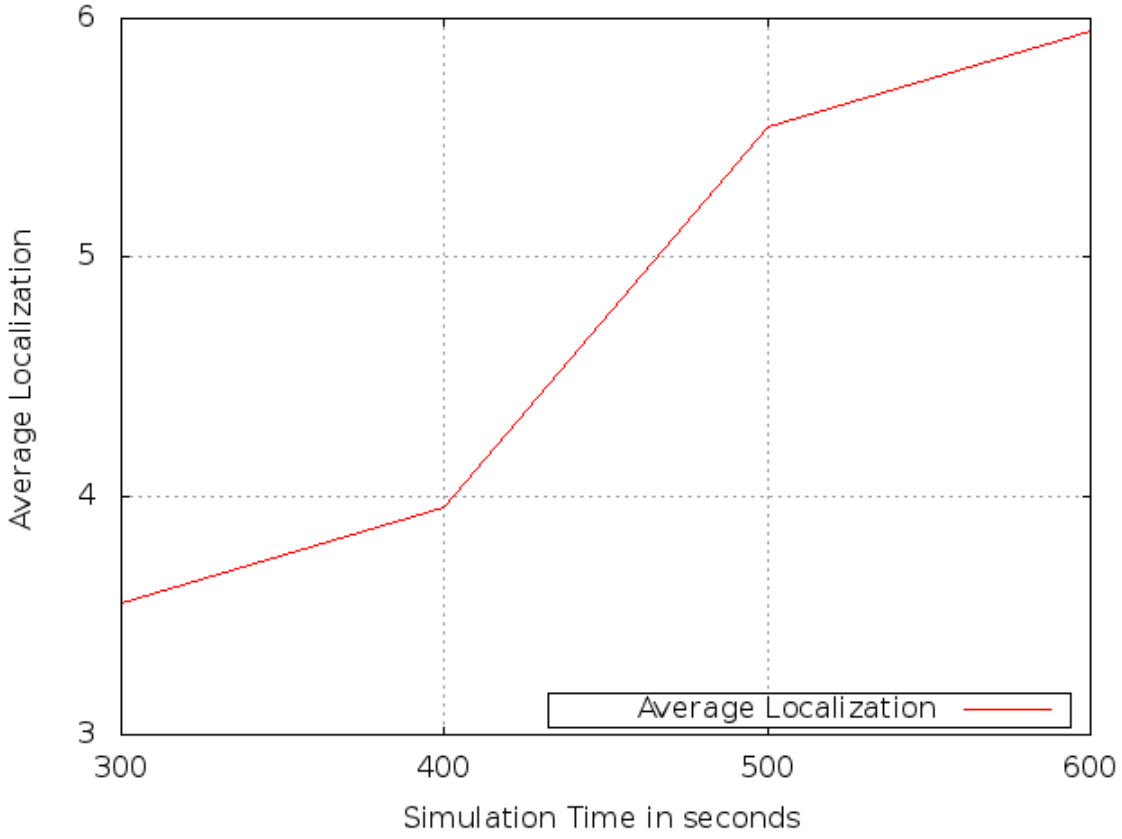

Figure 5-1: Variation in localization with Simulation Time

5.2 Results based on GPS Implementation

5.2.1 Average Localization versus Simulation Time and Number of Host

Although the localization count is not the same in every run, with increase in simulation time, average localization increases as shown in figure 5-1. It should be noted that average localization decreases with increase in number of hosts as shown clearly in figure 5-2.

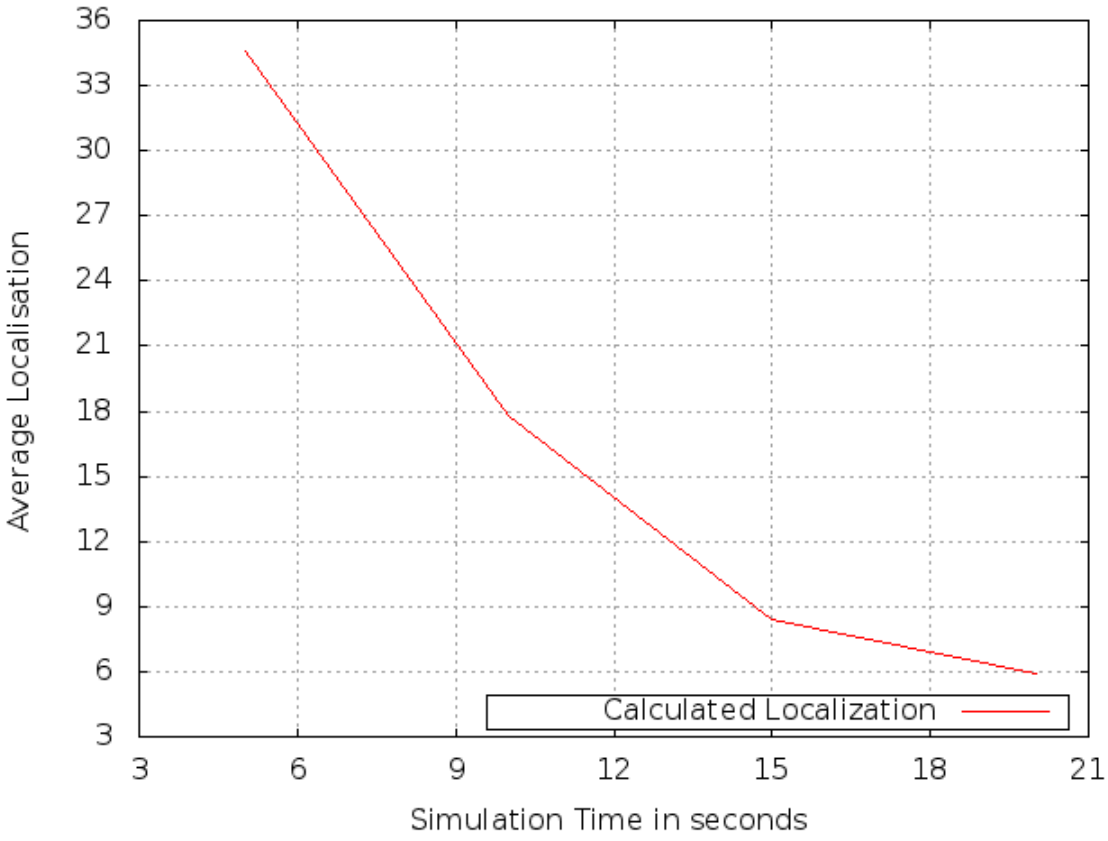

Figure 5-2: Variation in localiztion with number of host

5.2.2 Effect of Seed Value variation

Seed value is a number used by the PRNG (pseudo random number generator) algorithm of OMNeT++. This random number is used to randomize initial values of various parameters, such as position. Changing the seed value changes the position of the satellite on the map as shown in the figures 5-3 and 5-4. When the seed value is 2352610, the host is on the lower left corner of the map and when it is changed to 7399210, host moves to the lower right corner. It is important to evaluate the effect of seed value because the initial position impacts the localization of the host as it affects the possibility of its presence in the communication range of an access point. In order to cover the whole map for correct localization, we have deployed four access

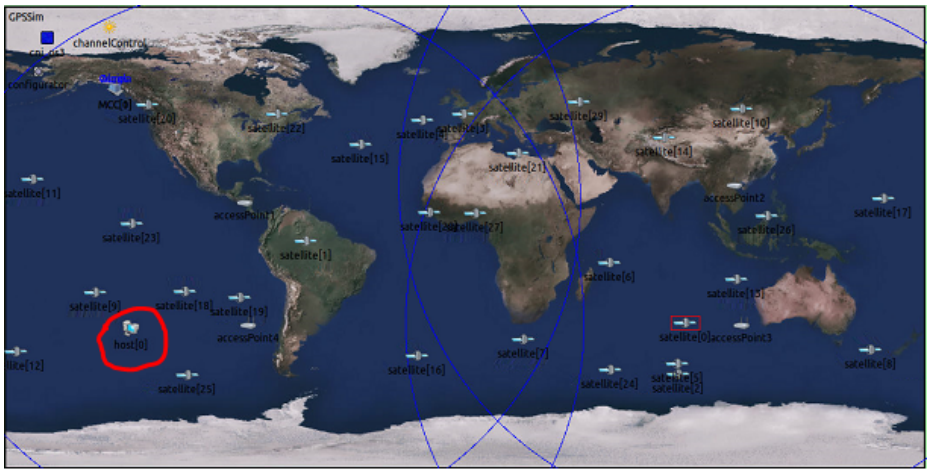

Figure 5-3: UAV host on the lower left corner of the map

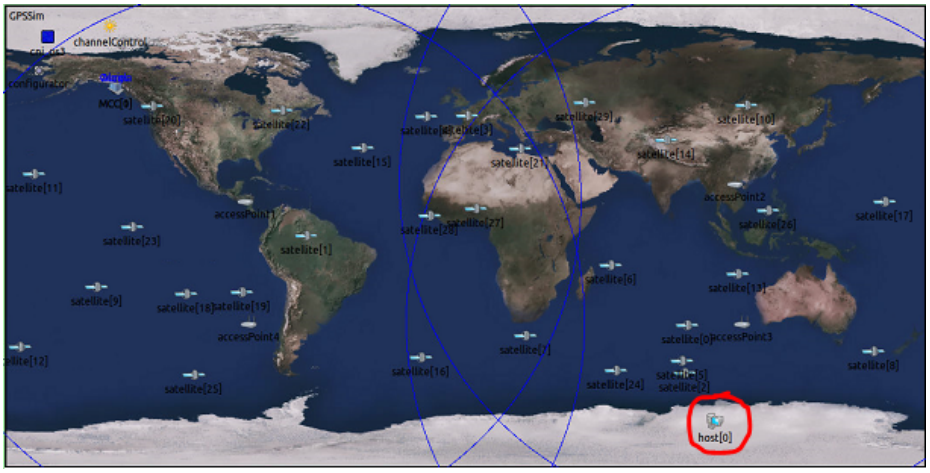

Figure 5-4: UAV host on the lower right corner of the map

points. These access points resemble the Differential GPS (DGPS) stations on earth which help in easier localization and communication.

5.2.3 Average Localization versus Satellite Lock

Obtaining a lock is important to get a very accurate position. Satellites broadcasts signals all the time. Also, there can be more than 3 satellites (2D implementation) in line-of-sight but at different distances with the host. The calculated distance is prone to error but the effort should be to have a minimal error. If every computation of the

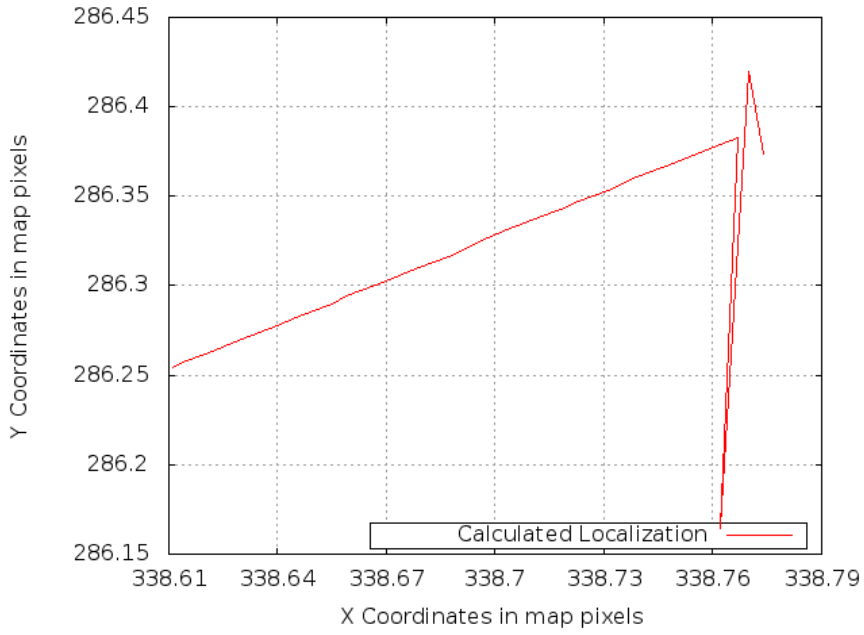

Figure 5-5: Localization Curve with lock implemented on satellites during localization

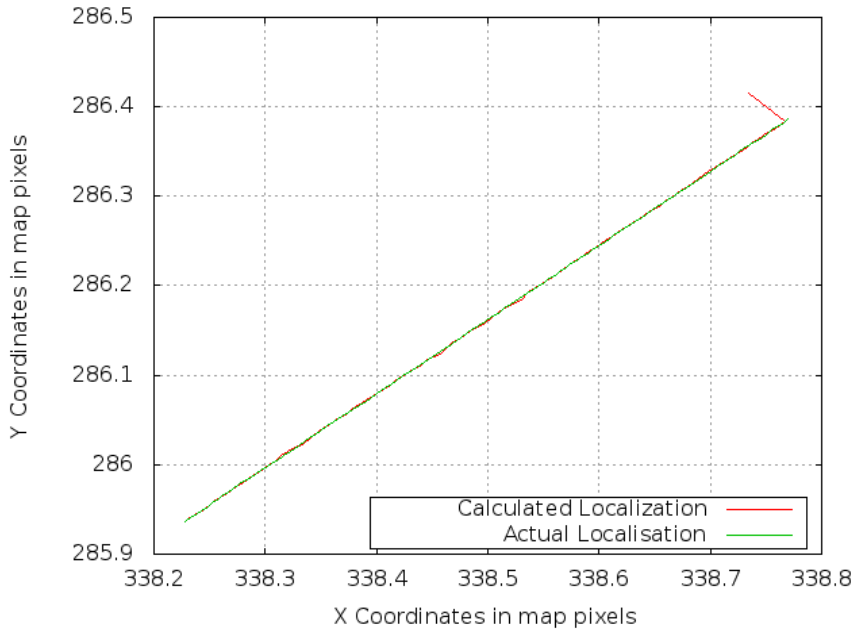

Figure 5-6: Localization Curve without lock implemented on satellites during localization

localization uses data from different sets of satellites, it will lead to large variation in error between the calculated position and the original position.Figure 5-6 shows the variation in localization when no lock on the satellites is implemented. The variation between the calculated localization curve and the actual curve was so large that they could not be plotted together.

When the lock on the satellites is implemented the variation in error reduces to a minimum of 0.001. The calculated position almost overlaps the original position. Figure 5-5 shows the localization curve to be in sync with the original path through the lock on the satellites.

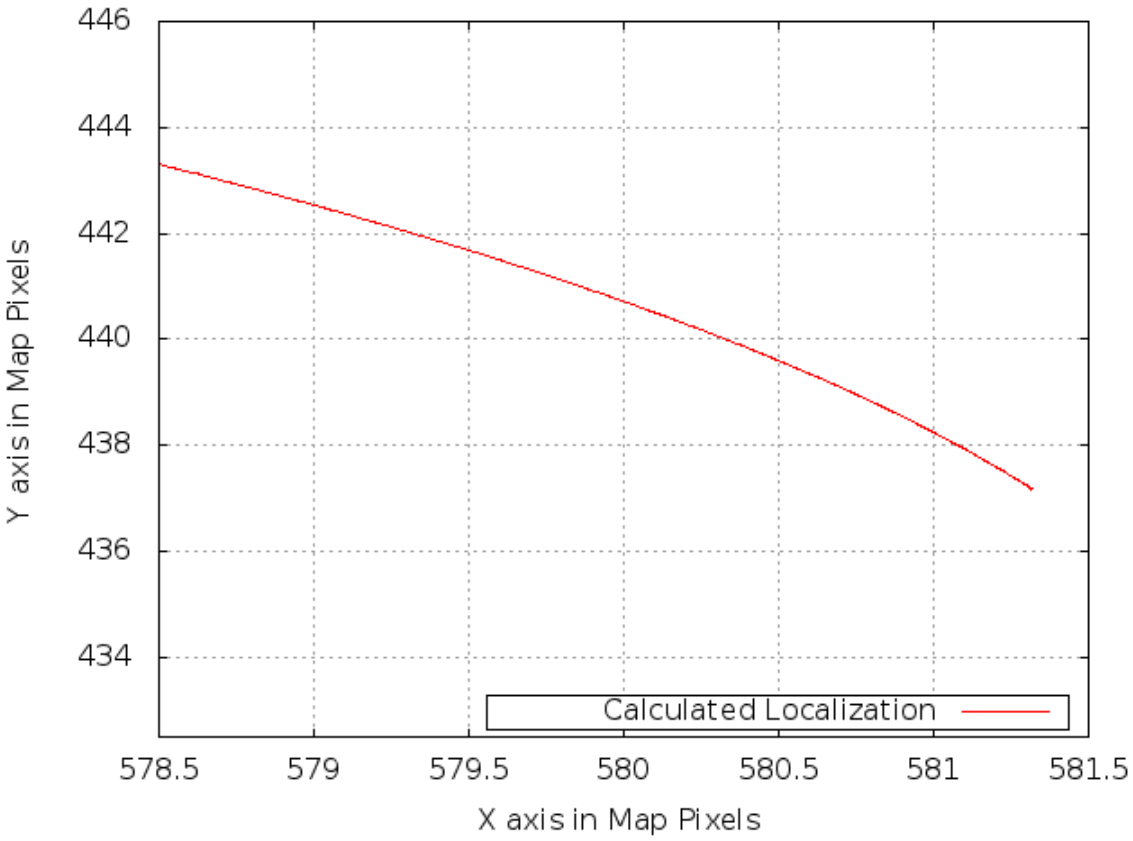

Figure 5-7: Graph of Circle Mobility

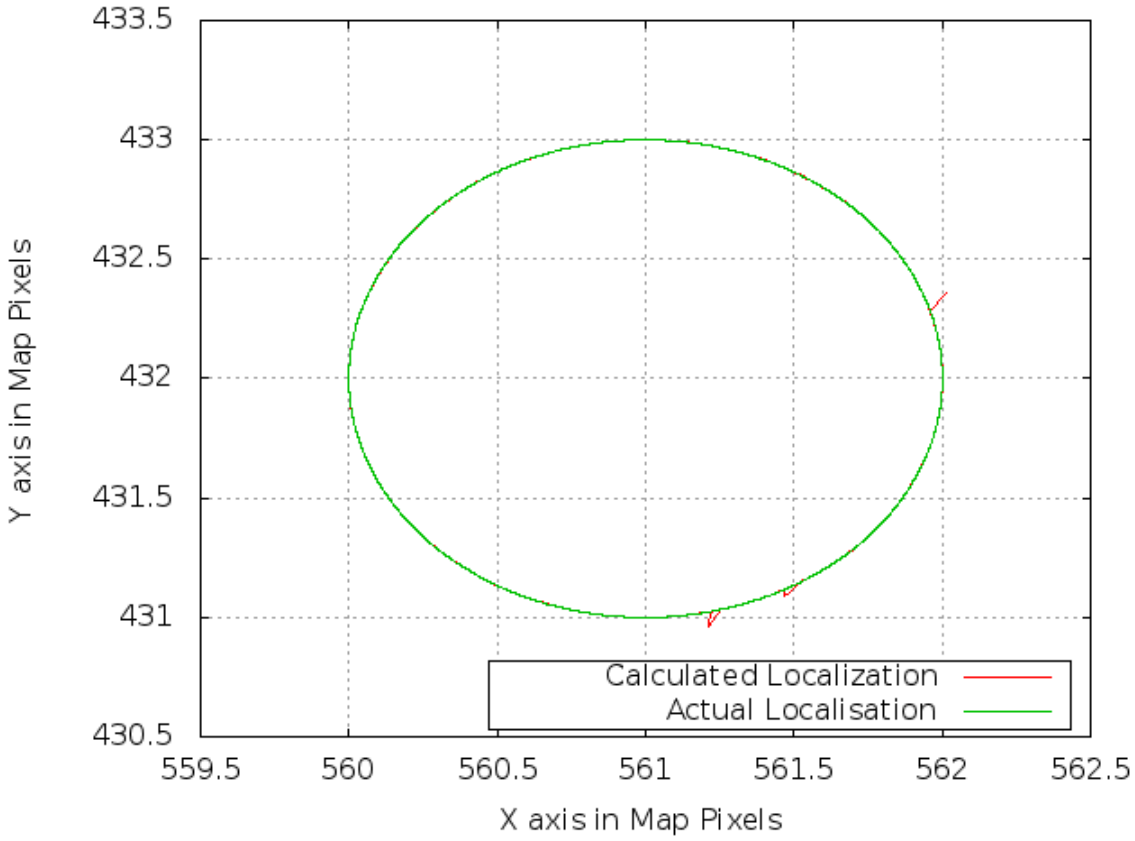

Figure 5-8: Graph of Circular Mobility Complete Trajectory

5.2.4 Implementation of Circular Mobility Model

The simulation runs as expected in different mobility models. For Linear Mobility, the UAV travels a linear path as shown in the above figures. Most of our simulation analysis is performed with Linear Mobility. The figure 5-7 shows the circular path as taken by the UAV host when the mobility model is changed to Circle Mobility. One need to define the center of the path and the radius. To get the complete circular trajectory as in figure 5-8, the simulation was run for 1000 simulation seconds. For this particular case, the center was taken as (x,y)=(561m, 432m) in map pixels and radius as 1meters which corresponds to 18.5 km in the real world . These mobility model are already defined in INET module. Other mobility models can also be selected and

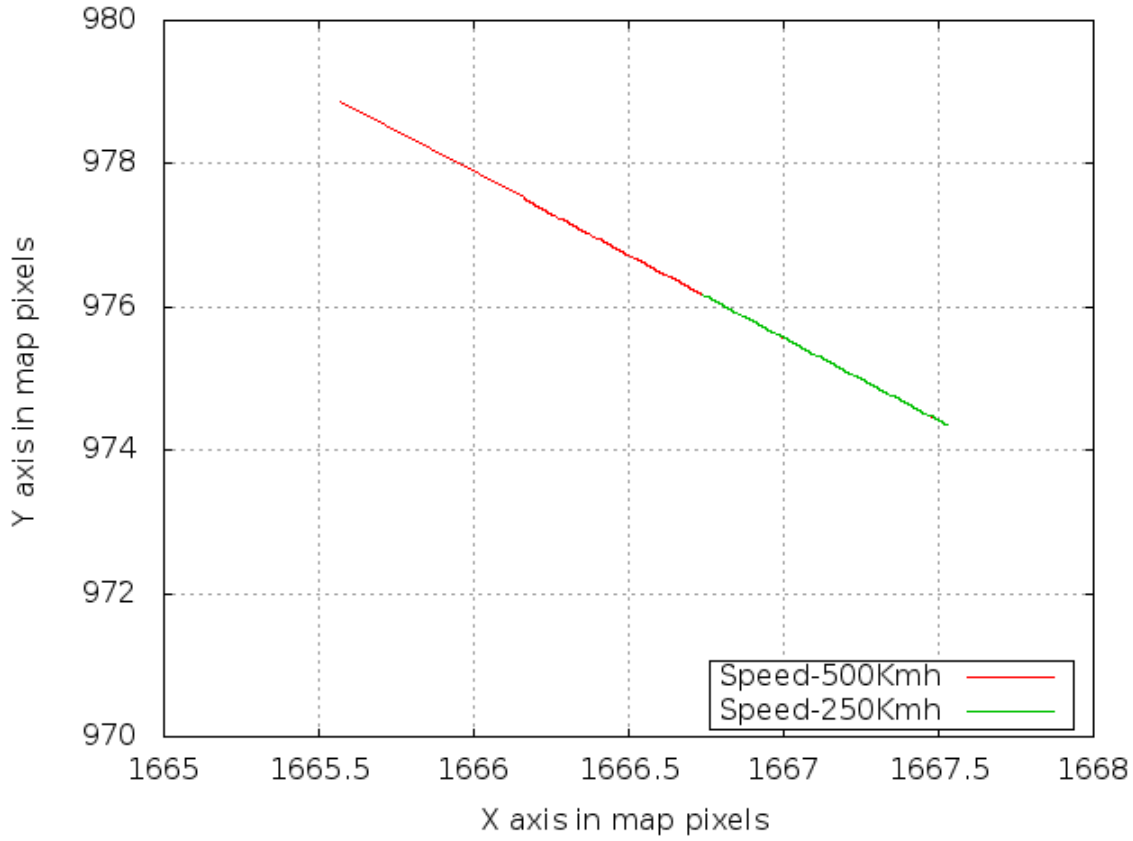

Figure 5-9: Distance Traveled With Speed

simulated.

5.2.5 Average Localization versus Speed

As the speed of the UAV host increases, as expected, distance covered by the UAV increases. Figure 5-9 shows the distance traveled by the UAV with different speeds. The green curve shows the distance traveled when the speed is 250 km/h and the red curve shows the distance traveled when the speed is 500 km/h keeping the simulation time as 400 seconds. With increase in speed, average localization error increases as shown in figure 5-10. The default value of speed is 0 mps i.e., the host is stationary. The speed is defined in the configuration file as a PRNG function of

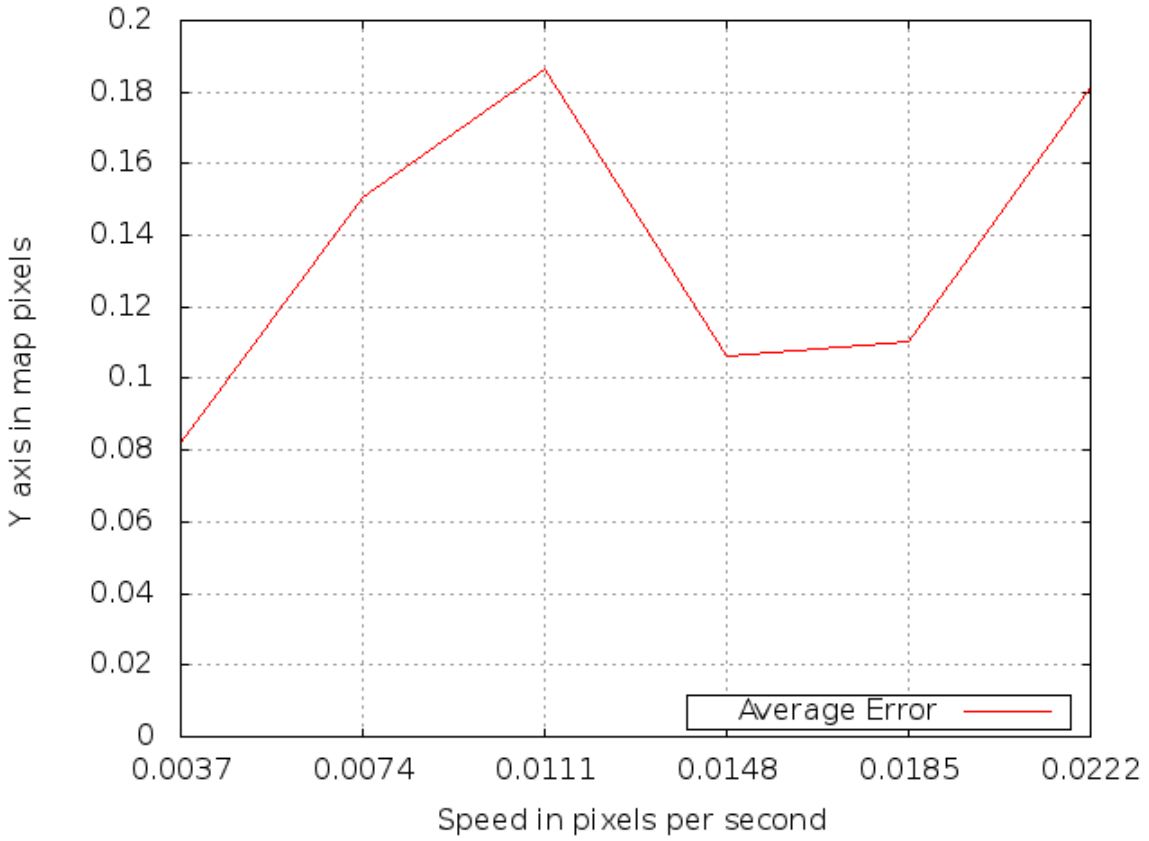

Figure 5-10: Speed Vs Average Error

normal distribution truncated to non-negative values. The standard deviation was set as 0.005.

5.2.6 Variation in Angle of Linear Motion

The default value of angle of Linear Motion is a random value with a uniform distribution between 0 degrees to 360 degrees. The graph 5-11 shows the calculated path of the satellite at different angles. As we can see from the graph, that the host follows correct angle path from its calculated localization data. The graph 5-12 shows the average error at different angles. Average error increases as we move from its linear axis and is again minimum at 90 degrees. The graph 5-13 shows the

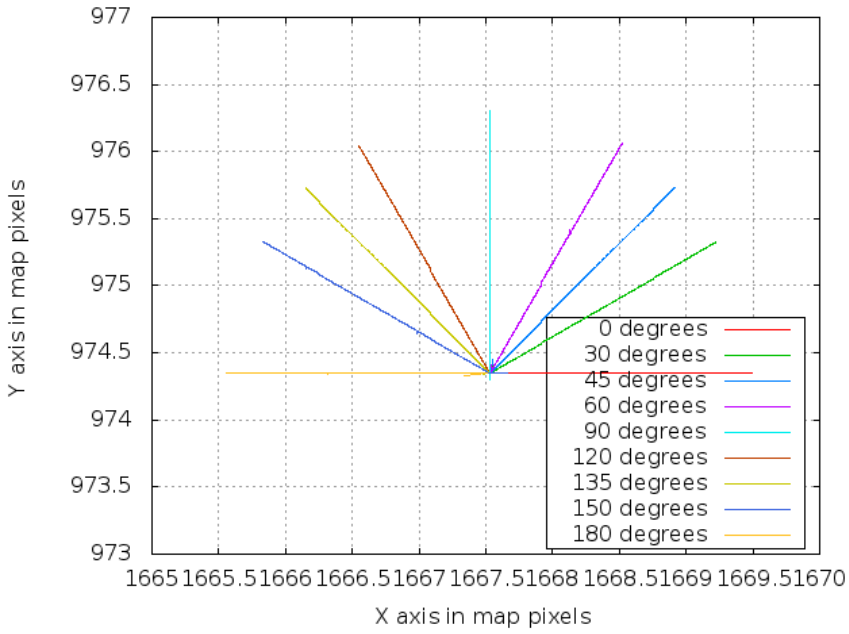

Figure 5-11: Path of Host at different Angles

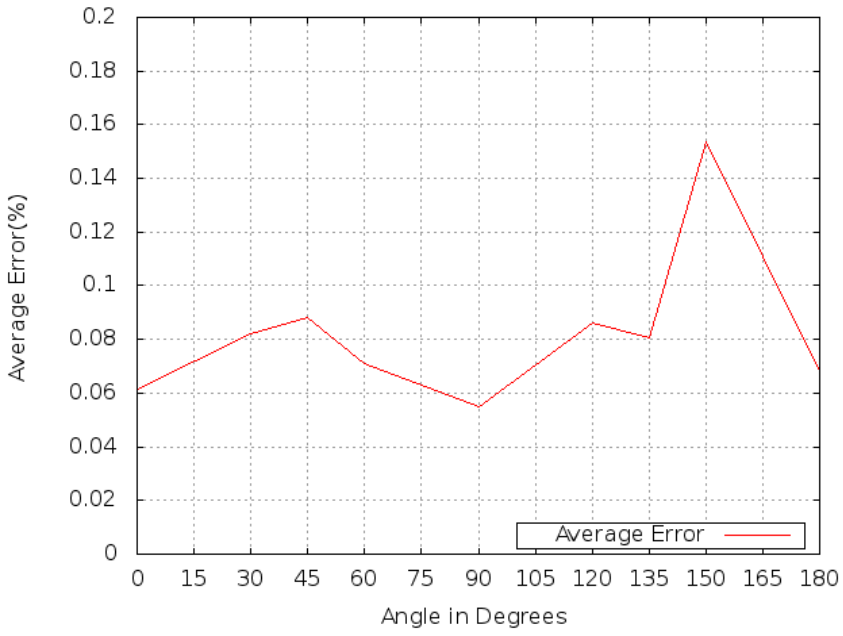

Figure 5-12: Angle vs Average Error

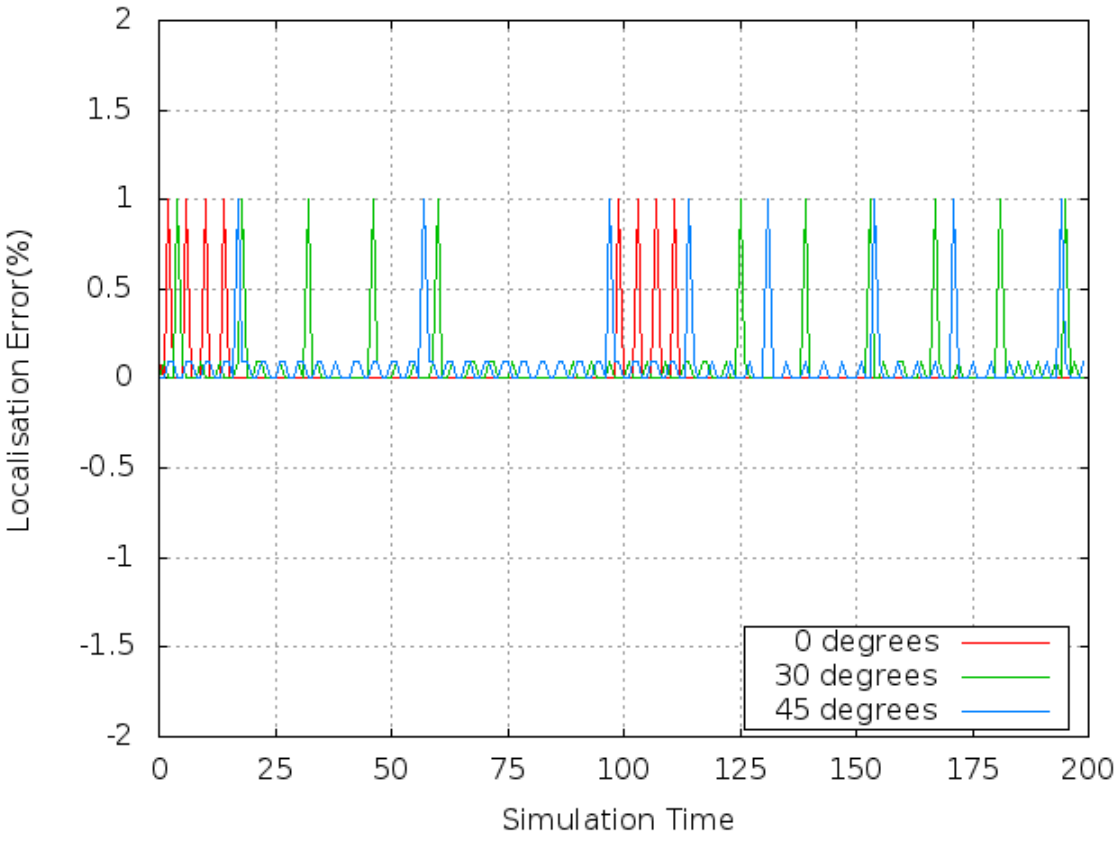

Figure 5-13: Localization Error versus Simulation Time

variation in localization error with simulation time for three different angles. The localization error is approx 1% only. As we can see from the graph, the number of times error in localization occurs increases with increase in the angle.

5.2.7 Sleep Duration versus Localization

GPS based navigation can offer relatively consistent accuracy if sufficient GPS signals can be tracked during the entire UAV mission [67]. Due to its low power range, intentional or unintentional interference can cause UAVs to lose the signals. The GPS signal outage can cause a significant deviation in the navigation solutions. To represent such a scenario where UAVs lose the signals because of interference from

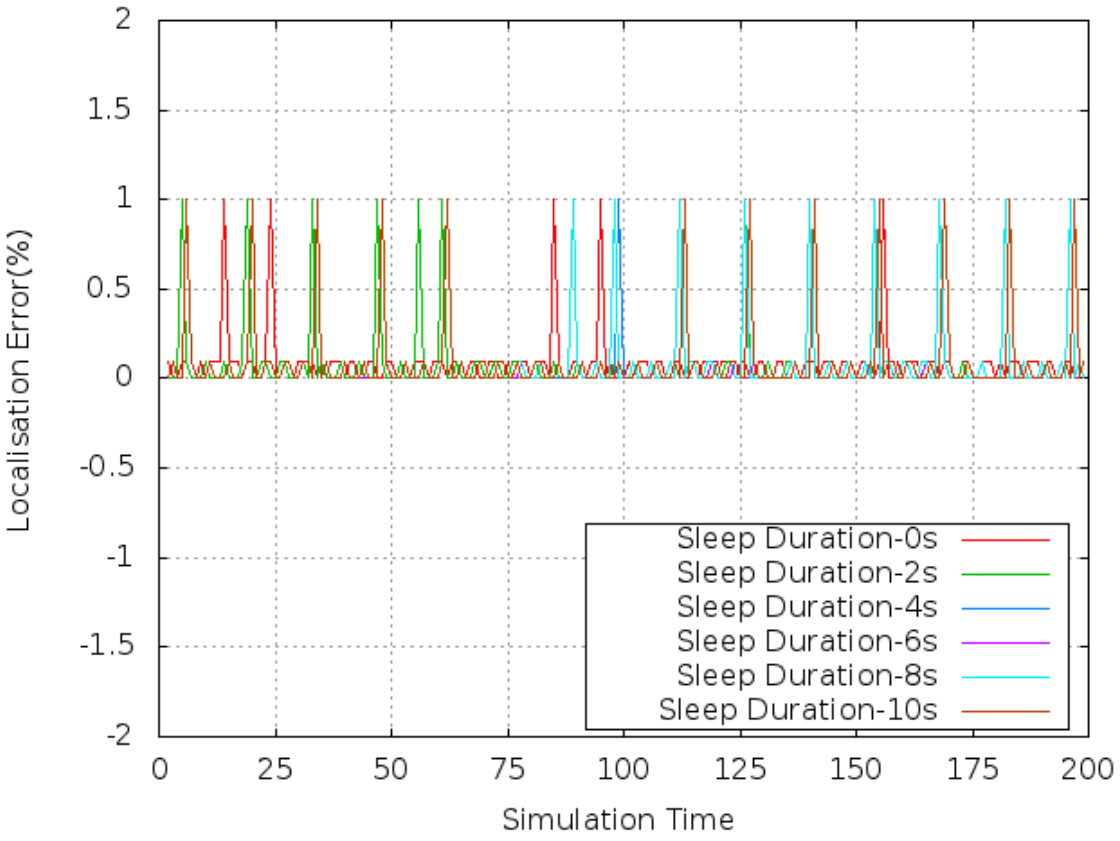

Figure 5-14: Sleep Duration Vs Localization Error

television signals, mobile signals, ultra wideband communications or when traveling around high buildings and trees, our testbed has a parameter called 'sleep duration', which corresponds to such an GPS signal outage. The parameter can be assigned to both the satellites and the host. In case of satellites, during the specified sleep duration, they would be inactive and would not broadcast any packets while the hosts would reject packets even if the satellites are sending them. The default sleep duration has been kept at 0s for the satellites as well as the host assuming that there is no outage and the UAV receives signals during its entire simulation time.

Figure 5-14 shows the effect of change of sleep duration on the localization error. The simulation has been mapped from 0s - 10s of sleep duration with an interval of

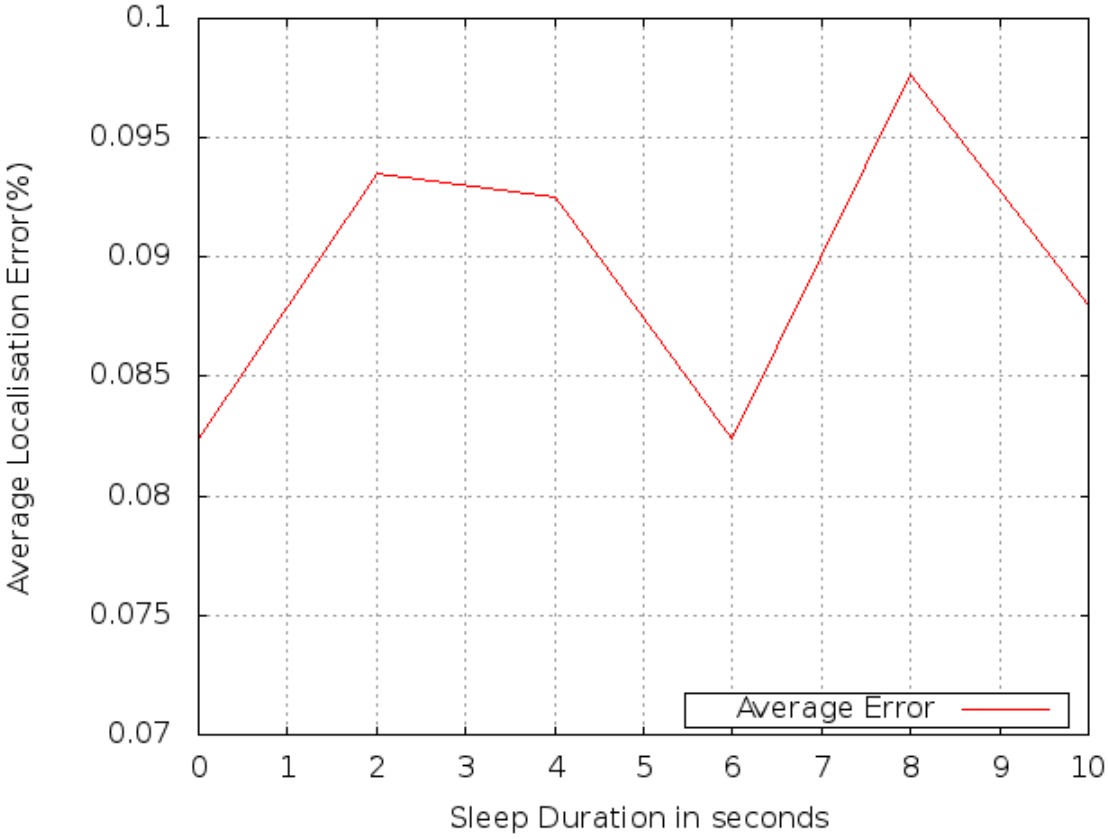

Figure 5-15: Sleep Duration Vs Average Localization Error

2s for 200 localization values. It clearly shows that the maximum error reaches about 1%. Figure 5-15 shows the overall localization error with respect to sleep duration which is less than 0.1%. Average localization error is almost negligible which shows the accurateness of the localization algorithm.

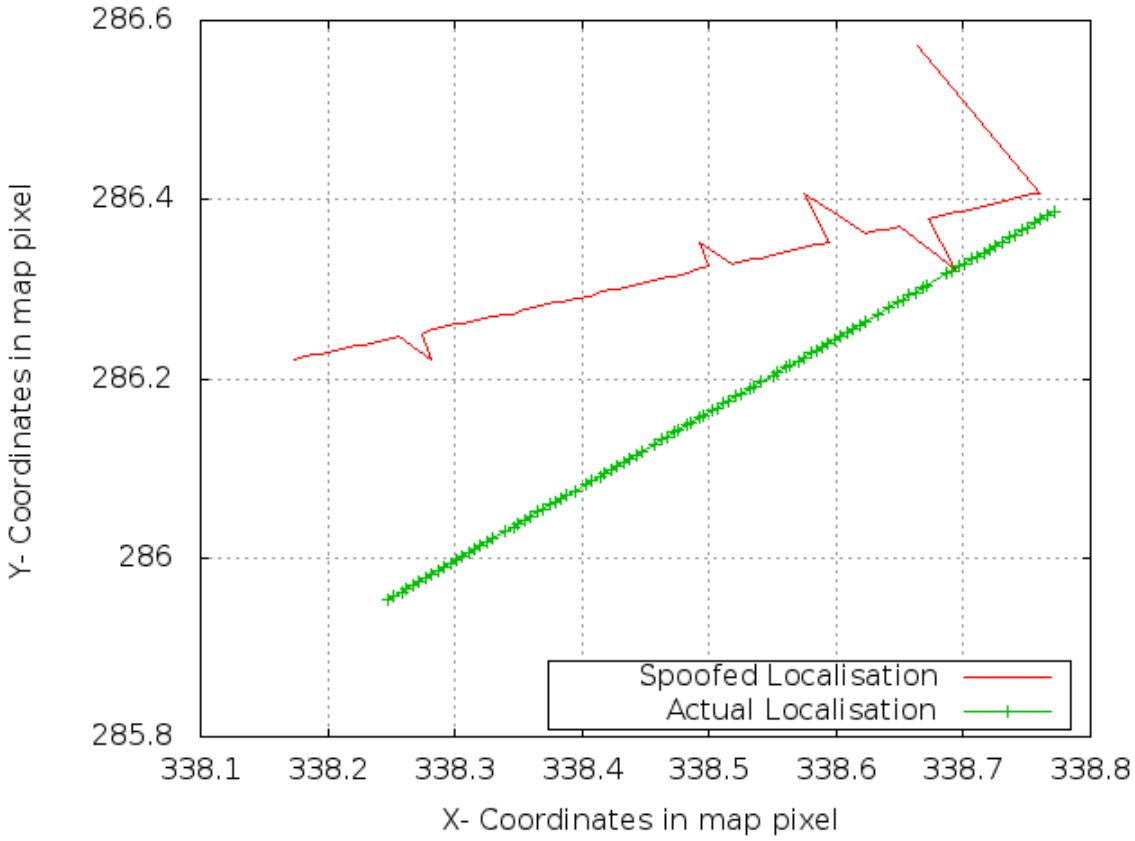

Figure 5-16: Effect of discrepancy introduction in Y-values of the spoofed GPS packet on Linear path of UAV

5.3 Results based on GPS Spoofing Attack

5.3.1 Effects of GPS Spoofing on Linear Path

Case I: Discrepancy in X-direction - In this case, we vary the x-value and keep the discrepancy increasing using the expression

$$x = x + (0.005 * s)$$

where s is initialized as 0 and incremented by 1 in each new packet generated. Figure 5-16 shows the results of this experiment. As seen in the figure, the original

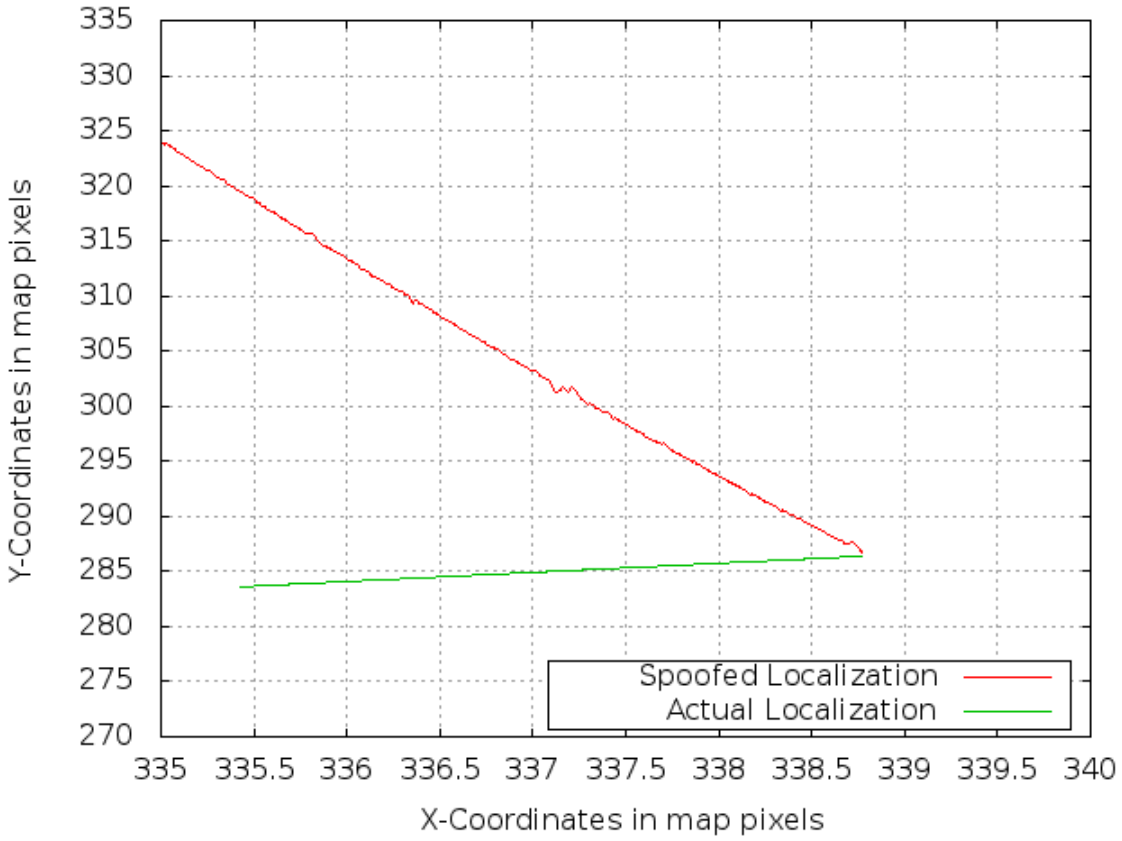

Figure 5-17: Effect of discrepancy introduction in Y-values of the spoofed GPS packet on Linear path of UAV

southwest direction of UAV is quite different than the spoofed direction of west. This shows an increase in calculated Y-values while a decrease in calculated X-values.

Case II: Discrepancy in Y-direction - In this case, we vary the y-value and keep the discrepancy increasing using the expression

$$y = y + (0.005 * s)$$

where s is initialized as 0 and incremented by 1 in each new packet generated. Figure 5-17 shows the result of this experiment. As seen in the figure, the (almost) west direction of UAV is the actual path while spoofed GPS makes the UAV think that

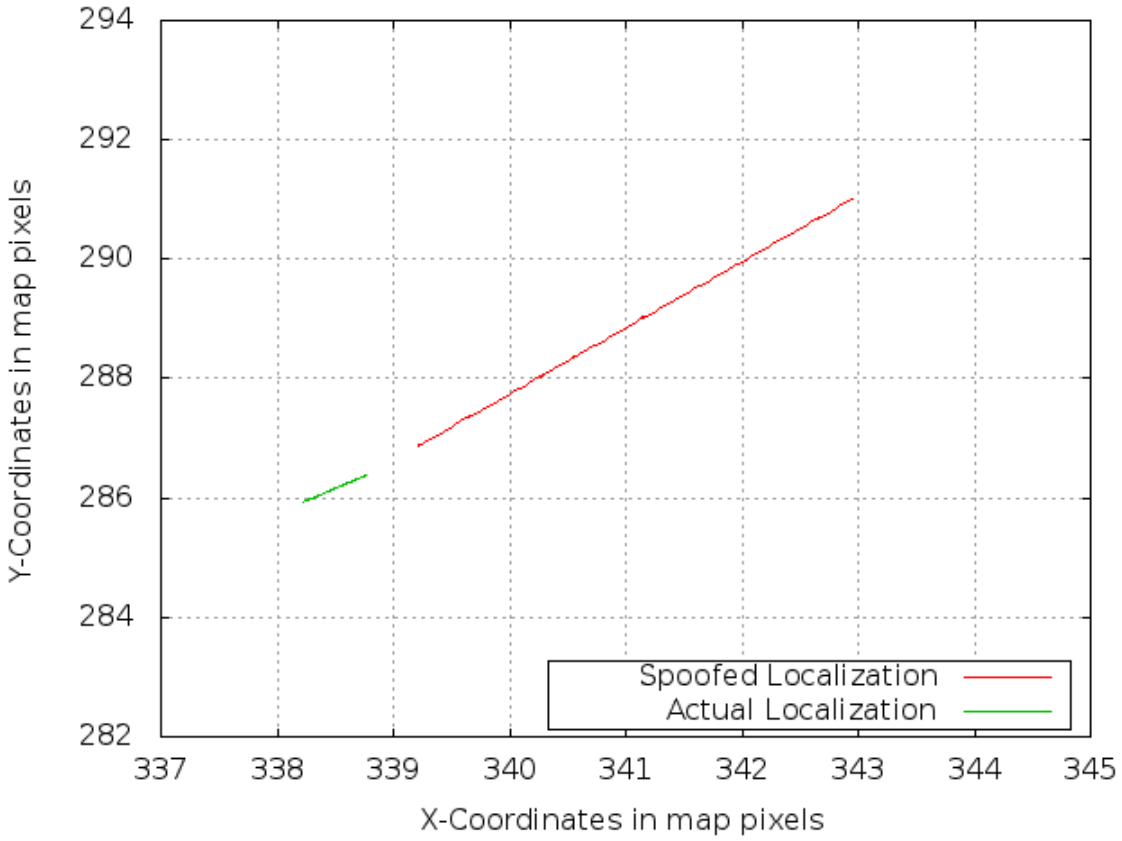

Figure 5-18: Effect of discrepancy introduction in both X and Y-values of the spoofed GPS packet on Linear path of UAV

it is going in the northwest direction. This shows a huge decrease in Y-values while very minimal impact on X-values comparatively. Clearly, this angle of variation will increase if we increase the discrepancy factor of 0.005.

Case III: Discrepancy in X and Y-direction - In this case, we vary the y-value and keep the discrepancy increasing using the similar expressions of Case I and II. Figure 5-18 shows the result of this experiment. As seen in the figure, the UAV thinks that it is moving in almost reverse of its actual direction. This shows that both X and Y values are now increasing very rapidly.

Case IV: Discrepancy in X, Y-direction and Distance - In this similar, a similar expression is used to introduce a discrepancy in all the three variables of x, y and the

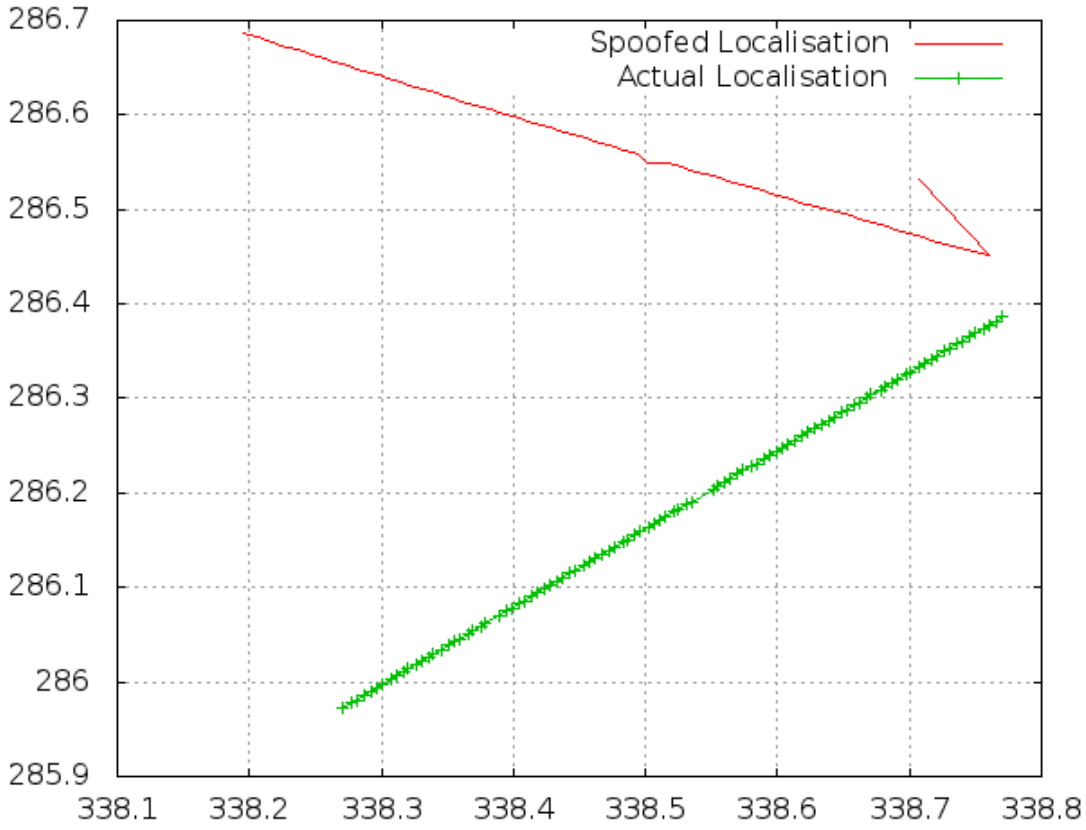

Figure 5-19: Effect of discrepancy introduction in distance as well as X and Y-values of the spoofed GPS packet on Linear path of UAV

distance. Figure 5-19 shows the result of this experiment. Such discrepancy introduction shows that the spoofed path is similar to the one obtained when discrepancy was introduced only in Y-values. This indicates that discrepancy in distance values somewhat negates the effect of discrepancy in X-values.

5.3.2 Effects of GPS Spoofing on Circular Path

In a second set of experiment, the GPS spoofing attack was carried out on a host moving in a circular path. Its initial position can be anywhere on a the circular path with radius of 1m and center (561m, 432m) on the map. The starting position was selected randomly in order to introduce randomness of UAV position and see if

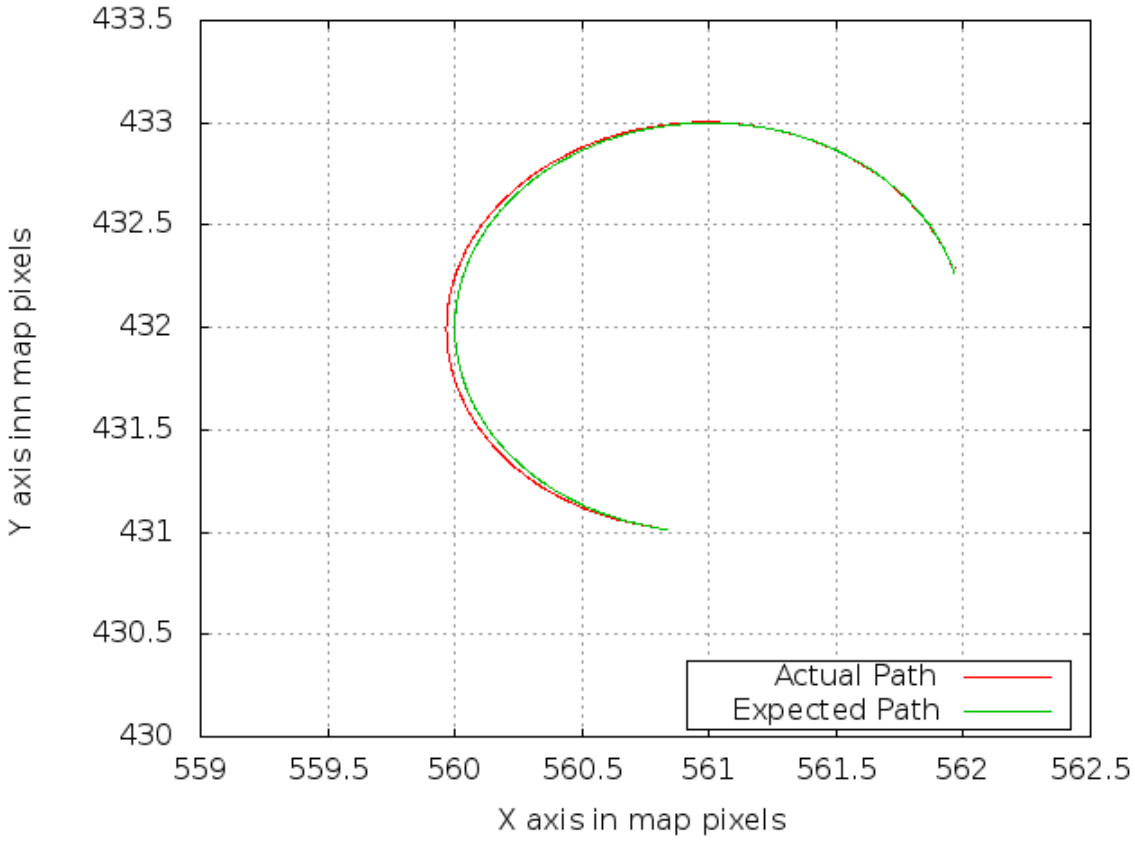

Figure 5-20: Effect of low discrepancy introduction in X-values of the spoofed GPS packet on Circular Path

results were location independent. The attack host also moves in a circular path with its starting position on a circular path of radius 2m and center (565m, 435m). The attacks were designed considering different data broadcast from the attack host. Five cases were analysed for this particular scenario which are different from linear path scenarios. These are discussed below:

Case I: Discrepancy in X-direction - In this case, a discrepancy s is added to X-values with a factor of 0.005 using the expression

$$x = x + (0.005 * s)$$

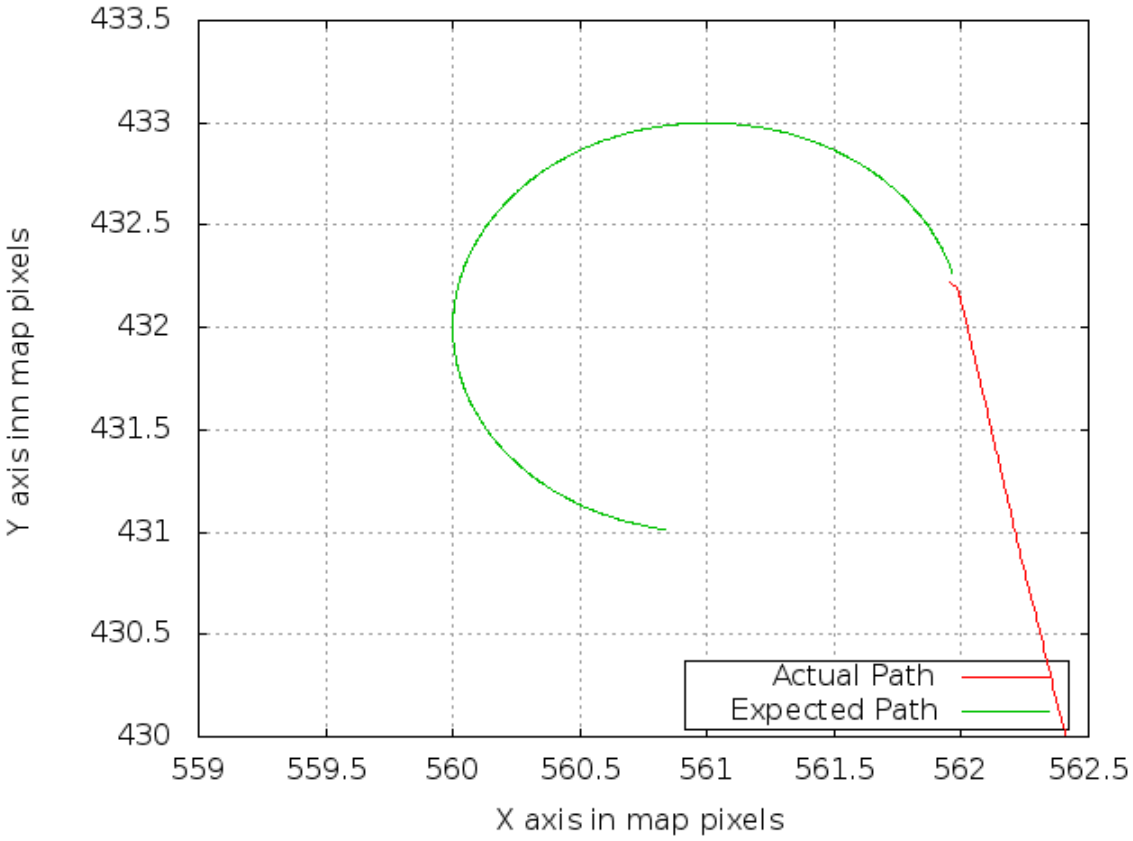

Figure 5-21: Effect of higher discrepancy introduction in X-values of the spoofed GPS packet on Circular Path

where, s is initialized as 10 and incremented by 1 as each new packet is generated. Figure 5-20 shows the obtained result for this experiment. It is clear that there was a very minor deviation of the host from its original circular path and the host traverses almost the same original path.

Case II: Higher discrepancy in X-direction - Since increasing the discrepancy factor little by little was not resulting in tangible changes, we increased the discrepancy factor in X-values by 3 times to 0.015 while keeping Y and distance values the same for this case. Similar to case I, s was initialized as 10 and incremented by 1 as each new packet is generated. Figure 5-21 shows the results obtained for this case. As shown, the spoofed path is quite different from the original path and becomes linear

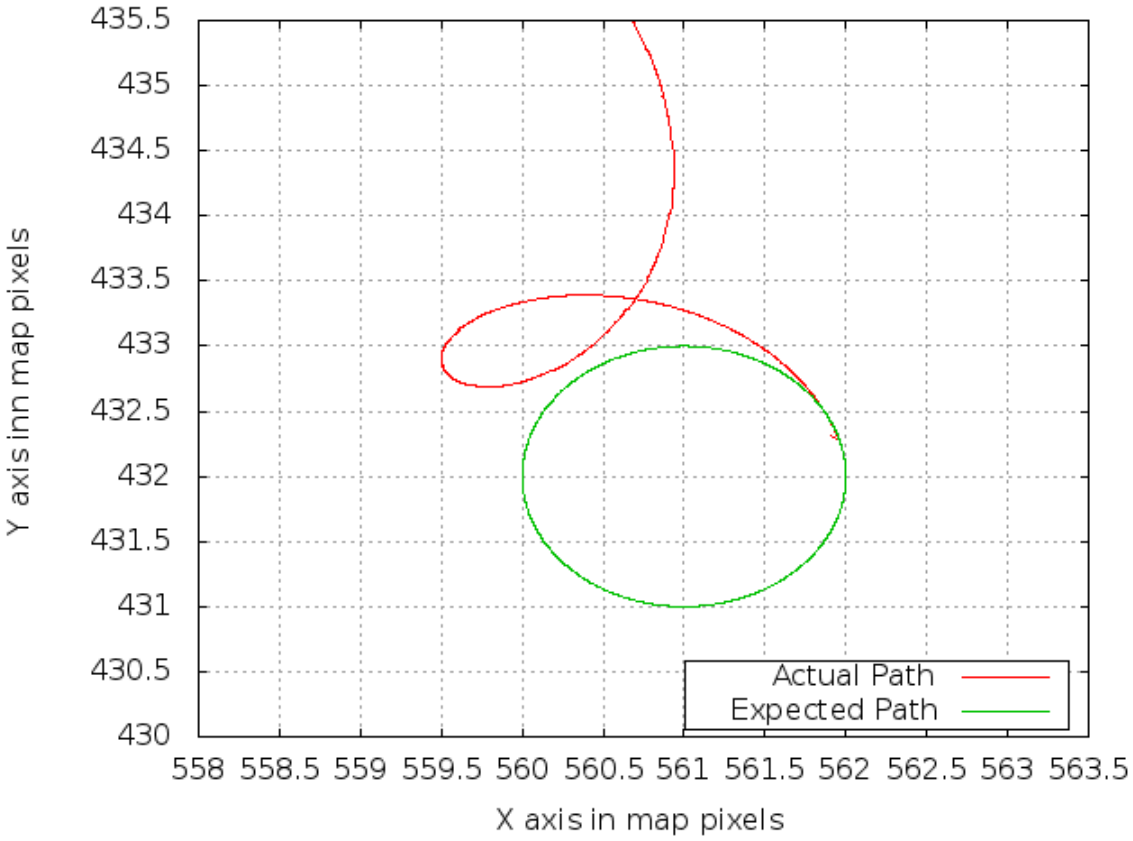

Figure 5-22: Effect of +ve discrepancy introduction in both X and Y-values of the spoofed GPS packet on Circular Path

starting from the original starting point in the opposite direction.

Case III: Positive discrepancy in X and Y-directions - In this case, we introduce positive discrepancy in both X and Y-values using similar expression as Case I. Result for this case is shown in Figure 5-22 and it shows that the host is actually moving outward in a helical path with varying pitch, while believing that it is moving in a circular path. It should be noted that the variation is mostly increasing Y-values and thus results in a helical path.

Case IV: Negative discrepancy in X and Y-directions - This case involves negative discrepancy introduction in both X and Y-values using similar expression as Case I. Related result are shown in Figure5-23 which clearly shows that the host followed

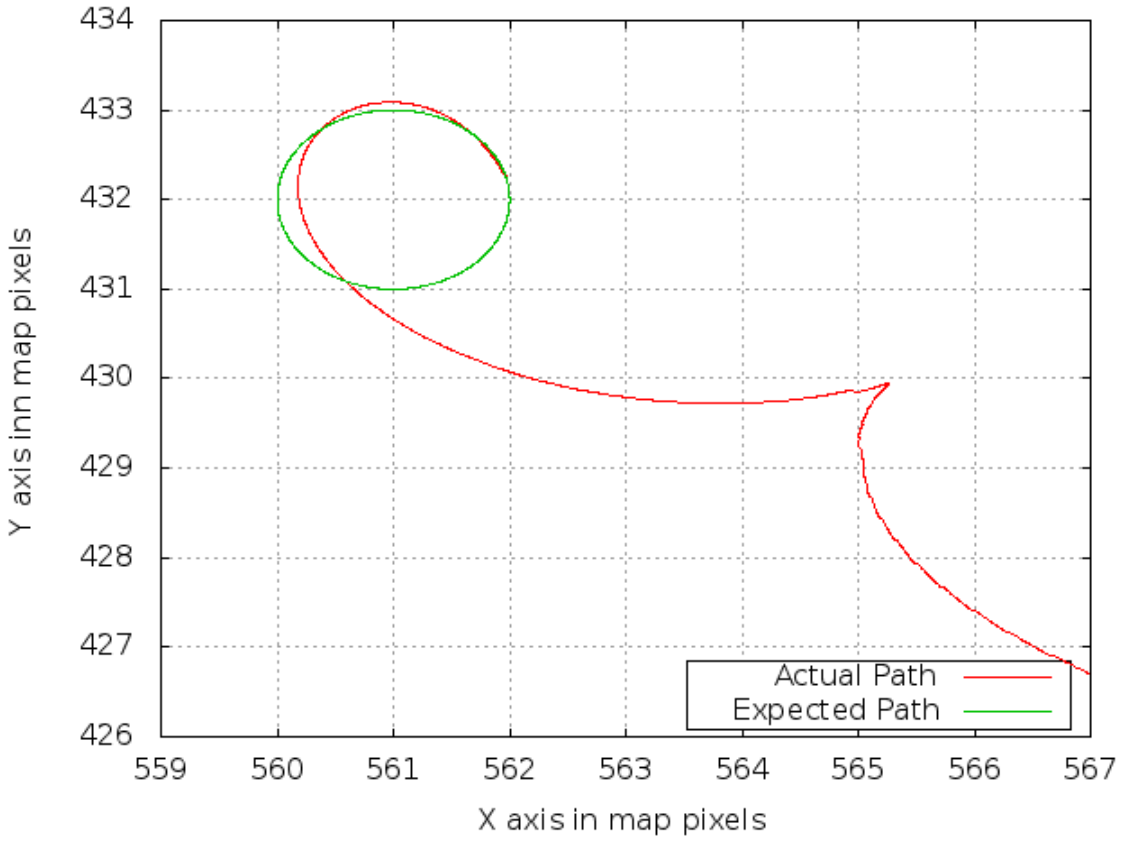

Figure 5-23: Effect of -ve discrepancy introduction in both X and Y-values of the spoofed GPS packet on Circular Path

the original path approximately and then moved outward following a modified helical paths. It should be noted that such a discrepancy is resulting in large negative variations in both X and Y-values.

Case V: Positive discrepancy in X-direction and Negative in Y-direction- In this last case, discrepancy was added to X-values and while subtracted from Y-values. Figure 5-24 shows the results obtained for this case. It can be seen that the host follows inward helical path moving away from its original position. It is clear from the graph that this kind of discrepancy is resulting in lower positive variation in Y-values while higher, or almost double positive variation in X-values, and this results in such an helical path.

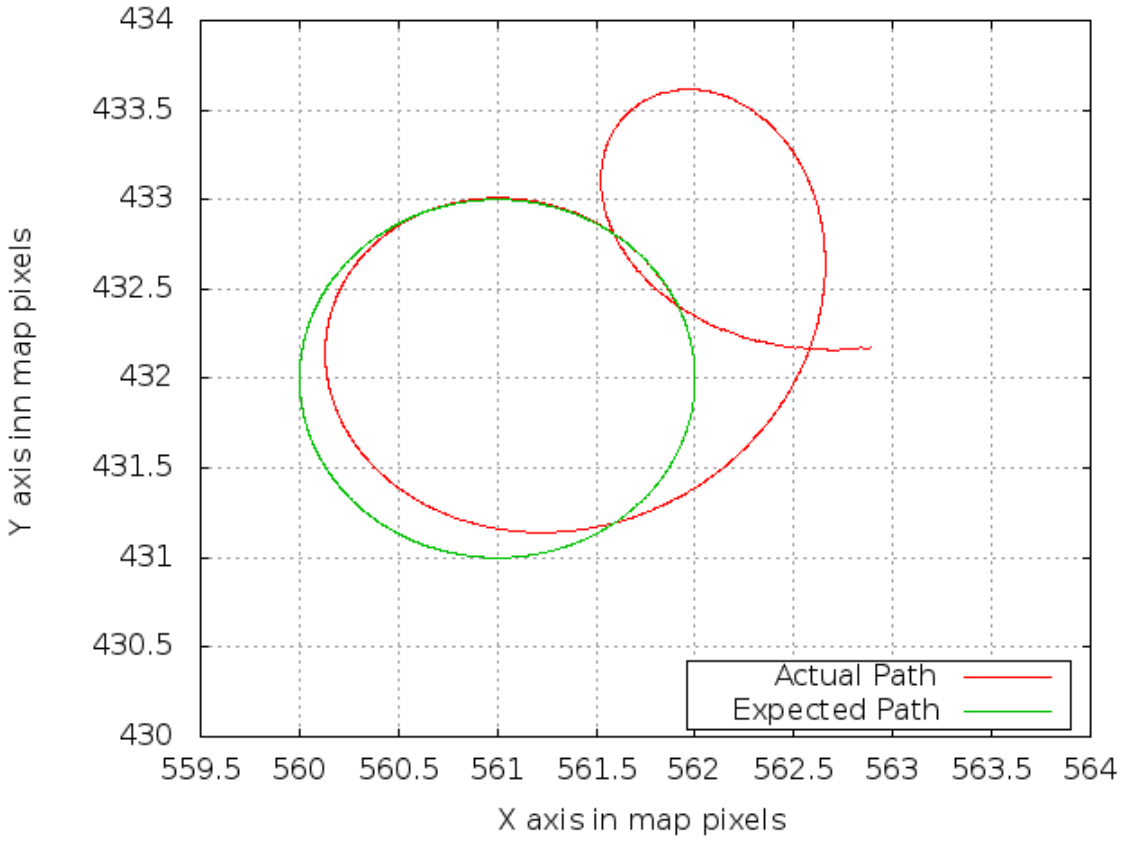

Figure 5-24: Effect of +ve discrepancy introduction in X-values and -ve discrepancy in Y-values of the spoofed GPS packet on Circular Path

5.3.3 Analysis

Through all these attack implementation, various results were obtained and valuable insights were gained. Some of them are listed here:

- Regarding GPS Jamming attack, the results were quite expected and variable, quite similar to real world jamming scenarios. As number of attack hosts were increased, average GPS packet loss increased and reached up to 90% which indicates quite successful jamming.

- When discrepancy was introduced in only X-values, it was noticed that different motion paths have different variations, which implies that the variations could not be generalized.

- Discrepancy factor variation results in variation of spoofed path as well. In case of circular path, this increase led to a spoofed linear path compared to a spoofed circular path when the factor was lower. Thus, low discrepancy factors would be hard to detect and can make a UAV lock on it as a real satellite then increase the factor to cause path deviation.

- In general, it was noticed that variation in Y-values are resulting in worse effects. In case of original linear path, the resultant deviations were huge while in case of original circular path, Y-value discrepancies caused resultant helical path, which could confuse the UAV and correction made to correct its path may lead to crash.

- In general, the spoofed paths are similar to original paths in terms of the class of curve, i.e., spoofed paths for original linear paths were linear while for circular paths, they were curved paths. This would result in tougher detection of discrepancy or path deviation if the discrepancy factor is quite low.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This work is an extension of previous work on UAVNet in UAVSim testbed. UAV is incomplete without a navigation module. It works on global positioning system in which UAVs receive GPS signals containing positional information broadcast by satellites. This information help UAVs know its location which can be used to plot its trajectory. To fully implement GPS, we needed GPS satellite constellation and a GPS signal receiver in UAV. This has been implemented in UAVSim. Simulations where performed with different parameter values in the simulation environment. Simulations were also done with different UAV mobility and results has been analyzed and discussed. To implement GPS attacks, some UAVHost were modified to be an attacker which would send counterfeit signals that would either jam real GPS signals or give false location information to UAVs. These attacks would make target UAV go astray or confuse its path. The results of such attacks was used to analyze the behavior of attacked UAV in different scenarios. Hence, this simulation testbed can be used to model real UAV and analyze all the factors and scenarios before testing it on civil grounds and avoid mishaps, major or minor. This will not only save high financial loss but would also avoid any causality to happen.

6.2 Future Work

This work shows the GPS implementation in 2D. Future works will be to mitigate the limitations of UAVSim and GPS/SATNAV implementation and to simulate it in a 3D environment. The attack simulations where done and analyzed with only one UAV. Next step would be to analyze how targeted UAV would behave if it was communicating with other UAVs. Also, anti-spoofing such as RAIM (Receiver Autonomous Integrity Monitoring) and anti-Jamming can be simulated to study how UAVs can be shielded from these attacks. Graphical User Interface and UAV model browser would be enhanced to include GPS attacks and plot its trajectory in the browser itself.

References

[1] Humanitarian uavs fly in china after earthquake (updated). http://irevolution.net/2014/08/25/humanitarian-uav-china-earthquake/, August 2014. Online; Last accessed: 16-Novemeber-2014.

[2] W. Sun A. Y. Javaid and M. Alam. Single and Multiple UAV Cyber-Attack Simulation and Performance Evaluation. *EAI Endorsed Transactions on Scalable Information Systems*, 1, 2014.

[3] Ilge Akkaya, Edward A Lee, and Patricia Derler. Model-based evaluation of gps spoofing attacks on power grid sensors. In *Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2013 Workshop on*, pages 1–6. IEEE, 2013.

[4] Andrew Amato. Is GPS Connectivity Important for Your Drone. http://dronelife.com/2014/10/24/drone-need-gps-connection/, October 2014. Online; Last accessed:19-November-2014.

[5] Karen Anderson and Kevin Gaston. Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Frontiers in Ecology and the Environment*, 11(3):138–146, March 2013.

[6] Reg Austin. *Unmanned Air Systems UAV Design, Development and Deployment*. Wiley, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, fisrt edition, May 2010.

[7] Ben Boughton. Unmanned Aerial Vehicles (UAV) in Precision Agriculture.

http://agmapsonline.com/?p=624, January 2014. Online; Last accessed:19-November-2014.

[8] Timothy X Brown, Sheetalkumar Doshi, Sushant Jadhav, and Jesse Himmelstein. Test Bed for a Wireless Network on Small UAVs. In *In Proceedings of AIAA 3rd Unmanned Unlimited Technical Conference*, pages 20–23, 2004.

[9] Barak J Carlson. Past uav program failures and implications for current uav programs. Technical report, DTIC Document, 2001.

[10] Serge Chaumette, Rémi Laplace, Christophe Mazel, and Raphaël Mirault. SCUAL, swarm of communicating uavs at LaBRI: An open UAVNet testbed. In *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, pages 1–5. IEEE, 2011.

[11] Metron High Perfomance Computing. Speeds. http://www.speedes.com/. Online; Last accessed:19-November-2014.

[12] J.J. Corner and G.B. Lamont. Parallel simulation of UAV swarm scenarios. In *Proceedings of the 2004 Winter Simulation Conference*, volume 1, pages –363, Dec 2004.

[13] Craig Whitlock. Crashes mount as military flies more drones in U.S. http://www.washingtonpost.com/sf/investigative/2014/06/22/crashes-mount-as-military-flies-more-drones-in-u-s/, June 2014. Online; Last accessed:01-February-2015.

[14] Dan Goodin. How to bring down mission-critical GPS networks with 2,500. http://arstechnica.com/security/2012/12/how-to-bring-down-mission-critical-gps-networks-with-2500/, December 2012. Online; Last accessed:07-February-2015.

[15] Daniel Shepard, Jahshan A. Bhatti, and Todd E. Humphreys. Drone Hack: Spoofing Attack Demonstration on a Civilian Unmanned Aerial Vehicle. http://gpsworld.com/drone-hack/, August 2012. Online; Last accessed:25-January-2015.

[16] Conor Dougherty. Google joins amazon in dreams of drone delivery. http://bits.blogs.nytimes.com/2014/08/28/google-joins-amazon-in-dreams-of-drone-delivery/?_r=0, August 2014. Online; Last accessed: 11-November-2014.

[17] FAA. *Unmanned Aircraft Systems*, November 2014. Online; Last accessed:19-November-2014.

[18] Bill Gabert. Nasa testing uav to detect fires. http://fireaviation.com/tag/uav/, October 2014. Online; Last accessed: 16-Novemeber-2014.

[19] Daniel Gilman. Unmanned aerial vehicles in humanitarian response. OCHA POLICY AND STUDIES. OCHA Policy Development and Studies Branch (PDSB)., June 2014. Online; Last accessed:16-November-2014.

[20] GPS World Staff. Massive GPS Jamming Attack by North Korea. http://gpsworld.com/massive-gps-jamming-attack-by-north-korea/, May 2012. Online; Last accessed:01-February-2015.

[21] GPS.Gov. *GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE SIGNAL SPECIFICATION*, June 1995.

[22] GPS.gov. Solar Storm Leaves GPS Service Intact. http://www.gps.gov/news/2012/03/solarstorm/, March 2012. Online; Last accessed:07-February-2015.

[23] GPS.Gov. Systems: Space segment. http://www.gps.gov/systems/gps/space/, October 2014. [Online; Last accesses: 11-November-2014].

[24] Graeme Green. Rhino-saving drones: How uavs are being used for wildlife conservation. http://metro.co.uk/2014/05/09/rhino-saving-drones-how-uavs-are-being-used-for-wildlife-conservation-4721692/ May 2014. Online; Last accessed:16-November-2014.

[25] S. Hamilton, T. Schmoyer, and J.A Drew Hamilton. Validating a network simulation testbed for army UAVs. In *2007 Winter Simulation Conference*, pages 1300–1305, Dec 2007.

[26] Brian Handwerk. 5 surprising drone uses (besides amazon delivery). http://news.nationalgeographic.com/news/2013/12/131202-drone-uav-uas-amazon-octocopter-bezos-science-aircraft-unmanned-robot/ December 2013. Online; Last accessed: 11-November-2014.

[27] K. Hartmann and C. Steup. The vulnerability of uavs to cyber attacks - an approach to the risk assessment. In *Cyber Conflict (CyCon), 2013 5th International Conference on*, pages 1–23, June 2013.

[28] IANS. Mumbai pizza delivery drones raise security buzz. http://economictimes.indiatimes.com/industry/services/hotels-/-restaurants/mumbai-pizza-delivery-drones-raise-security-buzz/articleshow/35486155.cms, 2014. [Online; Last accessed: 30-July-2014].

[29] Aeronautical Division IDS Corporation. Hero UAVSim Unmanned Aerial Vehicle Simulator. https://www.idscorporation.com/images/aeronautical/homepage/BRO_AERO_HEROSIM.pdf. Online; Last accessed:19-November-2014.

[30] Aeronautical Division IDS Corporation. Hero UAVSim Unmanned Aerial Vehicle Simulator. https://www.idscorporation.com/images/aeronautical/homepage/BRO_AERO_HEROGCS.pdf. Online; Last accessed:19-November-2014.

[31] Aeronautical Division IDS Corporation. Hero UAVSim Unmanned Aerial Vehicle Simulator. https://www.idscorporation.com/images/aeronautical/homepage/BRO_AERO_HEROUAV.pdf. Online; Last accessed:19-November-2014.

[32] National Instruments. Ni gps simulator pxi rf test system for gps receiver test. http://sine.ni.com/nips/cds/view/p/lang/en/nid/206805. Online; Last accessed:19-November-2014.

[33] Irene Klotz. 2 solar flares may mess with GPS. http://www.dispatch.com/content/stories/national_world/2014/09/12/2-solar-flares-may-mess-with-gps.html, September 2014. Online; Last accessed:07-February-2015.

[34] ITT Exelis. GPS Interference Detection and Geolocation System. http://www.exelisinc.com/tradeshows/ION/Documents/ITT_EXELIS_FINAL.pdf. Online; Last accessed:07-February-2015.

[35] Jarontec. uavplayground an approach to unmanned aerial vehicles (uav) with java and the processing development environment. https://code.google.com/p/uavplayground/, August 2010. Online; Last accessed:19-November-2014.

[36] Ahmad Y. Javaid, Weiqing Sun, and Mansoor Alam. UAVSim: A simulation testbed for unmanned aerial vehicle network cyber security analysis. In *2013 IEEE Globecom Workshops (GC Wkshps)*, pages 1432–1436, Dec 2013.

[37] Ahmad Y Javaid, Weiqing Sun, and Mansoor Alam. Uavsim: A simulation testbed for unmanned aerial vehicle network cyber security analysis. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 1432–1436. IEEE, 2013.

[38] A.Y. Javaid, Weiqing Sun, and M. Alam. UAVNet Simulation in UAVSim: A Performance Evaluation and Enhancement. In *9th International Conference*

on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM 2014), May 2014.

[39] A.Y. Javaid, Weiqing Sun, V.K. Devabhaktuni, and M. Alam. Cyber security threat analysis and modeling of an unmanned aerial vehicle system. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*, pages 585–590, Nov 2012.

[40] kborgen. Scratch build your own quad-copter! `http://www.instructables.com/id/Scratch-build-your-own-quad-copter/step2/Materials/`, August 2013. Online; Last accessed:19-November-2014.

[41] Heather Kelly. Drones: The future of disaster response. `http://whatsnext.blogs.cnn.com/2013/05/23/drones-the-future-of-disaster-response/`, May 2013. Online; Last accessed:16-November-2014.

[42] Alan Kim, Brandon Wampler, James Goppert, Inseok Hwang, and Hal Aldridge. Cyber attack vulnerabilities analysis for unmanned aerial vehicles. *The American Institute of Aeronautics and Astronautics: Reston, VA, USA*, 2012.

[43] Derek B Kingston and Randal W Beard. Real-time attitude and position estimation for small uavs using low-cost sensors. In *AIAA 3rd unmanned unlimited technical conference, Workshop and exhibit*, pages 2004–6488. sn, 2004.

[44] Miriam Kramer. European spacecraft lands on comet in historic space feat. `http://www.space.com/27740-rosetta-comet-landing-success.html`, November 2014. Online; Last accessed: 15-November-2014.

[45] M.E. Kuhl, J. Kistner, K. Costantini, and M. Sudit. Cyber attack modeling and simulation for network security analysis. In *Simulation Conference, 2007 Winter*, pages 1180–1188, Dec 2007.

[46] LabSat. Labsat gps simulator. http://www.labsat.co.uk/index.php/en/. Online; Last accessed:19-November-2014.

[47] Colin Lecher. Amazon is hiring drone testing pilots. http://www.theverge.com/2014/11/13/7213851/amazon-prime-air-drone-pilot-jobs, November 2014. Online; Last accessed: 15-November-2014.

[48] Andreas Lewandowski, Ralf Burda, and Christian Wietfeld. A multiscale real-time navigation and communication satellite simulation model for omnet++. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Simutools '08, pages 87:1–87:8, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[49] Daniel Marnach, Miguel Martins Sjouke Mauw, and Carlo Harpes. Detecting Meaconing Attacks by Analysing the Clock Bias of Gnss Receivers. In *Artificial Satellites*, volume 48, pages 63–83, June 2013.

[50] Brian Niehoefer, Sebastian Šubik, and Christian Wietfeld. The cni open source satellite simulator based on omnet++. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, SimuTools '13, pages 314–321, ICST, Brussels, Belgium, Belgium, 2013. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[51] NOAA Magazine. RESEARCHERS FIND GLOBAL POSITIONING SYSTEM IS SIGNIFICANTLY IMPACTED BY POWERFUL SOLAR RADIO BURST. http://www.noaanews.noaa.gov/stories2007/s2831.htm, April 2007. Online; Last accessed:07-February-2015.

[52] Brady W OHanlon, Mark L Psiaki, Todd E Humphreys, and Jahshan A Bhatti.

Real-time spoofing detection in a narrow-band civil gps receiver. *Proc. ION GNSS 2010*, pages 21–24, 2010.

[53] Joseph A Ouma, Wayne L Chappelle, and Amber Salinas. Facets of occupational burnout among us air force active duty and national guard/reserve mq-1 predator and mq-9 reaper operators. Technical report, DTIC Document, 2011.

[54] Eloi Pereira, Karl Hedrick, and Raja Sengupta. The C3UV Testbed for Collaborative Control and Information Acquisition Using UAVs. In *2013 American Control Conference (ACC)*, pages 1466–1471. IEEE, 2013.

[55] Yin Qiang, Xian Bin, Zhang Yao, Yu Yanping, Li Haotao, and Zeng Wei. Visual simulation system for quadrotor unmanned aerial vehicles. In *2011 30th Chinese Control Conference (CCC)*, pages 454–459, July 2011.

[56] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi. Uav photogrammetry for mapping and 3d modeling - current status and future perspectives. In M. Kunz H. Eisenbeiss and H. Ingensand, editors, *THE INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES*, Zurich, Switzerland, September 2011. Proceedings of the International Conference on Unmanned Aerial Vehicle in Geomatics (UAV-g).

[57] Elizabeth Rooney and Andrew Last. Glonass: As good as it should be? In *Proceedings of the 12th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1999)*, pages 1363–1368, 1999.

[58] FM Schubert, Roberto Prieto-Cerdeira, Patrick Robertson, and Bernard Henri Fleury. Snacs–the satellite navigation radio channel signal simulator. *Proc ION GNSS. Institute of Navigation, Savannah, GA*, pages 1982–1988, 2009.

[59] Hwajeong Seo and Howon Kim. Four anchor sensor nodes based localization algorithm over three-dimensional space. *Journal of information and communication convergence engineering*, 10(4):349–358, 2012.

[60] Daniel P Shepard, Jahshan A Bhatti, Todd E Humphreys, and Aaron A Fansler. Evaluation of smart grid and civilian uav vulnerability to gps spoofing attacks. In *Proceedings of the ION GNSS Meeting*, 2012.

[61] SOHO. Space Weather. `http://soho.nascom.nasa.gov/spaceweather/l`. Online; Last accessed:07-February-2015.

[62] Spirent. Gps simulation simulate gps signals for professional, controllable and testing in the lab. `http://www.spirent.com/Positioning-and-Navigation/GPS_Simulation`. Online; Last accessed:19-November-2014.

[63] M Thomas et al. Global navigation space systems: reliance and vulnerabilities. the royal academy of engineering. london, uk, 2011.

[64] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the requirements for successful gps spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 75–86. ACM, 2011.

[65] Brent M. Ledvina Todd E. Humphreys, Brady Mark L. Psiaki, W. O'Hanlon, and Paul M. Kintner Jr. Assessing the spoofing threat: Development of a portable gps civilian spoofer. In *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008),Savannah, GA, September 2008, pp. 2314-2325.*, 2008.

[66] Andrew J Turner. *An open-source, extensible spacecraft simulation and modeling environment framework*. PhD thesis, Virginia Polytechnic Institute and State University, 2003.

[67] Jinling Wang, Matthew Garratt, Andrew Lambert, Jack Jianguo Wang, S Han, and David Sinclair. Integration of gps/ins/vision sensors to navigate unmanned aerial vehicles. *IAPRSSIS*, 37(B1):963–9, 2008.

[68] What when how. A Few Standard Orbits (Orbital Mechanics Interlude) (Remote Sensing). `http://what-when-how.com/remote-sensing-from-air-and-space/a-few-standard-orbits-orbital-mechanics-interlude-remote-sensing/`. Online; Last accessed:19-November-2014.

[69] Prof. Dr. Ing. C. Wietfeld. The open source satellite simulator. `http://www-os3.kn.e-technik.tu-dortmund.de/index.php`, 2014. Online; Last accessed:19-November-2014.

[70] Prof. Dr. Ing. C. Wietfeld. The open source satellite simulator-features. `http://www-os3.kn.e-technik.tu-dortmund.de/index.php/features`, 2014. Online; Last accessed:19-November-2014.

[71] FlightGear Wiki. Global positioning system. `http://wiki.flightgear.org/Global_Positioning_System`, December 2013. Online; Last accessed: 19-November-2014.

[72] Wikipedia. Trilateration. `http://en.wikipedia.org/wiki/Trilateration`, November 2014. Online; Last accessed:19-November-2014.

[73] Jianan Wu, Wei Wang, Jinhong Zhang, and Bodong Wang. Research of a kind of new UAV training simulator based on equipment simulation. In *2011 International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT)*, volume 9, pages 4812–4815, Aug 2011.

[74] Joo Beom Yun, Eung Ki Park, Eul Gyu Im, and Hoh Peter In. A scalable,

ordered scenario-based network security simulator. In *Systems Modeling and Simulation: Theory and Applications*, pages 487–494. Springer, 2005.

[75] Frank Zimmermann, Thomas Haak, and Chris Hill. Galileo system simulation facility. In *8th International Workshop on Simulation for European Space Programmes, SESP*, 2004.